

Usability-Analyse und Optimierung des Bedienkonzeptes im Content- Management-System TYPO3

Eine Untersuchung aus der Perspektive von
Content-Editor:innen

vorgelegt von

Jessica Charzynski

Matrikelnummer: 923010

dem Fachbereich VI – Informatik und Medien
der Berliner Hochschule für Technik Berlin
vorgelegte Bachelorarbeit
zur Erlangung des akademischen Grades

Bachelor of Science (B.Sc.)

im Studiengang

Medieninformatik Online

Tag der Abgabe 8. Februar 2026



Betreuer

Sebastian Kreideweiß, M.Sc. Technische Hochschule Brandenburg

Gutachter

Prof. Dr. Hansjörg Mixdorff

Berliner Hochschule für Technik

Kurzfassung

Die vorliegende Arbeit untersucht die Usability des Backends des Content-Management-Systems TYPO3 aus Sicht von Content-Editor:innen. Ziel ist es, die Nutzerführung und die Unterstützung redaktioneller Arbeitsprozesse zu analysieren und gezielt zu verbessern. Der Fokus liegt auf der Bedienbarkeit zentraler Funktionen sowie auf der Effizienz wiederkehrender Arbeitsabläufe. Die Untersuchung basiert auf einer heuristischen Analyse des TYPO3-Backends sowie auf einem Vergleich mit den Content-Management-Systemen WordPress, Drupal und Storyblok. Auf dieser Grundlage wird ein Optimierungskonzept entwickelt und in einem klickbaren Prototyp visualisiert, der anschließend evaluiert wird. Die Arbeit leistet damit einen Beitrag zur systematischen Auseinandersetzung mit der Backend-Usability von TYPO3 und gibt Impulse für dessen nutzerzentrierte Weiterentwicklung.

Eidesstattliche Erklärung

Ich versichere, dass ich diese Abschlussarbeit ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

Mühlheim am Main, 08.02.2026

Jessica Charzynski

Inhaltsverzeichnis

1	Einleitung	1
1.1	Hintergrund und Motivation	1
1.2	Erkenntnisinteresse und Forschungsfrage	2
1.3	Zielsetzung	2
2	Grundlagen	3
2.1	Theoretischer Rahmen	3
2.1.1	Usability-Prinzipien und -Heuristiken	4
2.1.2	User Experience Design	5
2.1.3	User-Centered und Human-Centered Design	7
2.1.4	Mensch-Computer-Interaktion	8
2.2	TYPO3 als Content-Management-System	10
2.2.1	Technische Architektur	10
2.2.2	Rollen und Aufgaben	11
2.2.3	Prozesse der Inhaltsverwaltung	12
2.3	Forschungsstand zur Usability des TYPO3-Backends	13
2.3.1	Praxisgetriebene UX-Verbesserungen im TYPO3-Umfeld	14
2.3.2	Forschungsstand und Forschungslücken	14
3	Methodik	16
3.1	Analyse des aktuellen TYPO3-Backends	16
3.1.1	Dashboard als Einstiegspunkt in das Backend	16
3.1.2	Workflow-Effizienz	17
3.1.3	Barrierefreiheit	19
3.2	Vergleichende Analyse von Content-Management-Systemen	20
3.2.1	Analyse des WordPress-Backends	20
3.2.2	Analyse des Drupal-Backends	22
3.2.3	Analyse des Storyblok-Backends	23
3.2.4	Ableitung übertragbarer Best Practices	24
4	Optimierungskonzept	25
4.1	Ableitung von Optimierungspotenzialen	25
4.2	Entwurf optimierter Interface-Strukturen	26
4.3	Optimierung redaktioneller Workflows	27
5	Erstellung des Prototyps	29
5.1	Methodisches Vorgehen	29
5.2	Informationsarchitektur und Oberflächenstrukturierung	29
5.3	Visuelles Design und Interaktionskonzept in Figma	32
5.4	Validierung der Prototyp-Funktionalitäten	36

6 Evaluation	37
6.1 Vorgehen und Aufbau der Evaluation	37
6.2 Evaluation aus UX-Perspektive	38
6.3 Evaluation aus der Perspektive von Content-Editor:innen	39
6.4 Evaluation aus Entwicklungs- und Systemperspektive	40
7 Fazit und Ausblick	41
Literaturverzeichnis	43
A Prototyp	47
B Prototyp-Feedback	48
C Experteninterview	49

Abbildungsverzeichnis

2.1	Konzeptioneller Zusammenhang zwischen System, Usability, Nutzungskontext und User Experience. Eigene Darstellung in Anlehnung an [1]	7
3.1	TYPO3 Dashboard der Demo Installation mit vorkonfigurierten Standardwidgets	17
3.2	Anlegen einer neuen Subpage im Seitenbaum des TYPO3-Backends	18
3.3	Kopieren eines bestehenden Inhaltselements zur Wiederverwendung	18
3.4	Dashboard des WordPress-Backends mit vorkonfigurierten Standard-Widgets .	21
3.5	Visuelle Inhaltsbearbeitung mit dem blockbasierten WordPress-Editor	21
3.6	Dashboard-Ansicht des Drupal-Backends	22
3.7	Inhaltsbearbeitung in Drupal mit überlagernden Assistenz-Overlays	23
3.8	Storyblok-Dashboard mit Projektübersicht und Activity-Stream	23
3.9	Inhaltsbearbeitung in Storyblok mit Visual Editor und seitlicher Formularansicht	24
4.1	Pen-&-Paper-Wireframe eines TYPO3-Dashboards zur konzeptionellen Strukturierung von Layout und Informationsarchitektur	27
5.1	Dashboard-Ansicht im Figma-Prototyp	30
5.2	Seitenübersicht mit Status- und Sprachvergleich im Figma-Prototyp	31
5.3	Bearbeitung einer Hero Section im Figma-Prototyp	31
5.4	Frontend Editing eines Headline-Segments im Figma-Prototyp	33
5.5	Tabbed-Content-Sektion im Figma-Prototyp	34
5.6	Detailbearbeitung eines Tabbed-Content-Eintrags im Figma-Prototyp	35
5.7	Grid-Sektion mit Card-Elementen im Figma-Prototyp	35
5.8	Interaktive Klickpfade zwischen Bearbeitungsansichten im Figma-Prototyp . .	36
A.1	Übersicht aller Prototyp-Screens der optimierten TYPO3-Backend-Oberfläche .	47
B.1	Slack-Thread mit Feedback von TYPO3-Community-Mitgliedern	48
B.2	Dokumentation und Strukturierung des Feedbacks im Miro-Board	48

Abkürzungsverzeichnis

Abkürzung	Bedeutung
CLT	Cognitive Load Theory
CMS	Content-Management-System
EAA	European Accessibility Act
FAL	File Abstraction Layer
HCD	Human-Centered Design
MCI	Mensch-Computer-Interaktion
RBAC	Role-Based Access Control
UCD	User-Centered Design
UI	User Interface
UX	User Experience
WCAG	Web Content Accessibility Guidelines
WYSIWYG	What You See Is What You Get

Kapitel 1

Einleitung

Digitale Medien sind ein zentraler Bestandteil der heutigen Informationsvermittlung. Daraus ergibt sich die Notwendigkeit, Inhalte kontinuierlich zu aktualisieren, konsistent zu gestalten und zielgruppengerecht über zuverlässige digitale Angebote bereitzustellen.

1.1 Hintergrund und Motivation

Für Unternehmen und öffentliche Einrichtungen stellt der eigene Webauftritt einen zentralen Kommunikationskanal dar, über den eine konsistente digitale Präsenz aufgebaut und Informationen bereitgestellt werden. Inhalte werden regelmäßig erstellt, überarbeitet und erweitert, häufig durch Content-Editor:innen, deren Aufgabenprofil überwiegend kommunikativ und organisatorisch geprägt ist und sich nicht auf technische Systemkonfiguration ausrichtet. Ein Content-Management-System (CMS) unterstützt diese Prozesse, indem es die Erstellung, Pflege und Publikation digitaler Inhalte über eine klar strukturierte Oberfläche ermöglicht, redaktionelle Aufgaben von der technischen Website-Entwicklung trennt und Workflows für kollaborative Inhaltserstellung sowie Versionierung bereitstellt [2, S. 556].

TYPO3 zählt in diesem Kontext zu den etablierten Open-Source-Content-Management-Systemen und ist insbesondere im deutschsprachigen Hochschulbereich weit verbreitet. Mit Stand 2025 verwenden 49,8 % der deutschen Hochschulen TYPO3 als Content-Management-System [3].

Gleichzeitig gilt die hohe Funktionsdichte von TYPO3 als Herausforderung für die Bedienbarkeit, insbesondere für Nutzer:innen mit vorwiegend redaktionellem Aufgabenschwerpunkt. Die Kombination aus komplexen Navigationsstrukturen, zahlreichen Modulen und vielfältigen Konfigurationsmöglichkeiten kann die Orientierung erschweren und die kognitive Belastung erhöhen. Vor diesem Hintergrund besteht das Risiko, dass typische Aufgaben wie das Erstellen, Bearbeiten und Publizieren von Inhalten als aufwendig wahrgenommen werden oder fehleranfällig ablaufen [4].

Die Spannung zwischen einem funktionsreichen System und einer zugleich intuitiven Nutzung bildet den Ausgangspunkt dieser Arbeit. Im Mittelpunkt steht die Frage, wie die Nutzerführung in TYPO3 so gestaltet werden kann, dass Content-Editor:innen ihre Aufgaben effizient und mit möglichst geringer kognitiver Belastung erledigen können. Die Motivation besteht darin, konkrete Ansatzpunkte für Verbesserungen zu identifizieren und in ein nutzerzentriertes Optimierungskonzept zu überführen.

Aktuell wird mit TYPO3 Version 14 eine neue Hauptversion entwickelt, deren Roadmap [5] "User Experience (UX) Improvements" als erstes High-Level Objective hervorhebt. Dies verdeutlicht, dass die Qualität der Nutzererfahrung im Backend einen zentralen Entwicklungsschwerpunkt darstellt und dass die in dieser Arbeit vorgenommene Auseinandersetzung in einem fachlich wie zeitlich aktuellen Kontext steht.

Darüber hinaus erhalten die in dieser Arbeit gewonnenen Erkenntnisse einen erweiterten Wir-

kungsraum, da der Government Site Builder in Version 11 technisch auf TYPO3 basiert und dessen Backend als redaktionelle Oberfläche nutzt [6]. Verbesserungen der Nutzerführung im TYPO3-Backend betreffen damit nicht nur TYPO3 selbst, sondern zugleich den Government Site Builder 11 und damit zahlreiche staatliche Webauftritte des Bundes.

1.2 Erkenntnisinteresse und Forschungsfrage

Angesichts der komplexen Funktionsdichte des TYPO3-Backends stellt sich die Frage, welche konkreten Nutzungshürden für Content-Editor:innen bestehen und wie sich diese durch eine optimierte Nutzerführung reduzieren lassen. Ausgangspunkt der Arbeit ist damit die Perspektive derjenigen, die Inhalte im Alltag erstellen, bearbeiten und veröffentlichen und dabei auf eine klar strukturierte und gut nachvollziehbare Oberfläche angewiesen sind. Im Zentrum steht die Überlegung, wie Aufbau, Navigation und Rückmeldungen des Systems so gestaltet werden können, dass typische Aufgaben mit weniger Aufwand, geringerer kognitiver Belastung und einem geringeren Fehlerrisiko erledigt werden. Zugleich soll nachvollziehbar werden, wie sich Unterschiede in Vorerfahrung, Schulungsstand und Nutzungskontext auf die Wahrnehmung der Bedienoberfläche auswirken.

Zunächst soll daher systematisch sichtbar gemacht werden, an welchen Stellen das aktuelle Backend die Arbeit erschwert, etwa durch unübersichtliche Strukturen, schwer auffindbare Funktionen oder inkonsistente Bezeichnungen. Ziel ist es, nicht nur einzelne Probleme zu beschreiben, sondern jene Aspekte der Nutzerführung zu identifizieren, die den Arbeitsfluss besonders stark beeinflussen und als zentrale Ansatzpunkte für Optimierungen dienen.

Darauf aufbauend richtet sich das Erkenntnisinteresse auf die Frage, welche konkreten Veränderungen kurz- und langfristig sinnvoll und realistisch umsetzbar sind. Im Blick sind einerseits kleinere Anpassungen, die mit überschaubarem Aufwand realisiert werden können, andererseits aber auch weitergehende konzeptionelle Ideen, die das Backend langfristig verständlicher und konsistenter machen. Die leitende Forschungsfrage der Arbeit lautet daher:

Wie kann die Nutzerführung im TYPO3-Backend durch umsetzbare Optimierungsmaßnahmen und weiterführende Neugestaltungsideen so verbessert werden, dass Arbeitsabläufe für Content-Editor:innen intuitiver und effizienter werden?

1.3 Zielsetzung

Die Arbeit hat das Ziel, die Nutzerführung in TYPO3 aus Sicht von Content-Editor:innen systematisch zu untersuchen und darauf aufbauend ein fundiertes Optimierungskonzept zu entwickeln. Im Mittelpunkt steht die Frage, wie sich typische Arbeitsabläufe verständlicher, konsistenter und effizienter gestalten lassen, ohne die funktionale Breite des Systems einzuschränken. Ausgehend von einer heuristischen Analyse werden zunächst jene Stellen identifiziert, an denen Strukturen, Navigation oder Rückmeldungen die Arbeit im System unnötig erschweren. Ergänzend liefert ein Vergleich mit anderen Content-Management-Systemen Hinweise auf übertragbare Gestaltungsprinzipien.

Darauf aufbauend werden Optimierungspotenziale priorisiert und in einem zweistufigen Konzept ausgearbeitet, das sowohl kurzfristig umsetzbare Verbesserungen als auch weitergehende strukturelle Ansätze für ein verbessertes Bedienkonzept umfasst. Die entwickelten Vorschläge werden in einem klickbaren Figma-Prototyp visualisiert und anschließend in der TYPO3-Community geteilt. Ziel ist es, Rückmeldungen zur Verständlichkeit, Praxistauglichkeit und Relevanz der vorgeschlagenen Maßnahmen zu erhalten.

Die Arbeit verbindet damit die Analyse bestehender Usability-Schwächen mit der Entwicklung und Evaluation praxisorientierter Lösungsansätze zur Verbesserung der Arbeitsbedingungen von Content-Editor:innen im TYPO3-Backend.

Kapitel 2

Grundlagen

Das folgende Kapitel legt die fachliche Grundlage für die Analyse und Weiterentwicklung des TYPO3-Backends aus der Perspektive von Content-Editor:innen. Zunächst wird der theoretische Rahmen skizziert, innerhalb dessen die Interaktion zwischen Mensch und System betrachtet wird. Dabei werden zentrale begriffliche Unterscheidungen sowie Bewertungsmaßstäbe eingeführt, die für die spätere Analyse leitend sind. Anschließend erfolgt eine Einordnung von TYPO3 als Content-Management-System, bevor der bisherige Forschungsstand zur Usability des TYPO3-Backends dargestellt wird. Auf diese Weise wird die vorliegende Arbeit im bestehenden fachlichen Diskurs verortet.

2.1 Theoretischer Rahmen

Der theoretische Rahmen dieser Arbeit bündelt zentrale Konzepte aus den Bereichen Usability, User Experience (UX), nutzerzentrierte Gestaltung und Mensch-Computer-Interaktion (MCI). Er definiert die fachliche Perspektive, aus der die Interaktion von Content-Editor:innen mit dem TYPO3-Backend betrachtet wird, und legt die Kriterien fest, anhand derer das bestehende Bedienkonzept analysiert und bewertet werden kann. Ziel ist es, die Nutzung des Systems nicht aus einer rein technischen Sicht, sondern aus der Perspektive realer Arbeitsprozesse und Nutzungserfahrungen einzuordnen.

Usability und User Experience bilden dabei zwei komplementäre Betrachtungsebenen. Während Usability die funktionale Qualität der Nutzung in den Blick nimmt und danach fragt, inwieweit Aufgaben effektiv, effizient und fehlerarm ausgeführt werden können, richtet sich die User-Experience-Perspektive stärker auf das subjektive Erleben im Umgang mit dem System. Sie berücksichtigt Aspekte wie wahrgenommene Kontrolle, Klarheit, Vertrauen und Akzeptanz. Beide Konzepte sind für die Analyse des TYPO3-Backends relevant, da redaktionelle Arbeit sowohl funktionale Anforderungen als auch qualitative Erwartungen an die tägliche Nutzung eines Arbeitssystems stellt.

Ergänzt werden diese Perspektiven durch Ansätze der nutzerzentrierten und menschenzentrierten Gestaltung. Sie liefern konzeptionelle Leitlinien dafür, wie interaktive Systeme systematisch an den Bedürfnissen, Fähigkeiten und Nutzungskontexten ihrer Anwender:innen ausgerichtet werden können. Gerade bei komplexen Content-Management-Systemen ist diese Perspektive zentral, da Gestaltung hier immer im Spannungsfeld zwischen technischer Flexibilität, organisatorischen Vorgaben und individueller Nutzbarkeit erfolgt.

Die Auswahl dieser theoretischen Konzepte ergibt sich aus den Besonderheiten redaktioneller Arbeitsumgebungen. Content-Management-Systeme wie TYPO3 werden in der Regel nicht explorativ genutzt, sondern dienen der effizienten Bewältigung klar definierter, wiederkehrender Aufgaben. Dazu zählen unter anderem das Erstellen, Überarbeiten, Übersetzen und Veröffentlichen von Inhalten innerhalb vorgegebener Strukturen und Workflows. Entscheidend für die

Beurteilung der Interaktionsqualität ist daher weniger die isolierte Betrachtung einzelner Gestaltungselemente als vielmehr die Frage, ob diese Arbeitsprozesse nachvollziehbar unterstützt werden, Orientierungsprobleme vermieden und fehlerarmes Arbeiten ermöglicht wird.

Die Mensch-Computer-Interaktion bildet in diesem Zusammenhang den übergeordneten Bezugsrahmen, in dem das TYPO3-Backend als sozio-technisches Arbeitssystem verstanden wird. In diesem System wirken Mensch, Software und organisatorische Rahmenbedingungen zusammen und beeinflussen sich gegenseitig [7]. MCI ermöglicht es, Interaktionsprobleme nicht nur auf Ebene der Benutzeroberfläche zu beschreiben, sondern sie in Beziehung zu Arbeitsabläufen, Rollenmodellen und Systemstrukturen zu setzen. Damit schafft sie die theoretische Grundlage, um Nutzungshürden systematisch zu analysieren und deren Ursachen nachvollziehbar zu identifizieren.

Die in diesem Kapitel dargestellten Konzepte werden in der vorliegenden Arbeit nicht isoliert betrachtet, sondern als Bestandteile eines gemeinsamen theoretischen Rahmens verstanden. Usability- und User-Experience-Ansätze beschreiben, wie sich das TYPO3-Backend funktional nutzen lässt und wie es von den Anwender:innen wahrgenommen wird. Nutzer- und menschenzentrierte Gestaltungsansätze liefern Orientierung für eine stärkere Ausrichtung an den Bedürfnissen der Content-Editor:innen, während die Mensch-Computer-Interaktion die Einbettung des Systems in reale Arbeitskontexte ermöglicht. Die folgenden Unterkapitel konkretisieren diese Perspektiven und bilden damit die Grundlage für die anschließende Ist-Analyse des TYPO3-Backends sowie für die Ableitung fundierter Optimierungsansätze.

2.1.1 Usability-Prinzipien und -Heuristiken

Usability (deutsch: Gebrauchstauglichkeit) beschreibt, in welchem Maß ein interaktives System bestimmte Nutzer:innen innerhalb eines definierten Nutzungskontexts dabei unterstützt, ihre Ziele effektiv, effizient und zufriedenstellend zu erreichen [8, S. 6]. Die internationale Norm ISO 9241-11 präzisiert diesen Begriff und legt fest, dass Gebrauchstauglichkeit stets in Relation zu den beteiligten Nutzer:innen, deren Aufgaben, der jeweiligen Arbeitsumgebung sowie den organisatorischen Rahmenbedingungen zu bewerten ist [9]. Usability wird damit nicht als isolierte Eigenschaft einer Benutzeroberfläche verstanden, sondern als Ergebnis des Zusammenspiels zwischen Mensch, Aufgabe, Technik und Kontext. Für das in dieser Arbeit untersuchte Bedienkonzept bedeutet dies, dass nicht einzelne Interface-Elemente im Vordergrund stehen, sondern die Qualität der Unterstützung redaktioneller Arbeit als zusammenhängender Prozess betrachtet wird.

Für die Analyse des TYPO3-Backends folgt daraus, dass Gebrauchstauglichkeit nicht punktuell anhand einzelner Bedienelemente beurteilt werden kann. Maßgeblich ist vielmehr, inwieweit typische redaktionelle Tätigkeiten wie das Erstellen, Bearbeiten, Übersetzen und Veröffentlichen von Inhalten konsistent unterstützt werden und ohne unnötige Umwege, Fehlinterpretationen oder Unterbrechungen ausführbar sind. Usability wird somit zu einer Eigenschaft des gesamten Arbeitssystems, das aus Benutzeroberfläche, Systemlogik, zugrunde liegenden Arbeitsabläufen und organisatorischen Vorgaben besteht. Eine rein visuelle oder funktionale Betrachtung einzelner Module greift daher zu kurz, wenn sie nicht in den Kontext realer redaktioneller Arbeitsprozesse eingebettet ist.

Zentral für die systematische Bewertung von Usability sind die drei Kernkriterien Effektivität, Effizienz und Zufriedenheit [9]. Effektivität beschreibt, in welchem Maß Nutzer:innen ihre Aufgaben korrekt und vollständig ausführen können, unabhängig vom dabei entstehenden Aufwand. Effizienz bezieht sich auf den erforderlichen Ressourceneinsatz, etwa in Form von Bearbeitungszeit, Anzahl der notwendigen Interaktionsschritte oder kognitiver Belastung. Zufriedenheit schließlich erfasst die subjektive Wahrnehmung der Nutzung, beispielsweise ob diese als unterstützend, verständlich oder belastend erlebt wird. Gerade bei komplexen Systemen wie Enterprise-CMS kann eine hohe Effektivität mit geringer Effizienz einhergehen, wenn Auf-

gaben zwar lösbar sind, jedoch nur über umständliche Workarounds oder mit dauerhaft erhöhtem Konzentrationsaufwand. Die getrennte Betrachtung dieser Kriterien erlaubt es, Nutzungshürden differenziert zu analysieren und gezielt Ansatzpunkte für Verbesserungen abzuleiten. Heuristiken ergänzen diese normativen Usability-Definitionen um praxisnahe Leitlinien zur Bewertung interaktiver Systeme. Eine Heuristik stellt dabei eine erfahrungsbasierte Faustregel dar, die keine detaillierten Gestaltungsvorgaben macht, sondern typische Problemmuster sichtbar werden lässt und Orientierung bei Gestaltungsentscheidungen bietet [10]. Ihr besonderer Vorteil liegt darin, dass sie mit vergleichsweise geringem Aufwand eingesetzt werden können und auch ohne umfangreiche Nutzerstudien systematisches Feedback zur Qualität einer Benutzeroberfläche liefern. Heuristische Verfahren eignen sich daher insbesondere zur frühen Identifikation struktureller Schwächen in der Nutzerführung, etwa im Hinblick auf Verständlichkeit, Konsistenz oder Fehlertoleranz.

Zu den bekanntesten Ansätzen zählen die von Jakob Nielsen formulierten Usability-Heuristiken, die grundlegende Anforderungen an interaktive Systeme bündeln. Dazu gehören unter anderem die Sichtbarkeit des Systemstatus, eine an der Sprache der Nutzer:innen orientierte Terminologie, konsistente Bezeichnungen und Interaktionsmuster sowie die Reduktion der Oberfläche auf wesentliche Informationen [10]. Im Unterschied zu nutzerbasierten Evaluationsmethoden liefern heuristische Analysen keine direkten Aussagen über subjektive Zufriedenheit oder emotionale Wahrnehmung. Ihr Wert liegt vielmehr in der systematischen Identifikation grundlegender Gestaltungsprobleme, etwa inkonsistenter Begriffe, unklarer Rückmeldungen oder überladener Navigationsstrukturen. Gerade bei komplexen Content-Management-Systemen dienen Heuristiken häufig als erster Analyseschritt, auf dessen Basis vertiefende Untersuchungen aufbauen können [11].

Ein weiteres etabliertes Usability-Inspektionsverfahren ist der Cognitive Walkthrough, der insbesondere die Lernbarkeit einer Benutzeroberfläche aus der Perspektive neuer oder wenig erfahrener Nutzer:innen untersucht. Im Gegensatz zur heuristischen Evaluation, die eine Oberfläche eher global anhand allgemeiner Gestaltungsprinzipien bewertet, fokussiert der Cognitive Walkthrough konkrete Aufgabenabfolgen. Dabei wird geprüft, ob Nutzer:innen die notwendigen Schritte durch Exploration erkennen und erfolgreich ausführen können. Für jeden Handlungsschritt wird systematisch hinterfragt, ob das jeweilige Teilziel ersichtlich ist, die passende Aktion im Interface auffindbar erscheint und die resultierende Systemrückmeldung korrekt interpretiert werden kann. Dieses Vorgehen erlaubt es, typische Arbeitsschritte aus unterschiedlichen Erfahrungsniveaus heraus nachzuvollziehen und sichtbar zu machen, an welchen Stellen Erwartungen, Systemlogik und Interface-Gestaltung auseinanderfallen. Insbesondere Einstiegshürden, Orientierungsprobleme und Unsicherheiten im Umgang mit Systemrückmeldungen lassen sich auf diese Weise strukturiert erfassen [11].

2.1.2 User Experience Design

Der Begriff der User Experience beschreibt die Gesamtheit der Erlebnisse, Wahrnehmungen und Reaktionen, die während der Nutzung eines interaktiven Systems entstehen. Im Unterschied zur Usability, die primär auf Effektivität, Effizienz und Fehlerfreiheit abzielt, richtet sich die User Experience auf die Qualität des subjektiven Erlebens. Sie thematisiert, wie ein Produkt auf Nutzer:innen wirkt, welche Emotionen es hervorruft und in welchem Maß es als sinnvoll, unterstützend oder angenehm wahrgenommen wird [8, S. 9]. Damit rückt nicht die reine Funktionalität eines Systems, sondern dessen wahrgenommener Wert im jeweiligen Nutzungskontext in den Vordergrund. User Experience beschreibt folglich nicht nur, ob ein System genutzt werden kann, sondern wie diese Nutzung erlebt, eingeordnet und im Arbeitsalltag bewertet wird.

Während Usability vor allem die Zielerreichung unter funktionalen Gesichtspunkten betrachtet, adressiert User Experience die qualitative Dimension der Interaktion. Beide Perspektiven stehen in einem komplementären Verhältnis, sind jedoch nicht deckungsgleich. Eine hohe Ge-

brauchstauglichkeit bildet zwar eine notwendige Grundlage für positive Nutzungserlebnisse, garantiert diese jedoch nicht automatisch. Gerade in komplexen Arbeitsumgebungen können Systeme formal korrekt bedienbar sein, zugleich aber als anstrengend, unübersichtlich oder wenig unterstützend wahrgenommen werden. In solchen Fällen erfüllen sie funktionale Anforderungen, verfehlen jedoch Erwartungen der Nutzenden an Orientierung, Kontrolle und Sinnhaftigkeit. Faktoren wie wahrgenommene Verlässlichkeit, Transparenz von Systemzuständen und Klarheit der Nutzerführung beeinflussen maßgeblich die Akzeptanz eines Systems und entscheiden darüber, ob es langfristig routiniert und ohne Widerstände eingesetzt wird [12].

User Experience entsteht stets im Zusammenspiel von Mensch, Aufgabe und System und ist damit untrennbar an den jeweiligen Nutzungskontext gebunden. Sie wird nicht allein durch die visuelle Gestaltung einer Benutzeroberfläche bestimmt, sondern ebenso durch individuelle Vorerfahrungen, Erwartungen, zeitliche Rahmenbedingungen sowie organisatorische Vorgaben. Eine identische Benutzeroberfläche kann daher je nach Situation unterschiedlich erlebt werden, etwa abhängig davon, ob eine Aufgabe unter Zeitdruck erfolgt, ob sie Teil eines etablierten Arbeitsablaufs ist oder ob Nutzende erstmals mit dem System konfrontiert sind. User Experience ist folglich kein statisches Merkmal eines Produkts, sondern ein dynamisches Ergebnis situativer Nutzung, das sich im Zeitverlauf verändern kann [13].

Ziel von UX-Design ist es, diese Nutzungserlebnisse gezielt zu gestalten und möglichst vorhersehbar zu machen. Dabei werden sowohl funktionale als auch emotionale Bedürfnisse der Nutzenden berücksichtigt, etwa das Bedürfnis nach Sicherheit, Kontrolle, Effizienz oder Vertrauen [14]. UX-Design versteht Gestaltung somit nicht als rein ästhetische Optimierung, sondern als bewusste Steuerung von Wahrnehmung, Handlungssicherheit und subjektivem Erleben über den gesamten Interaktionsverlauf hinweg. Ein gelungenes UX-Design unterstützt Nutzende dabei, Aufgaben ohne unnötige Unsicherheit oder erhöhte kognitive Belastung auszuführen, und trägt zugleich zu einer stabilen, positiven Bewertung des Systems bei.

Donald Norman [15] beschreibt die emotionale Gestaltung interaktiver Produkte als Zusammenspiel mehrerer Ebenen des Erlebens. Die viszerale Ebene bezieht sich auf den unmittelbaren ersten Eindruck, der durch visuelle Eigenschaften wie Farben, Formen und Anordnung entsteht. Die Verhaltensebene adressiert die Qualität der Interaktion selbst, also ob sich Bedienung nachvollziehbar, flüssig und kontrollierbar anfühlen. Die reflektive Ebene schließlich umfasst bewusste Bewertungen und Bedeutungszuschreibungen, die sich mit der Nutzung eines Systems verbinden, etwa Zufriedenheit über effiziente Arbeitsabläufe oder Frustration bei wiederkehrenden Hindernissen. Positive Nutzungserlebnisse entstehen dann, wenn diese Ebenen konsistent aufeinander abgestimmt sind und sich gegenseitig verstärken.

Für Arbeitsumgebungen wie das TYPO3-Backend bedeutet dies, dass eine positive User Experience nicht allein durch technisch fehlerfreie Abläufe erreicht wird. Entscheidend ist vielmehr, ob das System redaktionelle Arbeitsprozesse nachvollziehbar unterstützt, Orientierung über komplexe Strukturen hinweg bietet und eine konsistente sowie vertrauenswürdige Arbeitsumgebung schafft. UX-Design betrachtet daher nicht isolierte Funktionen oder einzelne Interaktionen, sondern das Nutzungserlebnis über den gesamten Arbeitsprozess hinweg. User Experience wird damit zu einem zentralen Qualitätsmerkmal digitaler Arbeitssysteme, das Effizienz, emotionales Erleben und wahrgenommene Sinnhaftigkeit der täglichen Nutzung miteinander verbindet [16].

Die Abbildung 2.1 veranschaulicht den konzeptionellen Zusammenhang zwischen System, Usability und User Experience innerhalb eines Nutzungskontextes. Das System bildet die technische Grundlage der Interaktion und bestimmt durch seine Gestaltung die Gebrauchstauglichkeit. Usability beschreibt die Qualität der Bedienbarkeit in Bezug auf Effektivität und Effizienz. User Experience geht darüber hinaus und umfasst das gesamte Nutzungserleben, das zusätzlich durch Umweltbedingungen sowie den inneren Zustand der Nutzenden beeinflusst wird. Sie entsteht nicht allein aus der Usability eines Systems, sondern aus dem Zusammenspiel von Systemmerkmalen, Nutzungskontext und subjektiven Faktoren. Positive Nutzungserlebnisse set-

zen daher voraus, dass neben funktionalen Aspekten auch situative und menschliche Einflussgrößen berücksichtigt werden.

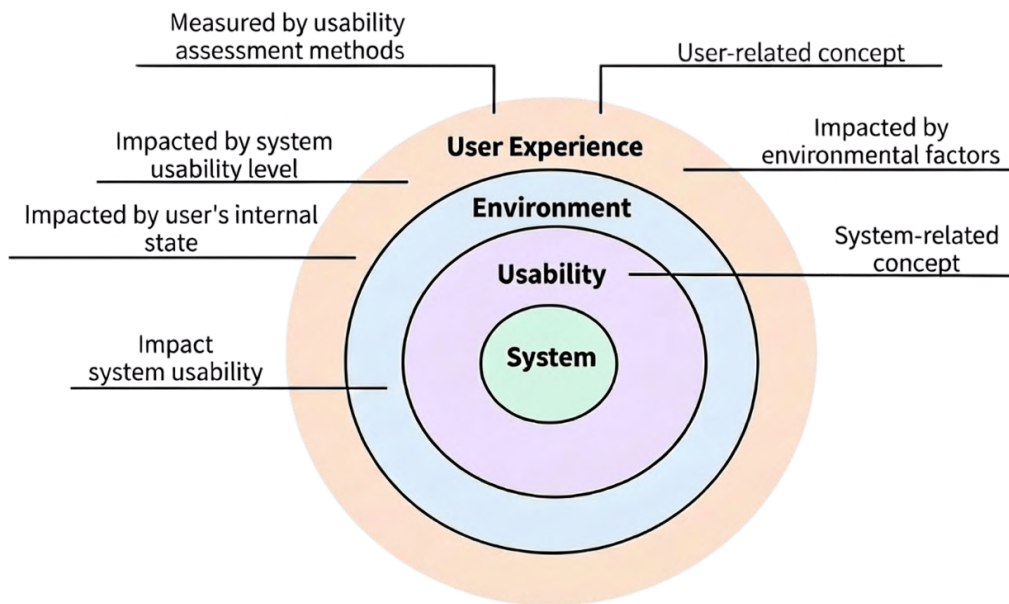


Abbildung 2.1: Konzeptioneller Zusammenhang zwischen System, Usability, Nutzungskontext und User Experience. Eigene Darstellung in Anlehnung an [1]

2.1.3 User-Centered und Human-Centered Design

User-Centered Design (UCD) bezeichnet einen Gestaltungsansatz, bei dem die Bedürfnisse, Erwartungen und Ziele der Nutzer:innen systematisch in den Mittelpunkt aller Phasen der Systementwicklung gestellt werden. Gestaltung orientiert sich dabei nicht an der internen Logik eines technischen Systems, sondern an den tatsächlichen Arbeitsaufgaben und Nutzungskontexten der Zielgruppe. Ein interaktives System gilt in diesem Verständnis nur dann als gelungen, wenn seine Struktur, Funktionen und Interaktionsabläufe die reale Nutzungspraxis unterstützen und nachvollziehbar abbilden [17]. UCD versteht Gestaltung somit als Mittel zur Ermöglichung effektiver Arbeit und nicht als Selbstzweck technischer oder ästhetischer Optimierung. Der nutzerzentrierte Gestaltungsprozess ist typischerweise iterativ aufgebaut und umfasst Phasen wie Nutzerforschung, Anforderungsdefinition, Entwurf und Evaluation. Diese Phasen werden nicht linear durchlaufen, sondern in wiederkehrenden Zyklen miteinander verknüpft. Erkenntnisse über Nutzungskontexte, Aufgabenstrukturen und Probleme der Anwender:innen werden frühzeitig erhoben, in den Gestaltungsprozess integriert und kontinuierlich überprüft. Durch dieses iterative Vorgehen können Annahmen schrittweise validiert und Fehlentwicklungen frühzeitig erkannt werden, bevor sie sich in stabilen Systemstrukturen verfestigen. Gestaltung wird damit zu einem lernenden Prozess, der sich fortlaufend an reale Nutzungssituationen anpasst [18].

Ein zentrales Merkmal von User-Centered Design ist die aktive Einbindung der Zielgruppe in den Gestaltungsprozess. Methoden wie Interviews, Beobachtungen, Kontextanalysen, Personas oder partizipative Workshops dienen dazu, implizites Wissen über Arbeitsabläufe, Routinen und Problemlagen sichtbar zu machen. Das Erfahrungswissen der Nutzer:innen wird dabei nicht als nachträgliches Feedback verstanden, sondern als konstitutiver Bestandteil der Gestaltung. Dieser Ansatz trägt dazu bei, Gestaltung nicht allein aus Expert:innenperspektive zu entwickeln, sondern unterschiedliche Nutzungsperspektiven systematisch zu berücksichtigen und

Annahmen kritisch zu hinterfragen [18].

User-Centered Design und Human-Centered Design (HCD) sind konzeptionell eng miteinander verwandt, unterscheiden sich jedoch in ihrer Reichweite. Während UCD primär auf die direkte Interaktion zwischen Nutzer:in und System fokussiert, verfolgt Human-Centered Design einen umfassenderen Ansatz [17]. HCD integriert neben nutzerzentrierten Aspekten auch arbeitswissenschaftliche, ergonomische und organisationale Perspektiven. Der Mensch wird nicht nur als Bediener:in eines Interfaces betrachtet, sondern als Teil eines sozio-technischen Systems, in dem technische, organisatorische und soziale Faktoren zusammenwirken [19, S. 66-67].

In dieser erweiterten Perspektive rückt Human-Centered Design Zielgrößen wie Zugänglichkeit, Nachhaltigkeit und langfristige Arbeitsfähigkeit in den Fokus. Die an der ISO orientierte menschenzentrierte Gestaltung fordert ein ganzheitliches Verständnis von Nutzer:innen, Aufgaben und Nutzungskontexten, die kontinuierliche Einbeziehung der Nutzenden sowie eine iterative Verfeinerung auf Basis nutzerzentrierter Evaluationen. Darüber hinaus wird die Zusammenarbeit interdisziplinärer Teams als wesentliches Gestaltungsprinzip hervorgehoben [19, S. 66-67]. Human-Centered Design bildet damit einen übergeordneten Rahmen, innerhalb dessen nutzerzentrierte Gestaltung nicht auf einzelne Interaktionen beschränkt bleibt, sondern auf die nachhaltige Gestaltung komplexer digitaler Arbeitssysteme abzielt.

2.1.4 Mensch-Computer-Interaktion

Mensch-Computer-Interaktion (MCI) ist ein interdisziplinäres Forschungs- und Anwendungsfeld, das sich mit der benutzer- und kontextgerechten Gestaltung interaktiver IT-Systeme befasst [20]. Im Zentrum steht die Frage, wie digitale Systeme so gestaltet werden können, dass sie sich an den Fähigkeiten, Erwartungen und Nutzungssituationen von Menschen orientieren und eine verständliche, effiziente sowie möglichst fehlerarme Interaktion ermöglichen [19, S. 27]. MCI verbindet dabei Erkenntnisse aus Informatik, Psychologie, Ergonomie und Arbeitswissenschaft und schafft einen theoretischen Rahmen, um die Wechselwirkungen zwischen Mensch und Technik systematisch zu analysieren.

Innerhalb der Informatik nimmt die Mensch-Computer-Interaktion eine zentrale Rolle ein, da digitale Systeme heute nahezu alle Lebens- und Arbeitsbereiche prägen [20]. Systeme, die zwar technisch funktionsfähig sind, aus Sicht der Nutzenden jedoch schwer verständlich, unübersichtlich oder fehleranfällig bleiben, können ihre Aufgaben nicht zuverlässig erfüllen und führen zu ineffizienten, fehleranfälligen oder sogar riskanten Arbeitsprozessen. Entsprechend richtet sich der Fokus der MCI nicht ausschließlich auf Oberflächenaspekte, sondern auf die Einbettung interaktiver Systeme in organisatorische, soziale und institutionelle Kontexte. Gerade in komplexen Anwendungsszenarien wie redaktionellen Backends wird deutlich, dass Interaktionsprobleme häufig weniger aus einzelnen Bedienelementen resultieren, sondern aus unklaren Systemzuständen, inkonsistenten Arbeitslogiken oder unzureichender Abstimmung zwischen System und Organisation.

Ein zentraler Bezugspunkt der Mensch-Computer-Interaktion ist die ergonomische Ausrichtung technischer Systeme an menschlichen Wahrnehmungs-, Denk- und Handlungsmöglichkeiten [19, S. 24-27]. Grundlage bilden Modelle der Informationsverarbeitung, Erkenntnisse zu Wahrnehmung, Aufmerksamkeit und Lernen sowie Analysen typischer Fehlerquellen in Nutzungssituationen [19, S. 110-112]. Diese Konzepte ermöglichen es, Interaktionsprobleme nicht nur zu beschreiben, sondern auf grundlegende kognitive und arbeitsbezogene Mechanismen zurückzuführen. Daraus lassen sich Gestaltungsziele wie Aufgabenangemessenheit, Unterstützung von Lern- und Routinisierungsprozessen, Fehlertoleranz und die Reduktion kognitiver Belastung ableiten [19, S. 26-27].

Für diese Arbeit ist MCI damit der theoretische Bezugsrahmen, um das TYPO3-Backend sowohl auf Ebene einzelner Interaktionen als auch als Arbeitssystem zu untersuchen. Der Ansatz verbindet die Analyse konkreter Nutzungshürden mit übergeordneten Qualitätszielen wie Gebrauch-

stauglichkeit, Zugänglichkeit und konsistenter Nutzerführung [20]. Mentale Modelle, kognitive Belastung und Fehlermuster werden im Folgenden als drei komplementäre Perspektiven herangezogen, um Probleme im Bedienkonzept nicht nur zu beschreiben, sondern auch begründet einzuordnen.

Mentale Modelle beschreiben die inneren Vorstellungen, mit denen Nutzende Systemfunktionen und -zustände antizipieren und ihr Handeln daran ausrichten. Sie entstehen aus Erfahrung, Konventionen und erlernten Interaktionsmustern und prägen in redaktionellen Systemen Erwartungen an Navigation, Objektbeziehungen und Rückmeldungen. Wenn Systemlogik und mentales Modell übereinstimmen, wirkt Interaktion vorhersehbar und effizient. Weichen beide voneinander ab, steigen Suchaufwand, Unsicherheit und Fehlbedienungen [21].

Dieses Risiko ist in Content-Management-Systemen mit verschachtelten Strukturen, vielen Objekttypen und uneinheitlicher Terminologie besonders ausgeprägt, weil inkonsistente Benennungen und Logiken Orientierung erschweren. Mentale Modelle sind zudem rollen- und aufgabenspezifisch: Während Autor:innen primär in Inhalten und Publikationsschritten denken, orientieren sich Administrator:innen stärker an Konfigurationen und Systemstrukturen. Ein Backend sollte diese Perspektiven durch transparente Strukturen, konsistente Begriffe und vorhersehbare Abläufe unterstützen, um Lernen zu erleichtern, Unsicherheiten zu reduzieren und stabile Routinen zu fördern.

Kognitive Belastung bezeichnet den mentalen Aufwand, der während der Bearbeitung einer Aufgabe entsteht. Sie steigt insbesondere bei hoher Informationsdichte, unklaren Entscheidungswegen oder vielen parallel sichtbaren Handlungsoptionen. In redaktionellen Systemen kann dies etwa durch mehrdeutige Icons, tiefe Menüstrukturen oder häufige Kontextwechsel verstärkt werden. Da die Kapazität des Arbeitsgedächtnisses begrenzt ist, führt hohe Belastung häufig zu sinkender Leistung, mehr Fehlern und geringerer Orientierung [21].

Gestaltung zielt deshalb darauf ab, Belastung zu begrenzen und Handlungssicherheit zu erhöhen. Progressive Offenlegung reduziert wahrgenommene Komplexität, klare Priorisierung lenkt Aufmerksamkeit, und visuelle Hierarchien strukturieren Informationen nachvollziehbar. Eindeutiges Feedback unterstützt die Interpretation von Systemzuständen, während Automatisierung bei Routineaufgaben entlasten kann und konsistente Layouts den Lernaufwand senken. Die Wirksamkeit solcher Maßnahmen lässt sich empirisch etwa über Bearbeitungszeit, Fehlerraten und Nutzerzufriedenheit erfassen, wodurch alternative Bedienkonzepte vergleichbar werden [22].

Fehler in redaktionellen Systemen lassen sich unterschiedlichen Ursachen zuordnen. Teilweise entstehen sie als einfache Ausrutscher trotz korrekter Absicht, teilweise beruhen sie auf falschen Annahmen über den aktuellen Systemzustand. In anderen Fällen werden ungeeignete Routinen angewendet oder es fehlt ein grundlegendes Verständnis der Systemlogik. Solche Fehler treten besonders häufig in komplexen Workflows mit vielen Abhängigkeiten auf, da sie Orientierung erschweren und die kognitive Belastung erhöhen. Unklare oder verspätete Rückmeldungen des Systems machen es zusätzlich schwer, Ursachen nachzuvollziehen und begünstigen weitere Fehlhandlungen.

Eine nutzerzentrierte Gestaltung zielt daher vor allem auf Prävention und sichere Korrekturmöglichkeiten. Klare Strukturen, verständliche Hinweise und begrenzende Mechanismen können kritische Fehlbedienungen reduzieren. Funktionen wie "Undo" oder Versionierung ermöglichen Korrekturen ohne dauerhafte Konsequenzen, während Rollen- und Rechtssysteme risikante Aktionen von vornherein einschränken. Schulungen allein sind dafür nicht ausreichend. Nachhaltiger sind gestalterische Maßnahmen im Interface selbst, die Fehlerwahrscheinlichkeiten senken und die Wiederherstellung unterstützen. Die systematische Analyse von Fehlern liefert somit konkrete Ansatzpunkte für Priorisierung und Redesign, um Qualität, Robustheit und Nutzungssicherheit langfristig zu verbessern [23].

2.2 TYPO3 als Content-Management-System

TYPO3 ist ein webbasiertes Open-Source-Content-Management-System, das insbesondere für komplexe, mehrsprachige und langfristig angelegte Webprojekte im professionellen Umfeld konzipiert ist. Es richtet sich an Organisationen mit hohen Anforderungen an Skalierbarkeit, Sicherheit sowie an ein differenziertes Rollen- und Rechtmanagement. Durch klar definierte Release-Zyklen und eine stabile Systembasis eignet sich TYPO3 vor allem für den nachhaltigen Betrieb umfangreicher digitaler Plattformen in institutionellen und organisatorisch geprägten Kontexten. Charakteristisch ist dabei die konsequente Trennung zwischen technischer Systemarchitektur und redaktioneller Inhaltsarbeit, die sich sowohl in der Struktur des Backends als auch in den unterstützten Arbeitsprozessen widerspiegelt [24].

In der öffentlichen Positionierung wird TYPO3 als Enterprise-CMS beschrieben, das eine robuste und sichere Infrastruktur mit hoher Anpassungsfähigkeit verbindet. Als Open-Source-System bietet TYPO3 die Möglichkeit, Funktionalitäten projektspezifisch zu erweitern und an individuelle Anforderungen anzupassen. Gleichzeitig existiert ein professionelles Ökosystem aus Agenturen, Dienstleistern und kommerziellen Angeboten, das unter anderem Wartungsverträge, Service Level Agreements und langfristige Unterstützung umfasst. Diese Kombination aus Offenheit und institutioneller Absicherung unterscheidet TYPO3 von vielen leichtgewichtigen CMS Lösungen und macht es insbesondere für Organisationen attraktiv, die auf Stabilität, Planbarkeit und langfristige Wartbarkeit angewiesen sind [23].

Aus Sicht der Mensch-Computer-Interaktion und nutzerzentrierter Gestaltung stellt TYPO3 ein besonders anspruchsvolles Untersuchungsobjekt dar. Die hohe Funktionsdichte, zahlreiche konfigurierbare Module sowie komplexe Rollen- und Workflow-Strukturen ermöglichen zwar eine flexible Abbildung organisatorischer Anforderungen, erhöhen jedoch zugleich die strukturelle Komplexität des Systems und die Anforderungen an die Nutzerführung im Backend. Für Content-Editor:innen bedeutet dies, dass sie nicht explorativ, sondern aufgabenorientiert in einem System arbeiten, das auf die effiziente Bewältigung wiederkehrender und stark strukturierter Arbeitsprozesse ausgelegt ist [25].

Die klare Trennung zwischen redaktionellen und technischen Verantwortlichkeiten prägt sowohl die Systemarchitektur als auch die tägliche Nutzung. Während technische Konfigurationen, Erweiterungen und die Systempflege in der Regel von Administrator:innen übernommen werden, konzentrieren sich Content-Editor:innen auf die Erstellung, Pflege und Veröffentlichung von Inhalten innerhalb vorgegebener Strukturen. Diese Arbeitsteilung erhöht Stabilität und Sicherheit des Systems, steigert jedoch zugleich die wahrgenommene Komplexität des Backends und stellt höhere Anforderungen an eine verständliche und konsistente Nutzerführung [26].

Die folgenden Abschnitte vertiefen diese Perspektive, indem sie die technische Architektur, die Rollen und Aufgaben sowie die zentralen Prozesse der Inhaltsverwaltung näher betrachten.

2.2.1 Technische Architektur

TYPO3 ist ein modular aufgebautes, serverseitiges Content-Management-System, das überwiegend in PHP entwickelt ist und typischerweise in einer Umgebung aus Webserver und relationalem Datenbanksystem wie MySQL oder MariaDB betrieben wird. Der Systemkern stellt grundlegende Funktionalitäten bereit und wird durch sogenannte System-Extensions ergänzt, die zentrale Dienste wie Benutzerverwaltung, Rechtekonzepte oder Inhaltsstrukturen abbilden. Darüber hinaus kann der Funktionsumfang durch zusätzliche Extensions erweitert werden, die projektspezifische Anforderungen, Integrationen externer Systeme oder individuelle Backend Module umsetzen. Diese Architektur ermöglicht es, den Core als stabile Basis zu nutzen und zugleich flexibel auf unterschiedliche Einsatzszenarien zu reagieren, ohne den Systemkern selbst verändern zu müssen. Gleichzeitig ist TYPO3 darauf ausgelegt, eine möglichst hohe Abwärtskompatibilität zwischen Versionen zu gewährleisten, um langfristige Wartbarkeit und Update-

fähigkeit sicherzustellen [25].

Innerhalb des Systems sind Datenspeicherung, Anwendungslogik und Präsentation klar voneinander getrennt. Inhalte, Konfigurationen und Metadaten werden zentral in der Datenbank verwaltet, während der File Abstraction Layer (FAL) den Zugriff auf physische Dateisysteme sowie externe Speicherorte abstrahiert. Diese Abstraktionsschicht erlaubt es, Medien unabhängig von ihrem tatsächlichen Speicherort einheitlich zu verwalten und in redaktionelle Prozesse einzubinden. Core-Komponenten und Extensions greifen über definierte Schnittstellen auf diese Ressourcen zu, wodurch Seiten, Inhaltselemente, Medien und zugehörige Metadaten konsistent verarbeitet werden können. Die klare Trennung der Systemkomponenten trägt zur technischen Stabilität bei, erhöht jedoch zugleich die strukturelle Komplexität des Gesamtsystems [27].

Das TYPO3-Backend fungiert als webbasierte Arbeits- und Administrationsoberfläche, in der redaktionelle, organisatorische und entwicklungsbezogene Aufgaben zusammengeführt werden. Es stellt unter anderem den Seitenbaum als zentrales Strukturierungsinstrument, verschiedene Modulbereiche sowie formularbasierte Oberflächen zur Pflege von Datensätzen bereit. Die zugrunde liegenden Interaktionen werden über serverseitige Controller und Templating Technologien wie Fluid umgesetzt, wodurch Backend-Funktionalitäten modular und erweiterbar gestaltet sind. Die Vielzahl an Modulen, Konfigurationsoptionen und Kontextabhängigkeiten ermöglicht eine detaillierte Anpassung an unterschiedliche Arbeitsprozesse, setzt jedoch ein hohes Maß an Systemverständnis bei den Nutzenden voraus [28].

Die Ausgabe im Frontend ist konsequent vom Backend getrennt und erfolgt auf Basis von TYPOScript und Templating-Konfigurationen, die die in der Datenbank abgelegten Inhalte in konkrete Seitenstrukturen, Layouts und Ausgabekanäle überführen. Neben klassischen Website-Ausgaben unterstützt TYPO3 auch headless und API-basierte Nutzungsszenarien, bei denen Inhalte unabhängig vom Frontend konsumiert werden können [27]. Diese architektonische Flexibilität erweitert die Einsatzmöglichkeiten des Systems erheblich, führt jedoch zu zusätzlichen Abstraktionsebenen, die sich indirekt auf die Verständlichkeit und Nachvollziehbarkeit redaktioneller Prozesse im Backend auswirken.

Aus Sicht der Mensch-Computer-Interaktion bildet die technische Architektur somit den Rahmen, innerhalb dessen redaktionelle Arbeit stattfindet. Entscheidungen auf Systemebene, etwa hinsichtlich Modularisierung, Konfigurierbarkeit oder Trennung von Backend und Frontend, beeinflussen unmittelbar die Arbeitsweise von Content-Editor:innen. Die technische Architektur ist daher nicht nur als Implementierungsdetail zu verstehen, sondern als maßgeblicher Einflussfaktor auf die wahrgenommene Komplexität, Transparenz und Bedienbarkeit des TYPO3-Backends.

2.2.2 Rollen und Aufgaben

TYPO3 verfügt über ein differenziertes Rollen- und Rechtesystem, das auf Backend-Nutzerkonten und Nutzergruppen basiert und eine feingranulare Steuerung von Zugriffsrechten ermöglicht. Rollen entstehen in der Praxis nicht durch fest definierte, systemseitig vorgegebene Profile, sondern durch die Kombination mehrerer Gruppen, denen spezifische Berechtigungen zugewiesen sind. Diese Berechtigungen legen unter anderem fest, auf welche Seitenbereiche, Sprachen, Module, Inhaltstypen und Dateien Nutzer:innen zugreifen dürfen. Offizielle Beschreibungen heben hervor, dass TYPO3 insbesondere für größere, arbeitsteilig organisierte Umgebungen konzipiert ist, in denen zahlreiche Akteur:innen parallel innerhalb einer gemeinsamen Installation arbeiten und dennoch klar abgegrenzte Verantwortlichkeiten benötigen [29].

In der Praxis haben sich typische Rollen wie Administrator:innen, Integrator:innen und Content-Editor:innen etabliert, die jeweils unterschiedliche Aufgaben und Handlungsspielräume besitzen. Administrator:innen sind in der Regel für systemweite Konfigurationen, Benutzerverwaltung und Sicherheitsaspekte zuständig, während Integrator:innen Layouts, Templates und strukturelle Vorgaben umsetzen. Content-Editor:innen konzentrieren sich hingegen auf die Erstel-

lung, Pflege und Veröffentlichung von Inhalten. Für diese redaktionellen Nutzer:innen werden meist nur jene Backend-Module sichtbar gemacht, die unmittelbar für ihre Aufgaben relevant sind. Systemnahe Konfigurationsbereiche, technische Einstellungen oder sensible Administrationsfunktionen bleiben ausgeblendet, um Fehlbedienungen zu vermeiden und die Komplexität der Oberfläche zu reduzieren.

In umfangreichen Installationen mit mehreren Websites, Organisationseinheiten oder Sprachversionen kommen zusätzliche Mechanismen wie Seiten- und Sprach-Mounts, Workspaces sowie zeitgesteuerte Veröffentlichungen zum Einsatz. Diese Funktionen ermöglichen eine präzise Aufgabenverteilung zwischen globalen und lokalen Teams, etwa indem Redakteur:innen nur auf bestimmte Seitenbäume oder Sprachversionen zugreifen können oder Inhalte zunächst in einer Vorschauumgebung bearbeitet und geprüft werden, bevor sie veröffentlicht werden [30]. Gleichzeitig führen diese Mechanismen zu zusätzlichen Abstraktionsebenen im Backend, die das Verständnis von Zuständigkeiten, Sichtbarkeiten und Veröffentlichungszuständen erschweren können, insbesondere für weniger erfahrene Nutzer:innen.

Die Ausgestaltung des Rollen- und Rechtemodells folgt dem Prinzip der Role-Based Access Control (RBAC), bei dem Berechtigungen an Aufgabenprofile gebunden werden und Verantwortlichkeiten klar delegierbar sind [29]. Aus Sicht der Mensch-Computer-Interaktion haben solche Entscheidungen unmittelbaren Einfluss auf die Nutzungserfahrung im Backend. Welche Funktionen sichtbar sind, welche Aktionen erlaubt oder eingeschränkt werden und wie Statusinformationen dargestellt sind, prägt das mentale Modell der Nutzenden sowie deren Sicherheit im Umgang mit dem System. Ein konsistent und auf reale Arbeitsaufgaben abgestimmtes Berechtigungskonzept kann die kognitive Belastung reduzieren, Orientierung fördern und Fehler vermeiden.

Umgekehrt können unklar definierte Rollen, inkonsistente Berechtigungen oder überladene Benutzeroberflächen zu Unsicherheit, Orientierungsproblemen und ineffizienten Arbeitsabläufen führen. Content-Editor:innen müssen dann zusätzliche kognitive Ressourcen darauf verwenden, den eigenen Handlungsspielraum zu verstehen oder unerwartete Systemreaktionen zu interpretieren. Das Rollen- und Rechtemodell ist damit nicht nur ein technisches Sicherheitsinstrument, sondern ein zentraler Faktor für die wahrgenommene Bedienbarkeit und die Alltagstauglichkeit des TYPO3-Backends in redaktionellen Arbeitskontexten.

2.2.3 Prozesse der Inhaltsverwaltung

Die Inhaltsverwaltung in TYPO3 ist in vielen Installationen als mehrstufiger, klar strukturierter Prozess angelegt, der den gesamten Lebenszyklus redaktioneller Inhalte abbildet. Inhalte werden typischerweise zunächst erstellt, anschließend überprüft, gegebenenfalls überarbeitet und schließlich veröffentlicht. Neben einfachen Nutzungsszenarien, in denen Inhalte unmittelbar bearbeitet und freigeschaltet werden, unterstützt TYPO3 auch formalisierte redaktionelle Workflows mit eindeutig definierten Zuständen wie Entwurf, Review und Freigabe sowie optional zeitgesteuerten Veröffentlichungen [27]. Diese Prozesslogik ermöglicht es Organisationen, redaktionelle Verantwortung zu verteilen, Qualitätssicherungsmaßnahmen zu integrieren und Veröffentlichungen kontrolliert zu steuern.

Eine zentrale Grundlage dieser Prozesse bildet die integrierte Versionierung von Inhalten. Änderungen an Seiten oder Inhaltselementen werden nicht unmittelbar wirksam, sondern als neue Versionen gespeichert, wodurch frühere Zustände erhalten bleiben und jederzeit nachvollzogen werden können. Für Content-Editor:innen bedeutet dies, dass Änderungen schrittweise vorgenommen, miteinander verglichen und bei Bedarf rückgängig gemacht werden können. Versionierung unterstützt damit nicht nur die Fehlervermeidung, sondern auch exploratives Arbeiten innerhalb sicherer Rahmenbedingungen. Insbesondere in kollaborativen Szenarien erhöht sie die Transparenz redaktioneller Entscheidungen und erleichtert die Abstimmung zwischen mehreren Beteiligten über längere Bearbeitungszeiträume hinweg.

Workspaces erweitern diese Mechanismen um eine eigenständige Staging-Ebene, die eine klare Trennung zwischen produktivem System und redaktioneller Bearbeitung ermöglicht. Inhalte werden innerhalb eines Workspaces als versionierte Datensätze angelegt, bearbeitet und organisatorisch zusammengefasst, ohne den Live-Betrieb zu beeinflussen. Änderungen durchlaufen definierte Statusstufen und können gesammelt veröffentlicht werden, wodurch komplexe Anpassungen koordiniert umgesetzt werden können [31]. Dieses Vorgehen unterstützt insbesondere größere Teams, da paralleles Arbeiten ermöglicht und gleichzeitig die Kontrolle über den Veröffentlichungszeitpunkt gewahrt bleibt.

Ergänzend dazu bietet TYPO3 umfangreiche Vorschau- und Prüfmechanismen, die eine inhaltliche Kontrolle vor der Veröffentlichung unterstützen. Inhalte können in ihrem späteren Darstellungskontext betrachtet werden, einschließlich Navigationsstrukturen, Verlinkungen und sprachspezifischer Varianten. Über teilbare Preview-Links lassen sich zudem externe Akteure wie Fachabteilungen, Lektor:innen oder Auftraggeber in den Prüfprozess einbinden, ohne ihnen direkten Zugriff auf das Backend zu gewähren. Diese Funktionen fördern kollaborative Arbeitsweisen über organisatorische Grenzen hinweg und tragen wesentlich zur Qualitätssicherung redaktioneller Inhalte bei [30].

Aus Sicht der Mensch-Computer-Interaktion stellen diese Inhaltsverwaltungsprozesse hohe Anforderungen an die Verständlichkeit und Transparenz des Backends. Content-Editor:innen müssen jederzeit erkennen können, in welchem Bearbeitungszustand sich Inhalte befinden, welche Version aktuell gültig ist und welche Schritte bis zur Veröffentlichung noch ausstehen. Unklare Statusanzeigen, inkonsistente Workflow-Logiken oder schwer auffindbare Prozessinformationen können zu Fehlbedienungen, Verzögerungen und Unsicherheit führen. Die Gestaltung der Inhaltsverwaltungsprozesse ist daher nicht nur eine organisatorische Frage, sondern eine zentrale nutzerzentrierte Gestaltungsaufgabe, die maßgeblich zur Effizienz, Fehlervermeidung und Akzeptanz des TYPO3-Backends im redaktionellen Alltag beiträgt.

2.3 Forschungsstand zur Usability des TYPO3-Backends

Die Usability des TYPO3-Backends wird bislang vor allem innerhalb der TYPO3-Community sowie in projektbezogenen Praxiskontexten thematisiert. Eine zentrale Rolle nimmt dabei das offizielle UX-Team der TYPO3-Association ein, das sich kontinuierlich mit der Analyse und Weiterentwicklung der Nutzererfahrung im Backend befasst [16]. In Community-Formaten, Workshops und offenen Diskussionsrunden werden wiederkehrende Probleme, Nutzungshürden und Verbesserungsvorschläge gesammelt, dokumentiert und in die Priorisierung zukünftiger Entwicklungen eingebracht [32].

Diese Aktivitäten liefern wertvolle Einblicke in reale Nutzungssituationen, da sie unmittelbar aus dem Arbeitsalltag von Entwickler:innen, Integrator:innen und Content-Editor:innen entstehen. Typische Schwierigkeiten werden dadurch früh sichtbar und können zeitnah adressiert werden. Gleichzeitig sind die Initiativen in erster Linie auf die konkrete Produktentwicklung ausgerichtet. Ziel ist weniger eine systematische Untersuchung von Nutzungsmustern als vielmehr die schrittweise Verbesserung bestehender Funktionen und Arbeitsabläufe.

Dieser starke Praxisbezug prägt auch die Weiterentwicklung neuer Major-Versionen wie der Version 14. Technische Entscheidungen werden eng mit Rückmeldungen aus der Nutzungspraxis abgeglichen, etwa bei der Überarbeitung von Navigationskonzepten, Modulstrukturen oder Interaktionsmustern. Die Weiterentwicklung orientiert sich damit ausdrücklich an den Bedürfnissen der Nutzenden und nicht ausschließlich an technischen Zielsetzungen. Ein expliziter methodischer oder wissenschaftlicher Rahmen bleibt jedoch meist im Hintergrund [33].

2.3.1 Praxisgetriebene UX-Verbesserungen im TYPO3-Umfeld

Die Optimierung der Usability erfolgt überwiegend innerhalb der etablierten Community- und Entwicklungsprozesse des Open-Source-Projekts. Öffentliche Diskussionsplattformen, Issue-Tracker, Workshops sowie die kontinuierliche Arbeit des UX-Teams bündeln Erfahrungen aus unterschiedlichen Projekten und führen wiederkehrende Problembereiche zusammen [16]. Gestaltungsvorschläge werden dort diskutiert, priorisiert und anschließend in Form konkreter Anpassungen im Core oder in Erweiterungen umgesetzt.

Charakteristisch für diesen Ansatz ist der enge Bezug zur realen Nutzung in konkreten Arbeitskontexten. Rückmeldungen stammen häufig von Personen, die TYPO3 täglich einsetzen und dadurch ein detailliertes Verständnis für praktische Hürden entwickeln. So werden unter anderem unklare Navigationsstrukturen, inkonsistente Modulkonzepte, schwer nachvollziehbare Systemzustände oder hohe Einstiegshürden für neue Nutzende thematisiert. Diese Beobachtungen spiegeln tatsächliche Arbeitsbedingungen wider und liefern wichtige Hinweise auf Defizite der Nutzerführung.

Die Verarbeitung dieser Erkenntnisse erfolgt überwiegend praxisnah innerhalb der bestehenden Community- und Entwicklungsprozesse, beispielsweise durch Diskussionen, Designvorschläge oder gezielte Anpassungen im Core. Die Verbesserungen entstehen somit primär aus Erfahrung und kontinuierlichem Feedback und weniger aus formalisierten Untersuchungsdesigns oder standardisierten Messverfahren.

Aus wissenschaftlicher Sicht ist diese starke Praxisorientierung ambivalent. Einerseits entstehen Lösungen, die nah an konkreten Arbeitssituationen ansetzen und reale Probleme direkt adressieren. Andererseits fehlen häufig methodische Grundlagen, vergleichbare Betrachtungen unterschiedlicher Nutzungskontexte sowie eine nachvollziehbare theoretische Einordnung. Die Initiativen liefern wichtige Impulse für die Weiterentwicklung des Systems, ersetzen jedoch keine fundierte wissenschaftliche Untersuchung des TYPO3-Backends als sozio-technisches Arbeitssystem [34].

2.3.2 Forschungsstand und Forschungslücken

Die wissenschaftliche Auseinandersetzung mit der Usability von Enterprise-Content-Management-Systemen ist insgesamt nur schwach ausgeprägt. Vorliegende Arbeiten behandeln überwiegend allgemeine Aspekte von Usability oder User Experience digitaler Informationssysteme. Die spezifischen Anforderungen redaktioneller Backends werden dabei häufig nur am Rande berücksichtigt. Content-Management-Systeme dienen oft als Beispiele für generelle Gestaltungsprinzipien, nicht jedoch als eigenständige, komplexe Arbeitssysteme mit langfristiger Nutzung und differenzierten Rollenstrukturen [35].

Besonders deutlich zeigt sich diese Lücke in der geringen Berücksichtigung der Perspektive von Content-Editor:innen. Aspekte wie die Ausbildung mentaler Modelle im Umgang mit hierarchischen Seitenstrukturen, die Entwicklung routinierter Arbeitsabläufe über längere Zeiträume oder die Bedeutung kontextabhängiger Nutzungssituationen werden in der bestehenden Literatur nur vereinzelt thematisiert. Häufig liegt der Fokus auf isolierten Interaktionen oder einzelnen Interface-Elementen, während die kontinuierliche Nutzung innerhalb arbeitsteiliger Organisationen kaum systematisch untersucht wird [36].

Hinzu kommt, dass die Nutzung von CMS-Backends stark durch organisatorische Rahmenbedingungen geprägt ist. Rollenmodelle, Berechtigungskonzepte, mehrstufige Freigabeprozesse oder institutionelle Vorgaben strukturieren den Arbeitsalltag und beeinflussen die Interaktion mit dem System maßgeblich. Werden diese Faktoren in wissenschaftlichen Analysen nicht berücksichtigt, entsteht ein abstrahiertes Bild der Nutzung, das nur eingeschränkt auf reale Arbeitssituationen übertragbar ist [37].

Insgesamt besteht daher ein klarer Bedarf an Untersuchungen, die praxisnahe Nutzungsszenarien mit etablierten Konzepten der Mensch-Computer-Interaktion verbinden und die lang-

fristige Perspektive von Content-Editor:innen systematisch einbeziehen. An dieser Stelle setzt die vorliegende Arbeit an. Sie untersucht die Nutzung des TYPO3-Backends aus redaktioneller Sicht, identifiziert typische Nutzungshürden und analysiert deren Ursachen im Zusammenspiel von Systemstruktur, Arbeitsprozessen und organisatorischen Rahmenbedingungen. Ziel ist es, belastbare Ansatzpunkte für eine nachhaltige Verbesserung der Usability abzuleiten und damit einen Beitrag zur Schließung der dargelegten Forschungslücke zu leisten [38].

Kapitel 3

Methodik

Die Methodik dieser Arbeit ist darauf ausgerichtet, die Usability und User Experience des TYPO3 Backends aus der Perspektive von Content-Editor:innen systematisch zu untersuchen und darauf aufbauend ein fundiertes Optimierungskonzept zu entwickeln. Im Zentrum steht die Frage, wie das bestehende Bedienkonzept konkrete Arbeitsabläufe unterstützt oder behindert und welche Veränderungen kurz und mittelfristig realistisch umsetzbar sind. Hierzu wird ein mehrstufiges Vorgehen gewählt, das eine heuristische Analyse des aktuellen Backends sowie einen strukturierten Vergleich mit anderen Content-Management-Systemen umfasst.

3.1 Analyse des aktuellen TYPO3-Backends

Die Ist-Analyse gliedert sich in drei thematische Schwerpunkte. Zunächst wird das Dashboard als Einstiegspunkt in das Backend untersucht. Anschließend werden Aspekte der Workflow-Effizienz betrachtet, bevor abschließend Fragen der Barrierefreiheit in den Fokus genommen werden. Die Analyse orientiert sich an den Usability-Heuristiken nach Nielsen [10], die als analytischer Rahmen dienen. Für die Untersuchung wurde mit der offiziellen TYPO3-Demoversion gearbeitet [39], die auf TYPO3 13.4 basiert und einen schnellen, vorkonfigurierten Einstieg in das Backend ermöglicht.

3.1.1 Dashboard als Einstiegspunkt in das Backend

Das Dashboard stellt den ersten Berührungspunkt von Content-Editor:innen mit dem TYPO3-Backend dar und übernimmt damit eine zentrale Orientierungsfunktion. Als Einstiegspunkt sollte es ermöglichen, den aktuellen Systemzustand schnell zu erfassen, relevante Informationen einzuordnen und ohne unnötige Zwischenschritte in die redaktionelle Arbeit überzugehen.

Im Sinne der ersten Heuristik "Visibility of System Status" [10] stellt das TYPO3-Dashboard verschiedene systembezogene Informationen bereit. Dazu zählen unter anderem Hinweise zu fehlgeschlagenen Login-Versuchen, Systemmeldungen sowie statistische Übersichten, die in Form von Zählern oder Diagrammen visualisiert werden. Diese Informationen schaffen grundsätzlich Transparenz, sind jedoch stark technisch geprägt und für Content-Editor:innen nicht immer unmittelbar handlungsrelevant.

Auch die Übereinstimmung zwischen Systemdarstellung und dem redaktionellen Arbeitsalltag ist nur eingeschränkt gegeben. Die im Standard enthaltenen Widgets fokussieren sich überwiegend auf administrative Kennzahlen, Community-News oder Systeminformationen. Typische Bezugspunkte der redaktionellen Arbeit, etwa Inhalte, Seiten oder zuletzt bearbeitete Elemente, werden hingegen kaum berücksichtigt.

Positiv hervorzuheben ist die grundsätzliche Anpassbarkeit des Dashboards. Widgets lassen sich per Drag-and-Drop frei anordnen, wodurch individuelle Arbeitsweisen gezielt unterstützt

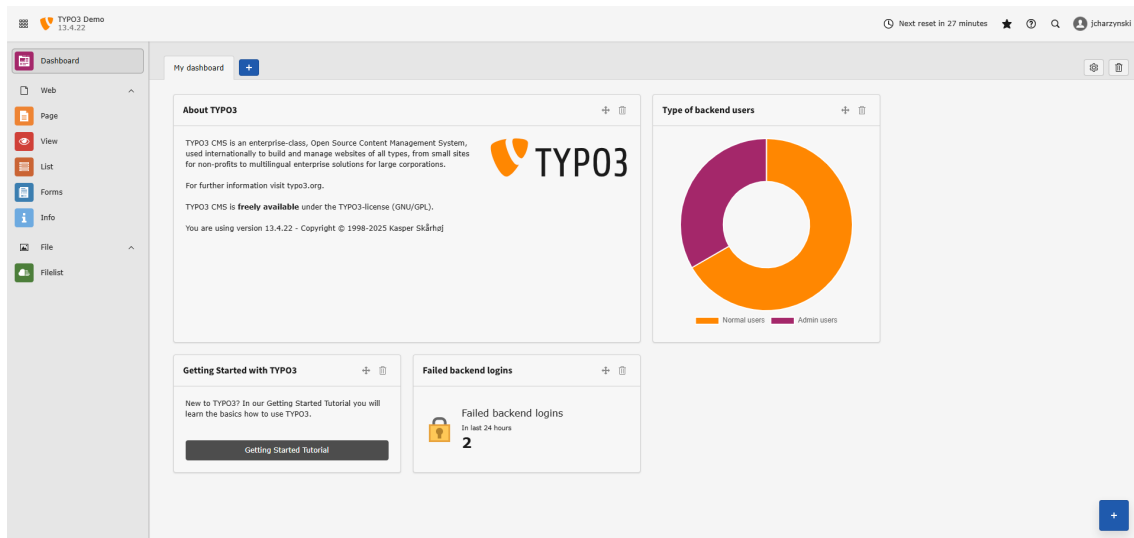


Abbildung 3.1: TYPO3 Dashboard der Demo Installation mit vorkonfigurierten Standardwidgets

werden. Diese Flexibilität bleibt jedoch begrenzt, da Größe und Darstellungsform der Widgets fest vorgegeben sind. Eine einfache Auswahl vordefinierter Größenstufen könnte die Nutzbarkeit erhöhen, ohne die Oberfläche wesentlich zu verkomplizieren.

Im Hinblick auf das Prinzip “Recognition Rather than Recall” [10] zeigt das Dashboard deutliche Defizite. Die Heuristik zielt darauf ab, Informationen und Handlungsmöglichkeiten sichtbar zu machen, sodass Nutzer:innen diese wiedererkennen können, anstatt sie aus dem Gedächtnis abrufen zu müssen. Zwar sind Informationen auf dem Dashboard präsent, es fehlen jedoch direkte Einstiege in zuletzt bearbeitete Inhalte oder häufig genutzte Seiten. Content-Editor:innen müssen ihren Arbeitskontext daher häufig über den Seitenbaum oder Suchfunktionen erneut herstellen, wodurch insbesondere bei unterbrochenen Arbeitsprozessen Effizienzverluste entstehen können.

Grundsätzlich bietet TYPO3 im Open-Source-Kontext die Möglichkeit, das Dashboard an projektspezifische Anforderungen anzupassen. Über eigene oder erweiterte Widgets können Administrator:innen redaktionelle Informationen stärker in den Fokus rücken [40]. Extensions wie *custom_dashboard_widgets* [41] zeigen, dass eine stärkere Orientierung an redaktionellen Aktivitäten technisch gut umsetzbar ist, etwa durch Übersichten zu kürzlich erstellten oder bearbeiteten Inhalten.

Aus methodischer Sicht lässt sich festhalten, dass das Dashboard seine Rolle als Einstiegspunkt klarer erfüllen könnte, wenn es weniger als allgemeine Informationsfläche und stärker als persönliches Arbeitsinstrument konzipiert wäre. Direkte Verweise auf relevante Inhalte, individuell konfigurierbare Favoriten oder einfache Organisationshilfen könnten dazu beitragen, Orientierung und produktive Nutzung enger miteinander zu verzahnen.

3.1.2 Workflow-Effizienz

Neben dem Dashboard prägen vor allem wiederkehrende Arbeitsabläufe die wahrgenommene Effizienz des TYPO3-Backends. Content-Editor:innen arbeiten im Redaktionsalltag häufig mit ähnlichen Seitenstrukturen, wiederkehrenden Inhaltselementen und festen Modulen. Workflow-Effizienz beschreibt in diesem Kontext, in welchem Maß das Backend diese Routinen unterstützt, indem Suchaufwand reduziert, Kontextwechsel minimiert und häufige Aufgaben über klar strukturierte, nachvollziehbare Schritte ermöglicht werden. TYPO3 stellt hierfür eine hohe funktionale Tiefe bereit, etwa durch modulare Inhaltsstrukturen, Versionierung, Workspaces und flexible Backend-Layouts. Diese Funktionsvielfalt geht jedoch mit einer erhöhten Komple-

xität der Benutzeroberfläche einher, die insbesondere für weniger erfahrene Nutzer:innen zu Orientierungsproblemen führen kann.

Ein exemplarischer Workflow zur Analyse der Effizienz ist das Anlegen einer neuen Rezeptseite in der TYPO3-Demoinstallation [39] sowie die anschließende inhaltliche Pflege der Seite. Der Ablauf bildet einen typischen redaktionellen Anwendungsfall ab und umfasst mehrere zentrale Interaktionsschritte.

Im ersten Schritt wird im Modul “Page” innerhalb des Seitenbaums unterhalb des Bereichs “Recipes” eine neue Unterseite erstellt (vgl. Abbildung 3.2). Dabei werden grundlegende Seiteneigenschaften wie Seitentitel, URL und Metadaten definiert. Die Erstellung der Subpage erfolgt über kontextbezogene Aktionen im Seitenbaum und erfordert einen Wechsel zwischen Struktur- und Formularansicht.

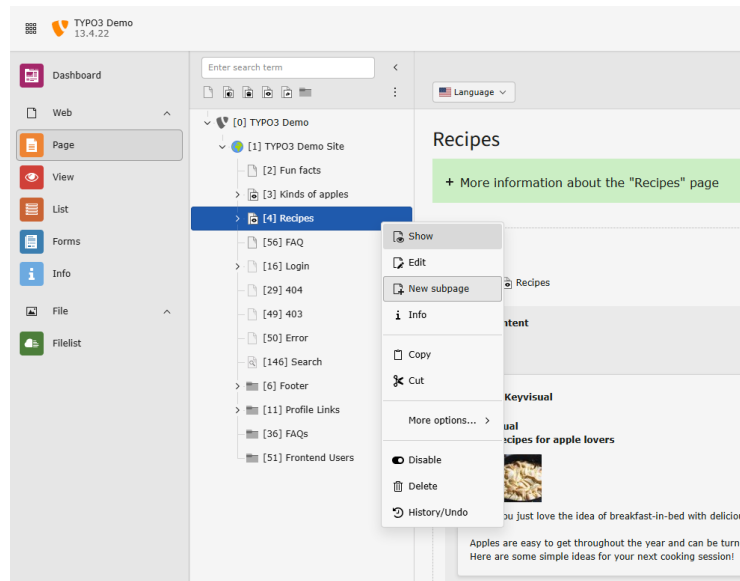


Abbildung 3.2: Anlegen einer neuen Subpage im Seitenbaum des TYPO3-Backends

Alternativ kann die neue Seite über die oberhalb des Seitenbaums angezeigten Seitentyp-Icons per Drag-and-Drop an der gewünschten Position platziert werden. Dies kann den Erstellungsprozess insbesondere für erfahrene Nutzer:innen beschleunigen.

Im Anschluss erfolgt die inhaltliche Pflege der neu angelegten Seite. Dabei werden verschiedene Inhaltsbereiche wie Hauptinhalt oder Teaser-Content ergänzt. Wiederkehrende Inhaltselemente können aus bestehenden Seiten kopiert und über die Zwischenablage an anderer Stelle eingefügt werden (vgl. Abbildung 3.3). Diese Funktion unterstützt die effiziente Wiederverwendung von Inhalten und reduziert den manuellen Pflegeaufwand, insbesondere bei strukturell ähnlichen Seiten.

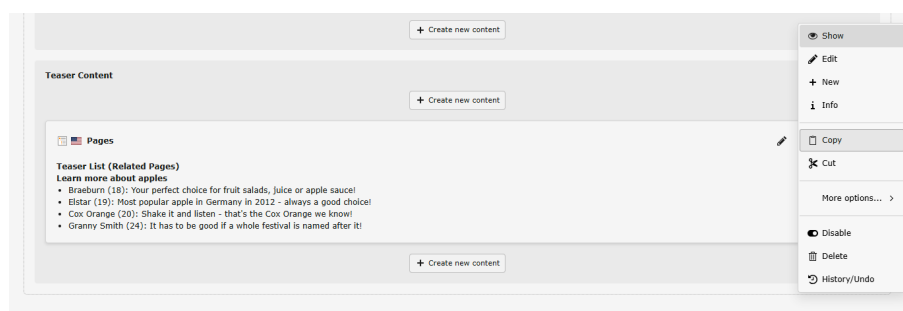


Abbildung 3.3: Kopieren eines bestehenden Inhaltselements zur Wiederverwendung

Nach Abschluss der inhaltlichen Pflege wird die Seite sichtbar geschaltet. In der Demoinstallation aktualisiert sich das Modul “Teaser List (Related Pages)” auf der zugehörigen Übersichtsseite automatisch, sodass kein zusätzlicher manueller Eingriff erforderlich ist. Aus redaktioneller Perspektive trägt diese automatische Aktualisierung wesentlich zur Workflow-Effizienz bei, da Inhalte konsistent ausgespielt werden, ohne dass weitere Konfigurationsschritte notwendig sind.

Im Rahmen der heuristischen Analyse lässt sich dieser Workflow gezielt anhand der Heuristik “Flexibility and Efficiency of Use” [10] bewerten. Betrachtet werden unter anderem die Anzahl der erforderlichen Einzelschritte, die Häufigkeit von Kontextwechseln sowie die Sichtbarkeit zentraler Aktionen. Ergänzend kann die Heuristik “Recognition Rather than Recall” [10] herangezogen werden. Diese ist erfüllt, wenn wiederkehrende Aktionen wie das Anlegen von Seiten oder das Kopieren von Inhaltselementen an erwartbaren Stellen sichtbar sind und durch konsistente Bezeichnungen unterstützt werden.

Für das TYPO3-Backend zeigt sich dabei ein gemischtes Bild. Positiv ist, dass zentrale Funktionen wie das Kopieren von Inhaltselementen, die Zwischenablage oder Drag-and-Drop-Aktionen direkt im Arbeitskontext verfügbar sind und Routineaufgaben dadurch spürbar beschleunigen. Insbesondere erfahrene Nutzer:innen profitieren von diesen Abkürzungen, da wiederkehrende Schritte ohne Umwege ausgeführt werden können. Gleichzeitig entstehen Effizienzverluste durch wiederholte Wechsel zwischen Seitenbaum, einzelnen Modulen und formularbasierten Ansichten. Diese Kontextwechsel unterbrechen den Arbeitsfluss und erhöhen die kognitive Belastung, da Nutzer:innen ihren Orientierungspunkt regelmäßig neu herstellen müssen.

Insgesamt unterstützt der Workflow typische Standardaufgaben zuverlässig, erreicht jedoch keine durchgängig flüssige Bearbeitung. Aus heuristischer Perspektive besteht daher Optimierungspotenzial insbesondere in der Reduktion von Navigationsschritten, einer stärkeren Bündelung zusammengehöriger Aktionen sowie in kontextnahen Bearbeitungsmöglichkeiten, die Wechsel zwischen unterschiedlichen Ansichten verringern.

3.1.3 Barrierefreiheit

Barrierefreiheit im digitalen Kontext bedeutet, dass digitale Angebote so gestaltet werden, dass auch Menschen mit unterschiedlichen körperlichen, sensorischen oder kognitiven Einschränkungen sie wirksam und ohne unverhältnismäßige zusätzliche Hürden nutzen können [42, S. 5]. Dabei geht es nicht nur darum, Barrieren abzubauen, sondern eine gleichberechtigte Teilhabe an digitalen Informations- und Interaktionsangeboten zu ermöglichen.

In den letzten Jahren hat Barrierefreiheit deutlich an Bedeutung gewonnen, unter anderem, weil sie zunehmend durch rechtliche Vorgaben adressiert wird. Mit den Web Content Accessibility Guidelines (WCAG) liegt ein international anerkannter Standard vor, der konkrete Anforderungen an die barrierefreie Gestaltung von Webinhalten formuliert [43]. Auf europäischer Ebene wird dieser Anspruch durch den European Accessibility Act (EAA) weiter konkretisiert, der für digitale Produkte und Dienstleistungen seit dem 28. Juni 2025 verbindliche Barrierefreiheitsanforderungen vorsieht [44].

Content-Management-Systeme nehmen in diesem Zusammenhang eine besondere Rolle ein, weil sie sowohl die technische Grundlage für barrierearme Frontends als auch den redaktionellen Arbeitsprozess prägen. Ohne geeignete Unterstützung im CMS, etwa für Alternativtexte, Überschriftenstrukturen oder Formularhinweise, lassen sich rechtliche und normative Anforderungen an Barrierefreiheit in der redaktionellen Praxis nur schwer konsistent umsetzen [28]. Barrierefreiheit und Usability stehen dabei in einem engen Verhältnis. Beide verfolgen das Ziel, Nutzungsbarrieren zu reduzieren und Interaktionen verständlich, effizient und möglichst fehlerarm zu gestalten, unterscheiden sich jedoch in ihrem Fokus. Während Usability in erster Linie die Effektivität, Effizienz und Zufriedenheit einer möglichst breiten Nutzergruppe adressiert, richtet sich Barrierefreiheit gezielt an Menschen mit unterschiedlichen Beeinträchtigungen und

stellt sicher, dass grundlegende Nutzung überhaupt möglich wird [42, S. 5].

Viele Gestaltungsprinzipien wirken zugleich als Usability- und Accessibility-Maßnahme. Dazu zählen etwa klare Informationsstrukturen, eine konsistente Navigation, verständliche Rückmeldungen des Systems oder eine gut nutzbare Tastaturnavigation. Solche Maßnahmen erleichtern nicht nur Menschen mit sensorischen oder motorischen Einschränkungen den Zugang, sondern steigern erfahrungsgemäß auch die wahrgenommene Verständlichkeit und Effizienz für Nutzer:innen ohne ausgewiesene Beeinträchtigung [45].

Im aktuellen Stand von TYPO3 wird Barrierefreiheit als Qualitätsmerkmal anerkannt und bei der Weiterentwicklung des Systems explizit berücksichtigt. Die Core-Entwicklung orientiert sich an den WCAG und arbeitet schrittweise an Verbesserungen im Backend, etwa bei der Tastaturbedienbarkeit, der semantischen Auszeichnung, der Sichtbarkeit von Fokusindikatoren sowie der Screenreader-Kompatibilität einzelner Module [46].

Fallstudien mit blinden Nutzer:innen zeigen jedoch, dass sich im praktischen Einsatz weiterhin Hürden ergeben, etwa durch fehlende oder unklare Beschriftungen, nicht angesagte Kontextwechsel oder aufwändige Navigationswege. Vor diesem Hintergrund wird Barrierefreiheit im TYPO3-Backend weniger als ein zu erreichender Zielzustand beschrieben, sondern eher als kontinuierlicher Verbesserungsprozess auf System- und Projektebene [46].

3.2 Vergleichende Analyse von Content-Management-Systemen

Die vergleichende Analyse verschiedener Content-Management-Systeme dient dem Ziel, Gestaltungsmuster, Interaktionsprinzipien und funktionale Konzepte zu identifizieren, die als Best Practices für die Optimierung des TYPO3-Backends herangezogen werden können. Durch den direkten Vergleich etablierter Systeme wie WordPress, Drupal und Storyblok lassen sich Gemeinsamkeiten und Unterschiede in der Backend-Gestaltung sowie in der Unterstützung redaktioneller Arbeitsprozesse aufzeigen. Der Fokus liegt dabei insbesondere auf Aspekten der Benutzerführung, der Informationsarchitektur und der Interaktionsgestaltung, die einen direkten Einfluss auf die Effizienz und Zufriedenheit von Content-Editor:innen haben.

3.2.1 Analyse des WordPress-Backends

WordPress zählt zu den weltweit am weitesten verbreiteten Content-Management-Systemen und dominiert den CMS-Markt sowohl hinsichtlich der Gesamtzahl der Websites als auch des Marktanteils unter CMS-basierten Installationen. Aktuelle Analysen auf Basis der W3Techs-Daten weisen WordPress mit rund 43,3 % aller erfassten Websites und einem Marktanteil von etwa 60,7 % unter allen CMS-basierten Seiten aus [47], womit es weiterhin klar führend ist. Diese starke Verbreitung lässt sich unter anderem auf eine niedrige Einstiegshürde, eine redaktionsorientierte Backend-Konzeption und ein großes Ökosystem an Erweiterungen zurückführen. Das WordPress-Backend verfolgt einen deutlich stärker auf Einfachheit und unmittelbare Nutzbarkeit ausgerichteten Ansatz als TYPO3. Der Einstieg über das Dashboard ist primär auf redaktionelle Tätigkeiten zugeschnitten und weniger von technischen oder administrativen Informationen geprägt, sodass insbesondere Gelegenheitsnutzer:innen sich schneller orientieren können.

Positiv hervorzuheben ist die modulare Struktur des Dashboards, bei der einzelne Widgets per Drag-and-Drop angeordnet oder ausgeblendet werden können. Die grundlegende Konfigurierbarkeit ähnelt dabei in ihrer Logik dem TYPO3-Dashboard. In der Standardkonfiguration stellt WordPress mehrere für Content-Editor:innen unmittelbar relevante Widgets bereit, etwa eine Übersicht kürzlich veröffentlichter Inhalte, ein Quick-Draft- bzw. Notizen-Widget sowie einfache Statistik- und Aktivitätsübersichten, die einen schnellen Einstieg in die redaktionelle Arbeit unterstützen (vgl. Abbildung 3.4).

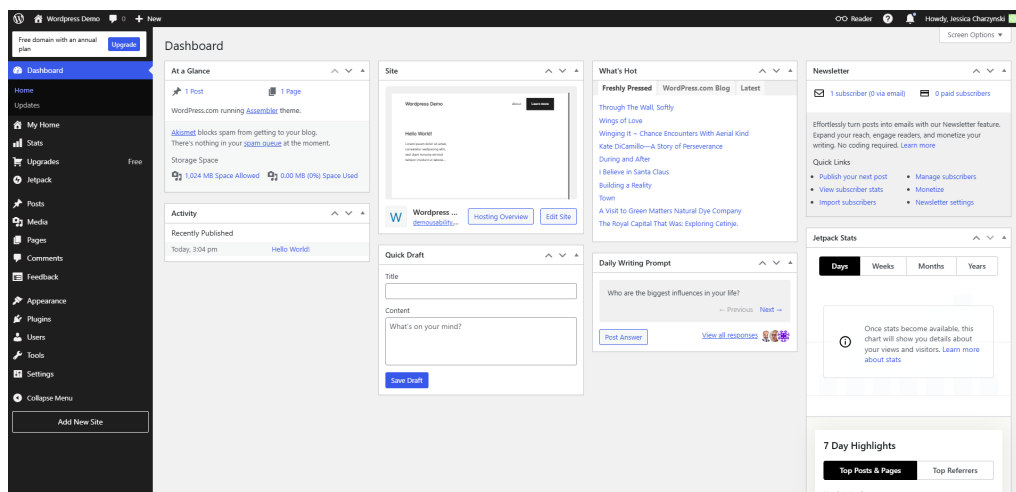


Abbildung 3.4: Dashboard des WordPress-Backends mit vorkonfigurierten Standard-Widgets

Auch die Workflow-Effizienz ist im WordPress-Backend auf lineare, gut nachvollziehbare Abläufe ausgerichtet. Zentrale Aufgaben wie das Erstellen, Bearbeiten und Veröffentlichen von Inhalten sind an wenigen, konsistent gestalteten Stellen gebündelt, wodurch sich typische Workflows mit geringem Schulungsaufwand erlernen lassen. Kontextwechsel zwischen unterschiedlichen Modulen sind selten erforderlich, da viele Aktionen innerhalb einer Ansicht durchgeführt werden können, wodurch die kognitive Belastung zusätzlich reduziert wird.

Ein wesentliches Unterscheidungsmerkmal ist die enge Verzahnung von Backend und Frontend über den blockbasierten Gutenberg-Editor. Dieser stellt Inhalte in einer WYSIWYG-orientierten Umgebung blockweise dar und bietet integrierte Vorschaufunktionen, sodass bereits während der Bearbeitung ein unmittelbarer Eindruck der späteren Darstellung im Frontend entsteht (vgl. Abbildung 3.5).

Inhalte werden damit nicht ausschließlich abstrakt in Formularfeldern gepflegt, sondern visuell strukturiert und im Kontext der Seite dargestellt, wodurch insbesondere bei layoutnahen Inhalten Entscheidungen zur Platzierung und Gewichtung von Elementen erleichtert werden.

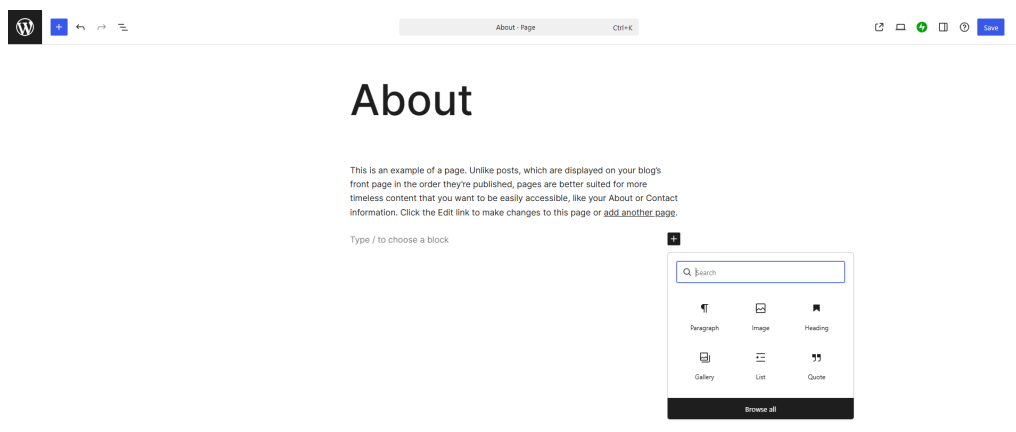


Abbildung 3.5: Visuelle Inhaltsbearbeitung mit dem blockbasierten WordPress-Editor

Zusammenfassend unterstützt WordPress redaktionelle Arbeit durch eine reduzierte Komplexität, klare Interaktionsmuster und eine starke visuelle Orientierung. Im direkten Vergleich zu TYPO3 liegt der Fokus weniger auf hochgradig konfigurierbaren Workflows und mehr auf einem niedrigschwelligen, schnell erfassbaren Nutzungskonzept, das vor allem für einfache bis mittelkomplexe Redaktionsprozesse geeignet ist.

3.2.2 Analyse des Drupal-Backends

Drupal ist ein modular aufgebautes, PHP-basiertes Content-Management-System, das vor allem in anspruchsvollen Umgebungen wie Enterprise-Projekten, öffentlichen Verwaltungen und Hochschulen eingesetzt wird. Seine Stärken liegen insbesondere in der strukturierten Modellierung von Inhalten sowie in einer sehr feingranularen Rollen- und Rechteverwaltung. Im Vergleich zu WordPress ist Drupal insgesamt weniger verbreitet, nimmt jedoch im Enterprise- und Public-Sector-Umfeld eine wichtige Rolle ein, da es hohe Anforderungen an Sicherheit, Mehrsprachigkeit und Integrationsfähigkeit zuverlässig abdeckt [48].

Das Backend folgt einem vertrauten Grundprinzip, das sich auch in TYPO3 und WordPress wiederfindet: einer globalen Navigation in Kombination mit einem zentralen Arbeitsbereich. Innerhalb des Dashboards erhalten Nutzer:innen über Elemente wie “Recent content” sowie über deutlich platzierte Shortcuts wie “Create content” und die Shortcut-Leiste schnellen Zugriff auf häufig genutzte Funktionen. Zentrale redaktionelle Aufgaben lassen sich dadurch mit wenigen Interaktionen erreichen (vgl. Abbildung 3.6).

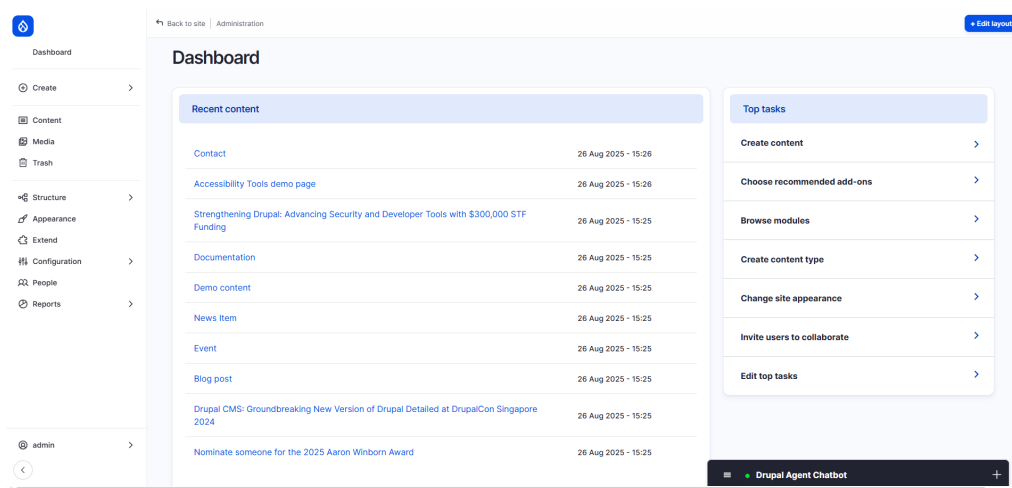


Abbildung 3.6: Dashboard-Ansicht des Drupal-Backends

Die eigentliche Inhaltsbearbeitung ist in Drupal überwiegend formularbasiert. Textfelder, Medienreferenzen und Workflow-Informationen werden in einer strukturierten Eingabemaske zusammengeführt. Seit neueren Versionen basiert der Editor auf CKEditor 5 und bietet damit eine zeitgemäße Bearbeitungserfahrung.

Zur Unterstützung der Nutzer:innen setzt Drupal häufig auf kontextuelle Hilfsysteme und Overlay-Elemente, die zusätzliche Informationen oder Funktionen direkt in der Oberfläche einblenden. In der betrachteten Demo-Installation wird jedoch deutlich, dass sich einzelne Werkzeuge, etwa ein Accessibility-Checker und ein Chatbot-Agent, gegenseitig überlagern und dabei zentrale Inhaltsbereiche verdecken (vgl. Abbildung 3.7). Statt die Arbeit zu erleichtern, entstehen so visuelle Unruhe und Irritationen, die vom eigentlichen Bearbeitungsprozess ablenken. Im Sinne der Usability-Heuristiken nach Nielsen werden damit insbesondere die Prinzipien “Aesthetic and Minimalist Design” sowie “Visibility of System Status” verletzt. Die Vielzahl gleichzeitig sichtbarer Elemente reduziert die Übersicht und erschwert es, relevante Systemrückmeldungen schnell und eindeutig wahrzunehmen [10].

Insgesamt zeigt sich Drupal als funktional starkes und vielseitig anpassbares System, das komplexe redaktionelle und organisatorische Anforderungen zuverlässig abbildet. Der Schwerpunkt liegt dabei weniger auf maximaler Vereinfachung als auf struktureller Robustheit.

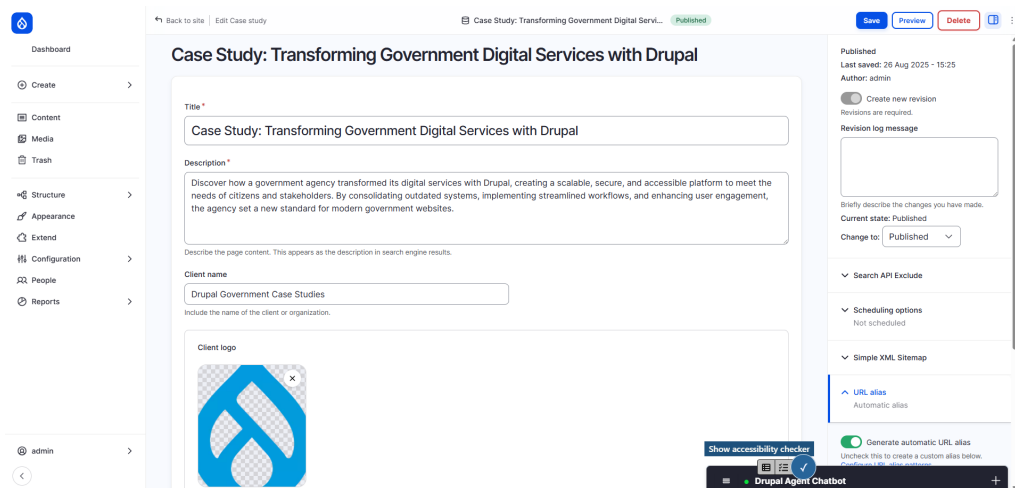


Abbildung 3.7: Inhaltsbearbeitung in Drupal mit überlagernden Assistenz-Overlays

3.2.3 Analyse des Storyblok-Backends

Storyblok ist ein headless, API-zentriertes Content-Management-System, das Inhalte komponentenbasiert modelliert und strukturiert verwaltet. Inhalte werden über standardisierte Schnittstellen bereitgestellt, während Backend und Frontend architektonisch getrennt sind. Charakteristisch ist die Verbindung dieses technischen Ansatzes mit einer visuell geprägten Arbeitsoberfläche. Ein integrierter Visual Editor erlaubt eine unmittelbare, seitennahe Bearbeitung und reduziert typische Brüche zwischen Formularlogik und tatsächlicher Darstellung [49].

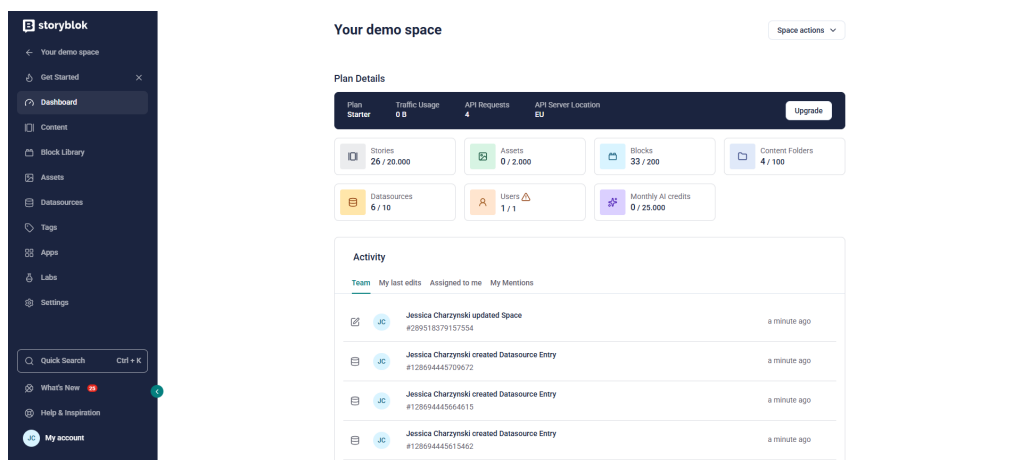


Abbildung 3.8: Storyblok-Dashboard mit Projektübersicht und Activity-Stream

Das Dashboard dient als kompakter Einstiegspunkt in das System. Es bündelt zentrale Informationen zur Projektkonfiguration, zeigt aktuelle Aktivitäten im Activity-Stream und bietet schnelle Zugriffe auf Inhalte, Spaces und weitere Verwaltungsbereiche (vgl. Abbildung 3.8). Im Vergleich zu stärker personalisierbaren Backends sind die Individualisierungsmöglichkeiten in der Standardkonfiguration eher reduziert.

Das Content-Editing ist konsequent am WYSIWYG-Prinzip ausgerichtet (vgl. Abbildung 3.9). Eine permanente Vorschau zeigt den aktuellen Seitenzustand, während Komponenten direkt im Layout oder über seitlich eingeblendete Formularfelder bearbeitet werden können. Ergänzend ermöglicht eine Layer- bzw. Strukturansicht den Zugriff auf die hierarchische Organisation der Inhalte. Gerade bei komplexeren oder verschachtelten Seiten erleichtert diese zusätzliche Perspektive die Navigation und reduziert den Suchaufwand.

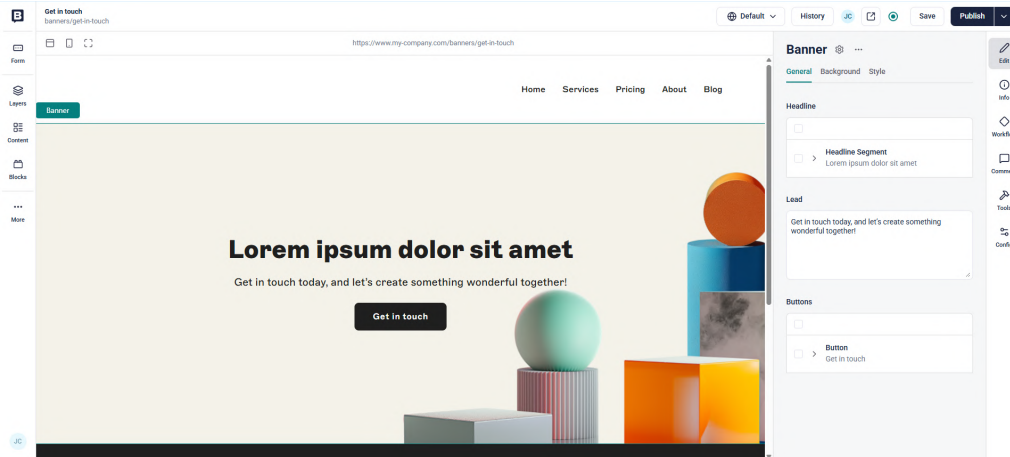


Abbildung 3.9: Inhaltsbearbeitung in Storyblok mit Visual Editor und seitlicher Formularansicht

Insgesamt zeigt sich Storyblok als System, das eine technisch flexible, API-basierte Architektur mit einer stark visuellen und niedrigschwelligen Redaktionsumgebung verbindet. Für Content-Editor:innen entsteht dadurch ein Arbeitskontext, der weniger von abstrakten Formularstrukturen und stärker von der tatsächlichen Seitendarstellung geprägt ist. Diese enge Verzahnung von Struktur und Vorschau stellt einen klaren Kontrast zu klassischen Backend-Oberflächen dar und bietet einen geeigneten Referenzpunkt für die spätere Bewertung und Einordnung alternativer Bedienkonzepte im TYPO3-Backend.

3.2.4 Ableitung übertragbarer Best Practices

Aus der vergleichenden Analyse von WordPress, Drupal und Storyblok lassen sich wiederkehrende Gestaltungsmuster erkennen, die als übertragbare Best Practices für die Weiterentwicklung des TYPO3-Backends dienen können. Trotz unterschiedlicher technischer Architekturen adressieren alle Systeme ähnliche Anforderungen des redaktionellen Alltags und zeigen dabei vergleichbare Lösungsansätze für Orientierung, Effizienz und Verständlichkeit.

Ein zentrales Muster betrifft die Informationsarchitektur. Eine klar gegliederte Oberfläche mit globaler Navigation und einem eindeutig definierten Arbeitsbereich erleichtert die Orientierung und reduziert Suchaufwand. Häufig genutzte Funktionen wie das Erstellen neuer Inhalte oder der Zugriff auf zuletzt bearbeitete Elemente sind dauerhaft sichtbar und ohne Umwege erreichbar. Aktionen werden dadurch wiedererkannt statt erinnert. Dies entspricht dem Ziel, kognitive Belastung gering zu halten und Arbeitsabläufe zu beschleunigen. Ebenso zeigt sich der Nutzen einer bewusst reduzierten visuellen Gestaltung. Aufgeräumte Dashboards unterstützen einen schnellen Einstieg. Überlagernde Hilfssysteme oder dauerhaft aktive Overlays stehen dem entgegen und verdecken zentrale Inhalte.

Darüber hinaus erweisen sich visuelle, kontextnahe Bearbeitungsumgebungen als besonders unterstützend. WYSIWYG-orientierte Editoren wie in WordPress oder Storyblok ermöglichen es, Inhalte direkt im Layout zu bearbeiten und ihre Wirkung unmittelbar nachzuvollziehen. Dies verkürzt den Abstand zwischen Eingabe und Ergebnis und unterstützt schnellere, sicherere Entscheidungen. Ergänzende strukturorientierte Ansichten, etwa Listen- oder Layer-Darstellungen, bieten zusätzliche Orientierung bei komplexen oder stark verschachtelten Inhalten und dienen als alternative Navigationswege.

Insgesamt verdeutlichen diese Beobachtungen, dass eine klare Struktur, reduzierte Oberflächenkomplexität und eine möglichst kontextnahe Bearbeitung zentrale Faktoren für effiziente redaktionelle Arbeit sind. Diese Prinzipien bilden die Grundlage für das im folgenden Kapitel entwickelte Optimierungskonzept für das TYPO3-Backend.

Kapitel 4

Optimierungskonzept

Das Optimierungskonzept knüpft unmittelbar an die Ergebnisse der heuristischen Ist-Analyse des TYPO3-Backends sowie an die Erkenntnisse aus der vergleichenden Analyse anderer Content-Management-Systeme an. Ziel dieses Kapitels ist es, die identifizierten Schwachstellen der bestehenden Nutzerführung zu bündeln, einzuordnen und in konkrete Optimierungsansätze zu überführen. Im Fokus stehen dabei jene Aspekte, die den redaktionellen Arbeitsfluss unmittelbar prägen und im Arbeitsalltag zu erhöhter kognitiver Belastung, Orientierungsverlust oder ineffizienten Umwegen führen.

Das Optimierungskonzept ist zweistufig aufgebaut und unterscheidet zwischen kurzfristig realisierbaren Verbesserungen und weitergehenden konzeptionellen Überlegungen.

Die erste Ebene bündelt sogenannte "Quick Wins". Sie umfasst Maßnahmen, die innerhalb der bestehenden TYPO3-Struktur technisch gut umsetzbar sind und vor allem die Orientierung, Verständlichkeit und Effizienz alltäglicher Arbeitsabläufe verbessern. Ziel ist es, den redaktionellen Arbeitsfluss spürbar zu entlasten, ohne tief in die Systemarchitektur einzugreifen.

Die zweite Ebene löst sich bewusst von diesen Rahmenbedingungen und untersucht, wie das Bedienkonzept des TYPO3-Backends grundsätzlich neu gedacht werden kann. Der Fokus liegt auf strukturellen Fragen der Nutzerführung und der Abbildung redaktioneller Arbeit. Diese Ebene versteht sich als konzeptionelle Perspektive für eine langfristige Weiterentwicklung und nicht als unmittelbarer Maßnahmenkatalog.

4.1 Ableitung von Optimierungspotenzialen

Ein zentrales Problemfeld zeigt sich im Dashboard als Einstiegspunkt in das Backend. Obwohl es als erster Berührungspunkt mit dem System gedacht ist, erleichtert es den Übergang in konkrete Arbeitsaufgaben nur bedingt. Die angezeigten Inhalte sind überwiegend system- und administrationsbezogen und stehen selten im direkten Zusammenhang mit dem aktuellen Arbeitskontext der Content-Editor:innen. Wichtige Bezugspunkte wie zuletzt bearbeitete Seiten, laufende Aufgaben oder anstehende Veröffentlichungen fehlen. Dadurch entsteht bereits zu Beginn zusätzlicher Orientierungsaufwand. Im Arbeitsalltag setzt sich dieses Problem fort. Das Dashboard fungiert eher als Informationsübersicht statt als Arbeitszentrale. Prioritäten, Zuständigkeiten oder inhaltliche Abhängigkeiten sind nicht klar erkennbar. Content-Editor:innen wechseln daher regelmäßig zwischen Seitenbaum, Listenansichten und externen Kommunikationskanälen, um ihre Aufgaben zu koordinieren. Eine gebündelte, aufgabenorientierte Übersicht, die Fristen und Status sichtbar macht, fehlt.

Ein weiteres Optimierungspotenzial betrifft die Unterstützung wiederkehrender redaktioneller Workflows. Die Analyse typischer Abläufe hat gezeigt, dass zentrale Aufgaben wie das Anlegen neuer Seiten, das Bearbeiten von Inhaltselementen oder das Wiederverwenden bestehender Inhalte zwar funktional vollständig abgebildet sind, jedoch häufig über mehrere Ansichten und

Module verteilt erfolgen. Diese Fragmentierung führt zu häufigen Kontextwechseln und unterbricht den Arbeitsfluss. Insbesondere für weniger routinierte Nutzer:innen erschwert dies die Bildung eines stabilen mentalen Modells der Systemlogik und erhöht die Fehleranfälligkeit bei der Ausführung alltäglicher Aufgaben.

Darüber hinaus weisen die Rückmeldungen des Systems in mehreren Situationen eine unzureichende Klarheit auf. Statusinformationen dazu, in welchem Bereich einer Seite Nutzer:innen gerade Inhalte bearbeiten oder welche Strukturkomponente betroffen ist, sind nicht durchgängig eindeutig erkennbar. Insbesondere beim Wechsel zwischen Seitenbaum, Listen- und Detailansichten ist der aktuelle Kontext der Bearbeitung nicht immer auf einen Blick sichtbar, sodass erst über Breadcrumbs, Seitentitel oder Tab-Beschriftungen rekonstruiert werden muss, wo im System gerade gearbeitet wird. In der Folge entstehen Unsicherheiten darüber, ob Änderungen an der vorgesehenen Stelle erfolgen und wie sich Bearbeitungsschritte auf die übergeordnete Seitenstruktur oder andere Ausgabekanäle auswirken.

Zusammenfassend lassen sich aus der Analyse drei übergeordnete Optimierungspotenziale ableiten: Erstens die stärkere Ausrichtung des Dashboards auf den tatsächlichen Arbeitskontext von Content-Editor:innen, zweitens die Reduktion von Komplexität innerhalb wiederkehrender Workflows durch klarere Strukturierung und Sichtbarkeit zentraler Aktionen und drittens die Verbesserung von Systemrückmeldungen, insbesondere im Hinblick auf Statusinformationen und Übergänge. Diese Potenziale bilden die Grundlage für die im Folgenden entwickelten konzeptionellen Vorschläge.

4.2 Entwurf optimierter Interface-Strukturen

Der Entwurf optimierter Interface-Strukturen zielt darauf ab, die grundsätzliche Verständlichkeit und Nachvollziehbarkeit des TYPO3-Backends zu verbessern. Im Mittelpunkt steht dabei nicht die Optimierung einzelner Funktionen, sondern die Frage, wie redaktionelle Arbeit im System räumlich, visuell und konzeptionell abgebildet wird. Interface-Struktur wird in diesem Zusammenhang als zentrales Mittel verstanden, um Orientierung zu ermöglichen und ein konsistentes mentales Modell der Systemlogik zu unterstützen.

Die Analyse hat gezeigt, dass sich viele Orientierungsschwierigkeiten aus einer stark systemorientierten Organisation des Backends ergeben. Module, Ansichten und Funktionen sind primär entlang technischer Strukturen angeordnet, während redaktionelle Aufgaben und Arbeitskontexte nur indirekt sichtbar werden. Für Content-Editor:innen bedeutet dies, dass sie sich ihren aktuellen Arbeitskontext häufig selbst erschließen müssen, etwa durch das Zusammenführen von Informationen aus Seitenbaum, Listenansichten und Detailansichten.

Das Optimierungskonzept setzt hier an und verfolgt eine nutzungsorientierte Perspektive. Zentrale Inhalte und Informationen sollen so angeordnet werden, dass sie den jeweiligen Arbeitskontext unmittelbar widerspiegeln. Dazu gehört insbesondere, dass klar erkennbar ist, auf welcher Seite, in welchem Sprachkontext und in welchem Status sich Nutzer:innen befinden. Diese Informationen bilden die Grundlage für jede redaktionelle Entscheidung und sollten daher visuell eindeutig und konsistent dargestellt werden.

Ein zentrales Prinzip des Interface-Entwurfs ist die klare Unterscheidung zwischen Orientierung, Übersicht und Bearbeitung. Nutzer:innen sollen jederzeit erkennen können, ob sie sich in einer überblicksartigen Ansicht befinden oder aktiv Inhalte bearbeiten. Gleichzeitig sollen Übergänge zwischen diesen Ebenen nachvollziehbar gestaltet sein, sodass kein Bruch im Verständnis des Arbeitskontexts entsteht. Ziel ist es, Orientierung nicht als zusätzlichen Schritt, sondern als integralen Bestandteil der Oberfläche zu begreifen.

Besondere Bedeutung kommt dem Dashboard als zentralem Einstiegspunkt in das Backend zu. Es wird nicht als neutrale Startseite verstanden, sondern als persönlicher Arbeitsraum, der die tägliche redaktionelle Arbeit strukturiert unterstützt. Ziel ist es, den aktuellen Arbeitskontext

auf einen Blick erfassbar zu machen und direkte Zugänge zu häufig benötigten Aktionen bereitzustellen. Entsprechend stehen redaktionsnahe Informationen wie zuletzt bearbeitete Seiten, offene Entwürfe oder ausstehende Aufgaben im Vordergrund, während systembezogene oder technische Inhalte bewusst zurücktreten (vgl. Abbildung 4.1).

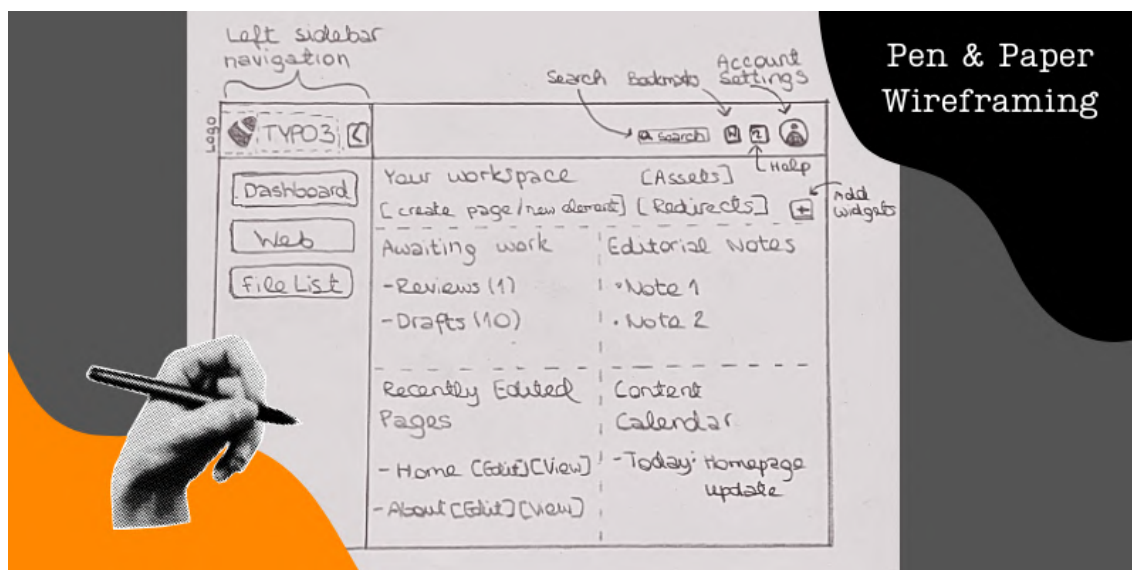


Abbildung 4.1: Pen-&-Paper-Wireframe eines TYPO3-Dashboards zur konzeptionellen Strukturierung von Layout und Informationsarchitektur

Erste Schritte im Entwurfsprozess erfolgten durchgehend in Form skizzenhafter Wireframes mit Stift und Papier, die als konzeptionelle Grundlage für den gesamten Prototyp dienten. Die Skizzen ermöglichten es, Ideen schnell festzuhalten und ohne technischen Aufwand weiterzuentwickeln. Als Low-Fidelity-Methode unterstützte dieses Vorgehen eine flexible und iterative Arbeitsweise. Varianten konnten unmittelbar ausprobiert, verworfen und angepasst werden, ohne Zeit in visuelle Details oder gestalterische Ausarbeitung zu investieren. Der Fokus lag stattdessen auf grundlegenden Fragen der Informationsarchitektur, der Navigationslogik und der Aufgabenstrukturierung. Durch die reduzierte Darstellung rückten Struktur, Inhalte und Nutzungskontext in den Vordergrund, wodurch zentrale Aspekte der Nutzerführung frühzeitig geklärt werden konnten [50].

Erst im Anschluss wurden die ausgearbeiteten Konzepte in Figma digital umgesetzt und zu einem interaktiven Prototyp weiterentwickelt. Ziel des Entwurfs ist eine klare, konsistente und arbeitsnahe Struktur des Backends, die Orientierung bietet und redaktionelle Workflows zuverlässig unterstützt.

4.3 Optimierung redaktioneller Workflows

Aufbauend auf der strukturellen Ebene der Interface-Gestaltung widmet sich dieser Abschnitt der Optimierung redaktioneller Workflows. Im Fokus stehen dabei nicht einzelne Systemfunktionen, sondern die tatsächlichen Handlungsabläufe, wie sie im redaktionellen Alltag gelebt werden. Workflows werden als Abfolge von Entscheidungen, Eingriffen und Rückmeldungen verstanden, die idealerweise als zusammenhängender Prozess wahrgenommen werden.

Die Analyse hat gezeigt, dass viele redaktionelle Tätigkeiten im TYPO3-Backend zwar funktional vollständig unterstützt werden, sich jedoch über mehrere Ansichten und Module verteilen. Tätigkeiten wie das Erstellen oder Bearbeiten von Seiten, das Überprüfen von Statusinformationen oder das Veröffentlichen von Inhalten erfordern häufige Kontextwechsel. Diese Unter-

brechungen erschweren es insbesondere weniger erfahrenen Nutzer:innen, einen stabilen Arbeitsfluss zu entwickeln.

Das Optimierungskonzept setzt daher auf eine stärkere Ausrichtung von Aktionen am jeweiligen Arbeitskontext. Zentrale Bearbeitungsschritte sollen dort angeboten werden, wo sie inhaltlich sinnvoll sind, und nicht losgelöst in globalen Menüstrukturen erscheinen. Dadurch wird deutlicher, welche Handlung sich auf welches Objekt bezieht, und der Zusammenhang zwischen Aktion und Ergebnis bleibt nachvollziehbar.

Ein weiterer Schwerpunkt liegt auf der Transparenz von Zuständen innerhalb eines Workflows. Content-Editor:innen müssen jederzeit erkennen können, ob Inhalte gespeichert, veröffentlicht oder noch in Bearbeitung sind und welche Auswirkungen einzelne Schritte haben. Unklare oder verzögerte Rückmeldungen des Systems führen zu Unsicherheit und unterbrechen den Arbeitsfluss. Das Konzept sieht daher vor, Statusinformationen konsistent sichtbar zu machen und Übergänge zwischen einzelnen Workflow-Schritten eindeutig zu kommunizieren.

Darüber hinaus berücksichtigt das Optimierungskonzept die Bedeutung wiederkehrender Routinen. Viele redaktionelle Aufgaben folgen ähnlichen Mustern, etwa beim Anlegen neuer Seiten auf Basis bestehender Strukturen oder beim Überarbeiten bereits veröffentlichter Inhalte. Diese Routinen sollen durch klar erkennbare Einstiegspunkte und nachvollziehbare Abläufe unterstützt werden, ohne erfahrene Nutzer:innen einzuschränken oder zusätzliche Komplexität zu erzeugen. Insgesamt zielt die Optimierung redaktioneller Workflows darauf ab, redaktionelle Arbeit als zusammenhängenden, verlässlichen Prozess erfahrbar zu machen. Während die strukturelle Ebene des Interface-Entwurfs Orientierung schafft, adressiert die Workflow-Ebene die konkrete Ausführung redaktioneller Aufgaben. Beide Ebenen greifen ineinander und bilden gemeinsam die Grundlage für die prototypische Umsetzung im folgenden Kapitel.

Kapitel 5

Erstellung des Prototyps

In diesem Kapitel wird beschrieben, wie das in Kapitel 4 entwickelte Optimierungskonzept in einen klickbaren Prototyp überführt wurde. Der Prototyp ist dabei nicht als fertiger Designentwurf zu verstehen, sondern als konzeptionelles Arbeitsmittel. Er dient dazu, zentrale Ideen zur verbesserten Nutzerführung und zur Unterstützung redaktioneller Arbeit anhand exemplarischer Nutzungsszenarien sichtbar und nachvollziehbar zu machen.

Der Prototyp bildet bewusst nicht das gesamte TYPO3-Backend ab. Stattdessen konzentriert er sich auf ausgewählte, typische Arbeitssituationen von Content-Editor:innen.

5.1 Methodisches Vorgehen

Die Entwicklung des Prototyps erfolgte in mehreren Schritten. Zunächst wurden grobe Ideen zur Anordnung von Navigation, Inhaltsbereichen und zentralen Aktionen in einfachen Skizzen festgehalten und anschließend in Wireframe-Layouts in Figma überführt. Auf dieser Grundlage entstanden früh erste Bildschirmwürfe, die bereits die späteren Hauptansichten mit Dashboard, Seitenübersicht und Bearbeitungsansichten in ihrer groben Struktur abbildeten.

Im nächsten Schritt wurden diese Wireframes schrittweise verfeinert und zu einem klickbaren Prototyp ausgebaut. Dabei orientierte sich die Grundstruktur zunächst eng am bestehenden TYPO3-Backend, um vertraute Elemente wie Seitenbaum und Modulnavigation zu erhalten. Ab der Ebene der inhaltlichen Bearbeitung wurden neue Interaktionsmuster integriert, etwa eine Frontend-orientierte Bearbeitung mit gleichzeitiger Formularansicht.

Der resultierende Figma-Prototyp bildet typische redaktionelle Abläufe vom Einstieg über das Dashboard bis hin zur Bearbeitung einzelner Inhaltselemente ab.

5.2 Informationsarchitektur und Oberflächenstrukturierung

Bevor einzelne Screens konkret ausgearbeitet wurden, stand die Frage im Vordergrund, welche Ansichten für den Prototyp tatsächlich notwendig sind. Ziel war es, eine überschaubare Informationsarchitektur zu schaffen, die sich an realen redaktionellen Arbeitsabläufen orientiert und typische Orientierungsprobleme adressiert.

Auf dieser Basis wurden drei zentrale Bereiche definiert: das Dashboard als Einstiegspunkt und persönlicher Arbeitsbereich, eine Seitenübersicht als vorgelagerte Orientierungsebene sowie eine Bearbeitungsansicht für die Arbeit an konkreten Inhalten.

Das Dashboard ist dabei als Weiterentwicklung innerhalb der bestehenden TYPO3-Grundstruktur zu verstehen. Es greift bekannte Konzepte auf, richtet den Fokus jedoch stärker auf den Arbeitskontext und die täglichen Aufgaben der Content-Editor:innen. Im Prototyp wird das Dashboard daher als zentraler Arbeitsraum gestaltet, der redaktionsnahe Informationen in den Vordergrund stellt (vgl. Abbildung 5.1).

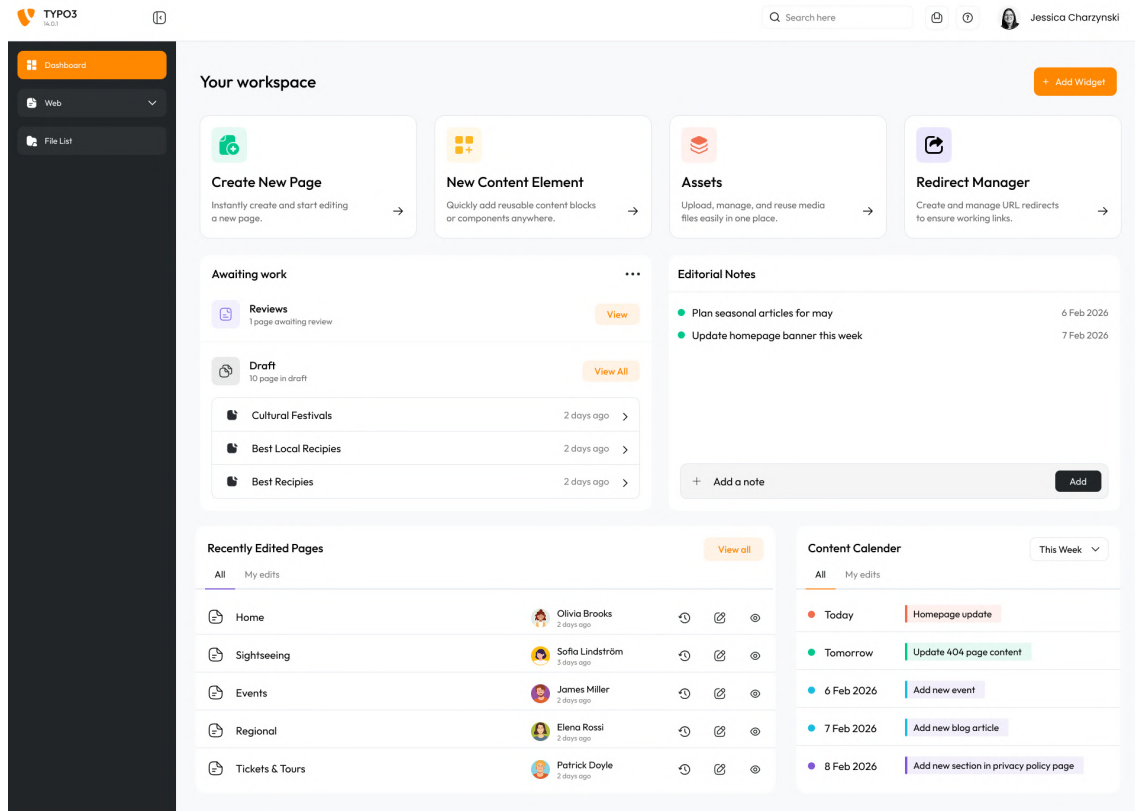


Abbildung 5.1: Dashboard-Ansicht im Figma-Prototyp

Im oberen Bereich des Dashboards stehen direkte Einstiege in zentrale Arbeitsfunktionen zur Verfügung. Darunter folgt eine Übersicht offener Aufgaben, etwa ausstehende Reviews oder Entwürfe. Zusätzlich gibt es einen separaten Bereich für persönliche Notizen. Das Dashboard zeigt außerdem zuletzt bearbeitete Seiten sowohl aus der eigenen Arbeit als auch im Teamkontext. Ein integrierter Kalender bietet einen zeitlichen Überblick über geplante Veröffentlichungen und anstehende Go-Lives.

Im Prototyp werden zwei exemplarische Seitenkonzepte gegenübergestellt. Die “Demo Page 1” (vgl. Abbildung 5.2) ist als Entwurf angelegt, der sich bewusst nah am aktuellen TYPO3-Zustand orientiert. Aufbau, Seitenstruktur und Interaktionsmuster folgen weitgehend der bestehenden Logik und ordnen sich klar in die vertraute Systemarchitektur ein. Der Ansatz setzt damit auf Kontinuität statt Bruch, reduziert Umstellungsaufwand und unterstützt die Wiedererkennung bestehender Arbeitsabläufe.

Innerhalb dieses Konzepts übernimmt die Seitenübersicht weiterhin die Rolle einer vorgelagerten Orientierungsebene vor dem eigentlichen Editieren. Sie bündelt strukturelle Informationen zur Seite und macht Seitenhierarchien sowie Sprachvarianten vergleichend sichtbar. Content-Editor:innen können dadurch schnell erfassen, wo sich Inhalte befinden und wie sie innerhalb der Struktur eingeordnet sind. Ergänzend zur bestehenden Struktur wurde die Übersicht um eine explizite Statusanzeige erweitert. Über einen klar gekennzeichneten Button wird sichtbar, ob sich eine Seite im Entwurfs- oder Veröffentlichungszustand befindet und ob die Inhalte zwischen den Sprachversionen synchron sind. Der Synchronisationsstatus zeigt, ob Übersetzungen übereinstimmen oder ob Änderungen noch nicht übertragen wurden. Dadurch wird der Bearbeitungsstand transparenter und schneller erfassbar.

Die “Demo Page 2” (vgl. Abbildung 5.3) verfolgt demgegenüber einen explorativen Ansatz. Die bisher stark formularbasierte Inhaltsbearbeitung wird hier um eine kontextbezogene, visuelle Ebene ergänzt.

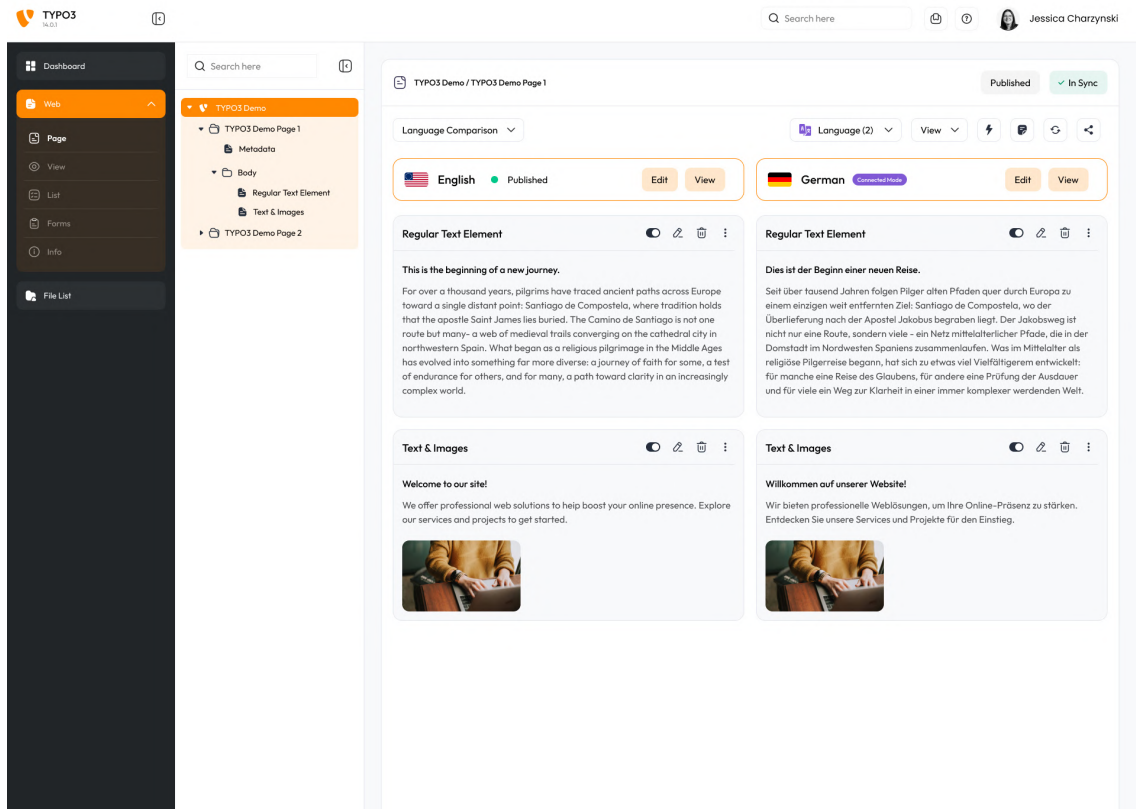


Abbildung 5.2: Seitenübersicht mit Status- und Sprachvergleich im Figma-Prototyp

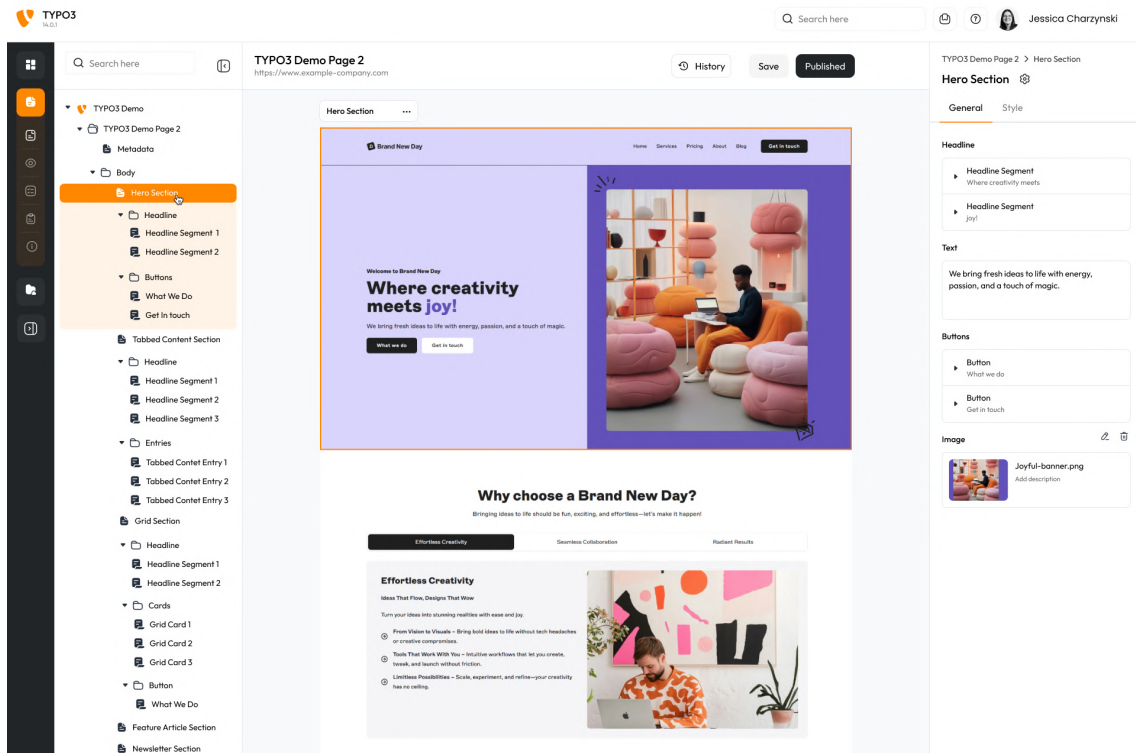


Abbildung 5.3: Bearbeitung einer Hero Section im Figma-Prototyp

Inhalte werden nicht mehr ausschließlich in abstrakten Formularen bearbeitet, sondern in direkter Beziehung zur Seitenstruktur und zur späteren Darstellung gezeigt. Dadurch wird der aktuelle Bearbeitungskontext für Content-Editor:innen deutlicher sichtbar. Aus UX- und Usability-Perspektive unterstützt dieser Ansatz insbesondere das Prinzip der Wiedererkennung gegenüber dem Erinnern (“Recognition Rather than Recall” [10]). Content-Editor:innen müssen weniger gedanklich zwischen Backend-Struktur und Frontend-Ergebnis wechseln, da Inhalt, Struktur und Wirkung enger miteinander verknüpft sind. Die Kombination aus frontendnaher Bearbeitung und weiterhin verfügbarer formularbasierter Struktur ermöglicht es, sowohl einfache inhaltliche Anpassungen als auch komplexere Änderungen effizient umzusetzen, ohne den gewohnten Systemrahmen vollständig aufzulösen.

Die linke Hauptnavigation bleibt dabei in beiden Varianten als stabiler Orientierungspunkt erhalten. Wiederkehrende Elemente erscheinen konsistent an denselben Positionen und unterstützen ein verlässliches mentales Modell der Systemstruktur, auch bei der Einführung neuer Bearbeitungskonzepte.

5.3 Visuelles Design und Interaktionskonzept in Figma

Die visuelle Gestaltung des Prototyps unterstützt den konzeptionellen Ansatz nicht nur auf ästhetischer, sondern vor allem auf funktionaler Ebene. Ziel war es, eine Oberfläche zu entwerfen, die Orientierung bietet, Zusammenhänge sichtbar macht und redaktionelle Arbeit intuitiv unterstützt. Statt eines rein neutralen Erscheinungsbildes wurde bewusst ein frischer, zeitgemäßer Look gewählt, der Orientierung fördert und zugleich eine positive Nutzungserfahrung vermittelt.

Typografisch kommt die Schriftart “Outfit” [51] zum Einsatz. Sie zeichnet sich durch eine klare, gut lesbare Formensprache aus und eignet sich besonders für strukturierte redaktionelle Oberflächen. Unterschiedliche Schriftgrößen und Abstände unterstützen die visuelle Hierarchisierung von Inhalten und erleichtern das Erfassen zentraler Informationen.

Die Farbgestaltung orientiert sich einerseits am charakteristischen Orange von TYPO3, das als Akzent für aktive Bereiche und zentrale Interaktionen eingesetzt wird. Ergänzend dazu greift der Prototyp gestalterische Impulse aus der aktuell in Entwicklung befindlichen TYPO3 Version 14 auf, etwa durch weichere Formen und den Einsatz farbiger Akzenttöne.

Das Interaktionskonzept ist eng mit der visuellen Gestaltung verknüpft. Navigationselemente, Aktionsflächen und Bearbeitungsbereiche sind klar voneinander abgegrenzt und zugleich visuell aufeinander bezogen. Übergänge zwischen Übersicht und Bearbeitung sind bewusst einfach gehalten, um den Arbeitsfluss nicht zu unterbrechen und den Fokus auf den jeweiligen Inhalt zu lenken.

Insgesamt unterstützt das visuelle Design die Zielsetzung des Prototyps, ein stärker an den Bedürfnissen von Content-Editor:innen ausgerichtetes Backend zu skizzieren. Gestaltung ist dabei kein nachgelagerter Aspekt, sondern ein zentraler Bestandteil der Nutzerführung. Sie prägt maßgeblich, wie Inhalte wahrgenommen, Zusammenhänge verstanden und Arbeitsabläufe im Backend eingeordnet werden.

Während das Dashboard und die Seitenübersicht bestehende Konzepte weiterdenken, setzt der Prototyp ab der Bearbeitungsebene bewusst neue Impulse. Die Bearbeitung von Inhalten wird nicht mehr als isolierter, formularbasierter Schritt verstanden, sondern als kontinuierlicher Prozess, der jederzeit im Kontext der Seite stattfindet.

Zentrales Gestaltungsprinzip ist dabei eine dreigeteilte Arbeitsfläche. Auf der linken Seite bleibt der Seitenbaum als vertrautes TYPO3-Element dauerhaft sichtbar. Er bietet insbesondere erfahrenen Nutzer:innen eine stabile strukturelle Orientierung und ermöglicht es, Inhalte jederzeit im Gesamtkontext der Seitenhierarchie einzuordnen. Im mittleren Bereich steht die inhaltliche Arbeit im Vordergrund. Der Prototyp zeigt exemplarisch ein Frontend-orientiertes Editing,

bei dem Inhalte direkt in ihrer visuellen Darstellung bearbeitet werden können. Am Beispiel der Hero Section wird deutlich, wie Content-Editor:innen einzelne Elemente unmittelbar in der Frontend-Vorschau auswählen und verändern können (vgl. Abbildung 5.4). Änderungen werden dort vorgenommen, wo ihre Wirkung sichtbar ist, wodurch Kontextwechsel reduziert und Unsicherheiten vermieden werden. Diese Form der Bearbeitung erleichtert insbesondere weniger erfahrenen Nutzer:innen den Zugang, da sie Inhalte nicht abstrakt über Formularfelder, sondern direkt in ihrer späteren Darstellung erfassen und verändern.

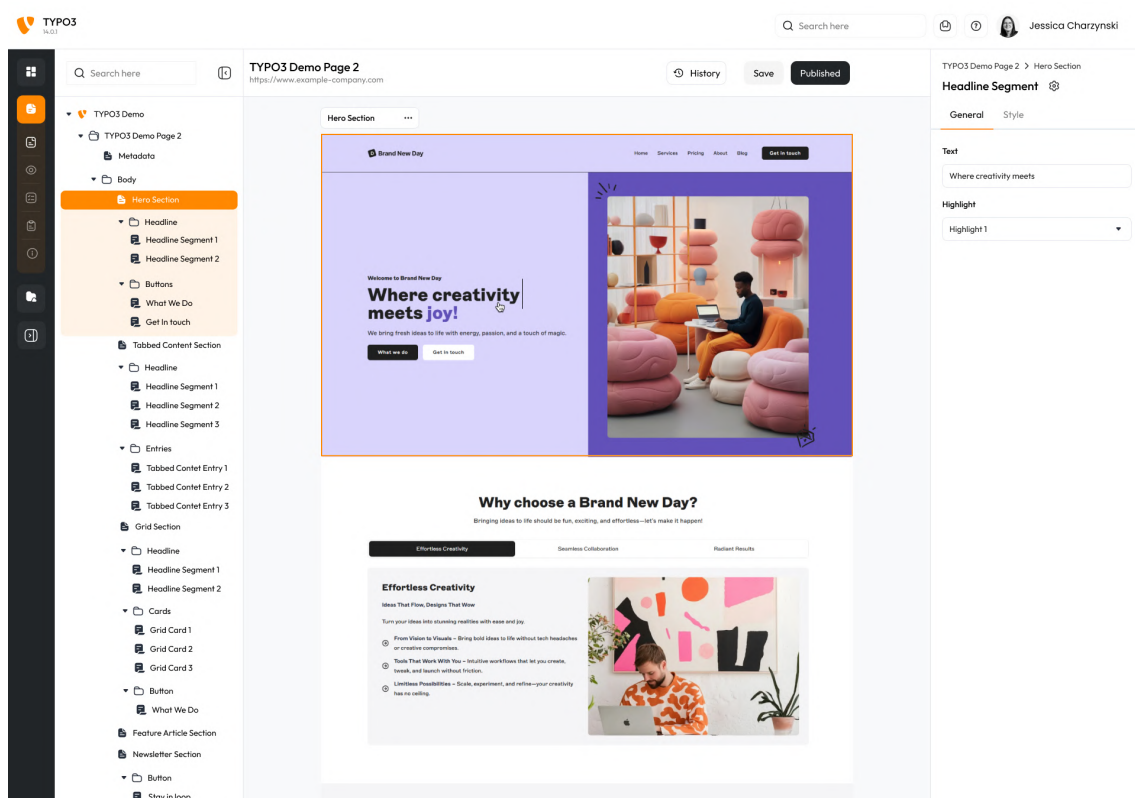


Abbildung 5.4: Frontend Editing eines Headline-Segments im Figma-Prototyp

Ergänzend dazu bleibt auf der rechten Seite eine begleitende Formularansicht sichtbar. Sie richtet sich insbesondere an erfahrene Nutzer:innen, die eine präzise Kontrolle einzelner Parameter gewohnt sind und gezielt mit strukturierten Eingabefeldern arbeiten möchten. Beide Bearbeitungsformen stehen dabei nicht in Konkurrenz, sondern ergänzen sich. Visuelles Editieren und formularbasierte Anpassungen greifen ineinander und machen unterschiedliche Arbeitsweisen gleichberechtigt möglich.

Dieses Prinzip lässt sich auch auf komplexer aufgebaute und stärker verschachtelte Inhaltsmodule übertragen. Gerade bei Modulen, die aus mehreren Ebenen bestehen und unterschiedliche Inhalte bündeln, ist eine klare Orientierung während der Bearbeitung entscheidend. Der Prototyp zeigt, dass auch solche Strukturen nachvollziehbar und kontrollierbar bearbeitet werden können, ohne dass der Überblick verloren geht.

Content-Editor:innen können dabei auf zwei gleichwertige Navigationswege zurückgreifen. Einerseits bleibt die strukturierte Navigation über den Seitenbaum und die modulare Inhaltsstruktur auf der linken Seite erhalten. Sie ermöglicht insbesondere bei umfangreichen Seiten ein gezieltes Ansteuern einzelner Bereiche und bietet eine verlässliche, hierarchische Orientierung innerhalb der Sitemap.

Ergänzend dazu erlaubt der Prototyp eine direkte Navigation über die Frontend-orientierte Vorschau. Einzelne Inhaltsbereiche und Module lassen sich unmittelbar in der Seitenvorschau anklicken. Der aktuell aktive Bereich wird dabei visuell durch einen orangefarbenen Rahmen hervor-

gehoben, sodass jederzeit eindeutig erkennbar ist, welches Modul oder welche Sektion gerade bearbeitet wird. Diese visuelle Markierung reduziert Unsicherheiten und unterstützt insbesondere bei verschachtelten Strukturen die räumliche und inhaltliche Zuordnung innerhalb der Seite. Diese Kopplung von Sitemap, Vorschau und Modulbearbeitung unterstützt insbesondere bei komplexeren Seitenstrukturen die Orientierung und verhindert, dass Content-Editor:innen den Bearbeitungskontext aus den Augen verlieren.

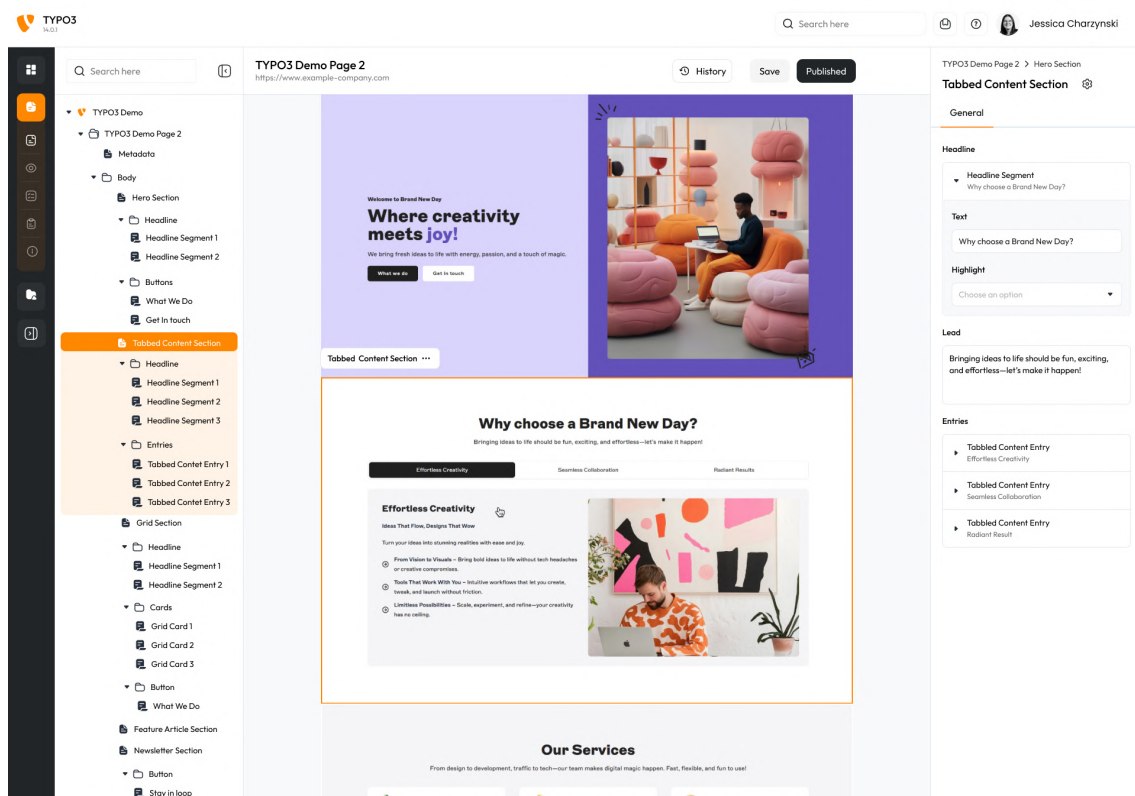


Abbildung 5.5: Tabbed-Content-Sektion im Figma-Prototyp

Die Tabbed-Content-Sektion zeigt exemplarisch, wie mehrere inhaltliche Einheiten innerhalb eines einzelnen Moduls übersichtlich strukturiert und effizient bearbeitet werden können (vgl. Abbildung 5.5). Tabs fungieren dabei als zusätzliche Gliederungsebene und bündeln thematisch zusammengehörige Inhalte, ohne den Arbeitsbereich zu überladen. Redakteur:innen können zwischen den Bereichen wechseln, ohne den Kontext zu verlieren. Die Detailbearbeitung einzelner Einträge erfolgt direkt im Seitenkontext (vgl. Abbildung 5.6). Der jeweils aktive Bereich wird sowohl in der Frontend-Vorschau als auch im Seitenbaum klar hervorgehoben. Dadurch bleibt selbst bei verschachtelten Strukturen jederzeit nachvollziehbar, welches Modul aktuell aktiv ist und in welchem Bereich gearbeitet wird.

Die Grid-Sektion zeigt ein layoutgebendes Modul, in dem Child-Items (im Beispiel "Cards") in einem klar definierten Raster angeordnet werden können (vgl. Abbildung 5.7). Neue Elemente lassen sich flexibel ergänzen und innerhalb der Struktur neu positionieren, ohne den Zusammenhang der Seite zu unterbrechen. Der Bearbeitungskontext bleibt dabei durchgehend erhalten, wodurch Anpassungen direkt im Layout vorgenommen und strukturelle Entscheidungen unmittelbar überprüft werden können.

Insgesamt macht der Prototyp deutlich, dass das entwickelte visuelle und interaktive Konzept sowohl für einfache als auch für komplexe Inhaltsstrukturen tragfähig ist. Die enge Verzahnung von struktureller Navigation, Frontend-orientierter Vorschau und visueller Rückmeldung ermöglicht eine konsistente und nachvollziehbare Bearbeitung, die unterschiedliche Arbeitsweisen unterstützt und auch bei umfangreichen Seiten Orientierung bietet.

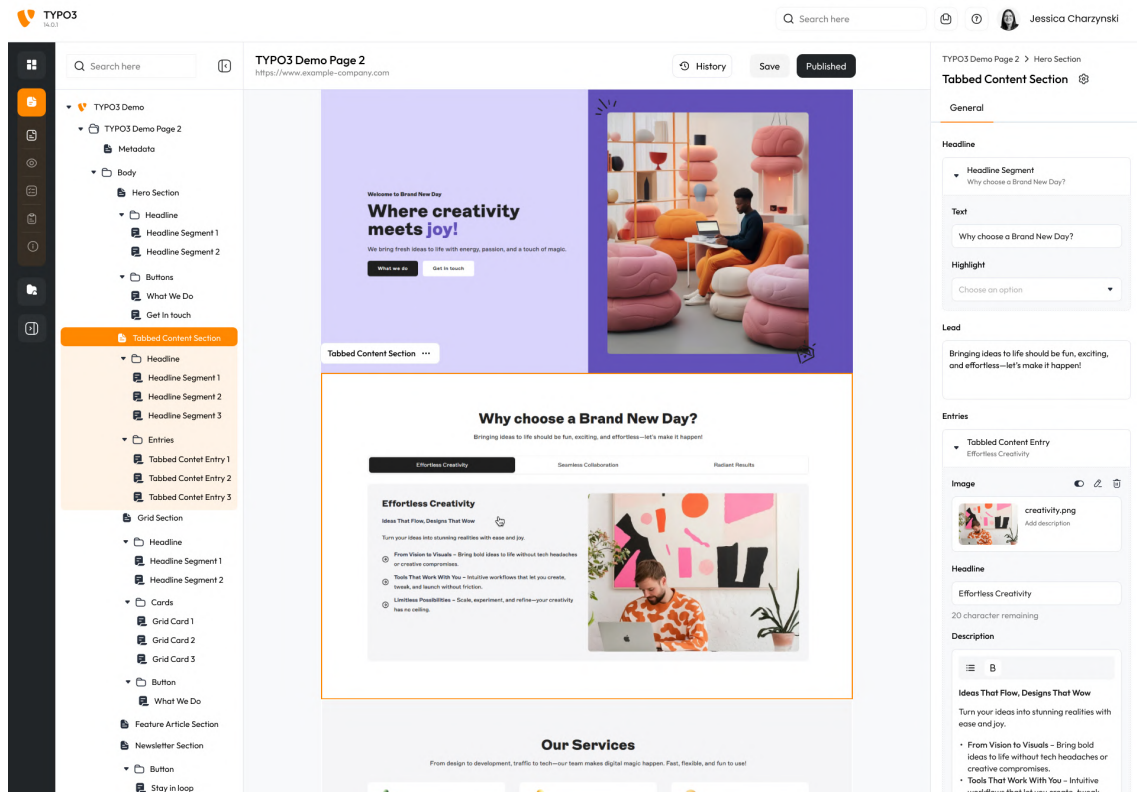


Abbildung 5.6: Detailbearbeitung eines Tabbed-Content-Eintrags im Figma-Prototyp

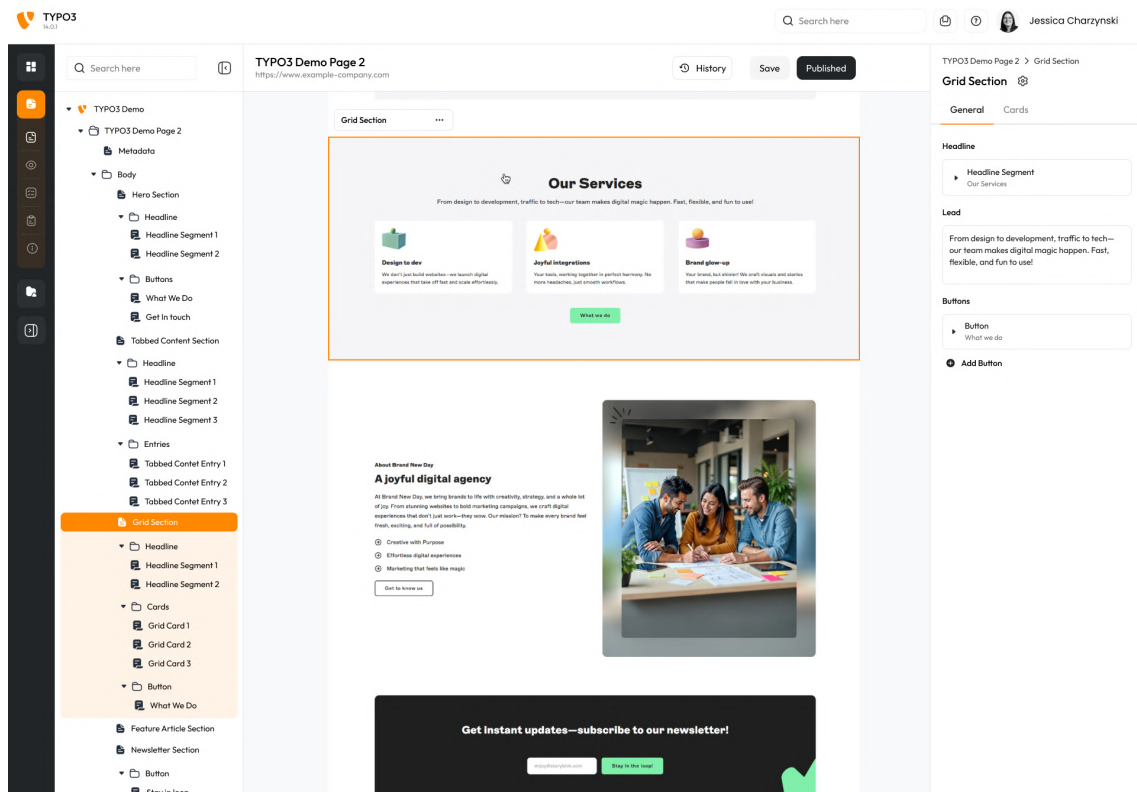


Abbildung 5.7: Grid-Sektion mit Card-Elementen im Figma-Prototyp

5.4 Validierung der Prototyp-Funktionalitäten

Die Überprüfung des Prototyps erfolgte in Form einer eigenständigen, reflektierenden Durchsicht. Anhand des klickbaren Prototyps wurde geprüft, ob die entwickelten Strukturen und Interaktionen die in den vorherigen Kapiteln beschriebenen Problemstellungen adressieren und redaktionelle Arbeit verständlicher und konsistenter unterstützen.

Hierzu wurde der Prototyp mehrfach von Anfang bis Ende durchgeklickt. Dabei wurden typische Nutzungsszenarien durchgespielt, etwa der Einstieg über das Dashboard, der Wechsel in die Seitenübersicht sowie die anschließende Bearbeitung einzelner Inhaltselemente. Gleichzeitig wurde geprüft, ob es jederzeit möglich ist, aus einzelnen Bearbeitungsschritten wieder in übergeordnete Ebenen zurückzukehren.

Eine zentrale Rolle spielte dabei die Interaktivität des Prototyps. Die angelegten Klickpfade machen sichtbar, wie Nutzer:innen zwischen Übersicht, Auswahl und Bearbeitung wechseln und wie diese Ebenen logisch miteinander verknüpft sind (vgl. Abbildung 5.8). Der Prototyp ermöglicht sowohl lineare Arbeitsabläufe als auch ein flexibles Zurückspringen zu vorherigen Ansichten. Gerade bei komplexeren redaktionellen Szenarien entsteht so ein stimmiger Eindruck typischer Arbeitsweisen, wodurch sich der Prototyp weniger wie eine statische Demo und eher wie ein reales Arbeitssystem anfühlt.

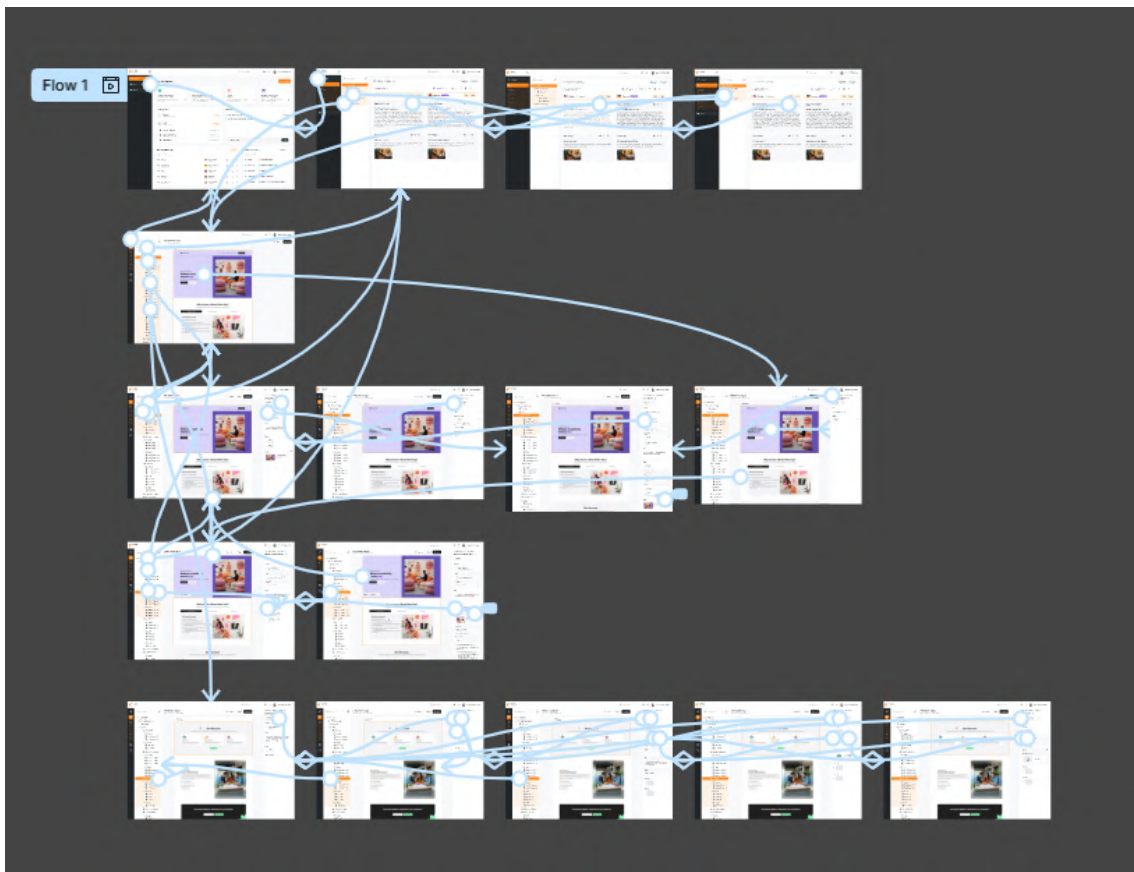


Abbildung 5.8: Interaktive Klickpfade zwischen Bearbeitungsansichten im Figma-Prototyp

Im Zuge dieser Durchläufe wurden kleinere Iterationsschleifen vorgenommen. Fehlende Interaktionen wurden ergänzt, bestehende Abläufe präzisiert und einzelne Übergänge angepasst, um die Nachvollziehbarkeit der Workflows weiter zu verbessern.

Die Ergebnisse dieser Überprüfung sichern die Funktionsfähigkeit des Prototyps und bilden die Grundlage für die anschließende Evaluation.

Kapitel 6

Evaluation

Ziel der Evaluation war es, den entwickelten Prototyp aus unterschiedlichen fachlichen Blickwinkeln zu prüfen und seine Verständlichkeit, Praxistauglichkeit sowie technische Umsetzbarkeit einzuschätzen. Im Mittelpunkt stand dabei keine messende Leistungsbewertung, sondern eine qualitative Betrachtung, die reale Arbeitssituationen, Routinen und Anforderungen der beteiligten Rollen berücksichtigt.

Statt klassischer Usability-Tests wurde bewusst ein exploratives Vorgehen gewählt. Der Prototyp diente dabei als gemeinsame Diskussions- und Reflexionsgrundlage, um Gestaltungsansätze frühzeitig zu prüfen und Stärken sowie Schwächen aus unterschiedlichen Blickwinkeln sichtbar zu machen. So konnten konzeptionelle Ideen unabhängig von einem konkreten Release-Rahmen eingeordnet werden. Die Auswertung erfolgt entlang dreier Perspektiven, die unterschiedliche fachliche Blickwinkel auf das TYPO3-Backend abbilden: eine UX-orientierte Gestaltungssicht, die redaktionelle Praxis sowie Entwicklung und Systemarchitektur. Diese Struktur erlaubt es, Anforderungen systematisch zu vergleichen und Überschneidungen ebenso wie Spannungen zwischen den Perspektiven herauszuarbeiten.

6.1 Vorgehen und Aufbau der Evaluation

Das Feedback zum Prototyp wurde über mehrere Kanäle eingeholt. Zum einen fand ein offener Austausch im #cig-ux-Slack-Channel der TYPO3-Community statt, zum anderen wurden Rückmeldungen im erweiterten Fachkreis gesammelt und strukturiert in einem Miro-Board dokumentiert (vgl. Abbildung B.1 und Abbildung B.2). Auf diese Weise konnten sowohl spontane Erstdrucke als auch ausführlichere und begründete Kommentare erfasst werden.

Das Miro-Board diente dabei als zentrales Sammel- und Strukturierungsinstrument. Nach einer kurzen Einführung in Zielsetzung, Kontext und Nutzung des Figma-Prototyps hielten die Teilnehmenden ihre Beobachtungen in Form digitaler Notizzettel fest. Diese wurden anschließend thematisch gebündelt und verschiedenen Perspektiven zugeordnet: allgemeinen Eindrücken, UX-Aspekten, redaktionellen Anforderungen sowie Entwicklungs- und Systemfragen. Die visuelle Sammlung erleichterte es, wiederkehrende Muster zu erkennen, Aussagen zu vergleichen und Schwerpunkte im Feedback herauszuarbeiten.

An der Evaluation beteiligten sich Personen mit unterschiedlichen fachlichen Hintergründen. Neben Mitgliedern der TYPO3-Community mit UX- und Integrationsschwerpunkt nahmen mehrere Content-Editorinnen aus dem beruflichen Umfeld der Autorin sowie Personen mit informatischem beziehungsweise technischem Hintergrund teil. Ergänzend wurde ein Experteninterview mit Laura Heine geführt, die das Backend-Redesign von TYPO3 Version 14 koordiniert. Ihr konzeptioneller Ansatz zur Weiterentwicklung des Systems [52] bildete einen zusätzlichen fachlichen Bezugspunkt. Das Interview ist im Anhang dokumentiert (vgl. Kapitel C) und wurde bei der Einordnung und Interpretation der Evaluationsergebnisse berücksichtigt.

Die Evaluation ist somit als qualitative, explorative Stichprobe zu verstehen. Ziel war kein repräsentatives Urteil, sondern ein differenziertes Stimmungsbild aus realen Nutzungskontexten. Die Kombination aus offenen Diskussionen, strukturiertem Feedback im Miro-Board und dem Experteninterview ermöglichte es, sowohl konkrete Detailprobleme als auch übergreifende Gestaltungstendenzen nachvollziehbar einzuordnen.

Neben ausführlichen Kommentaren zeigten sich im Slack-Austausch zudem mehrere spontane positive Reaktionen. Diese kurzen Beiträge oder Emoji-Rückmeldungen stellen zwar keine belastbare quantitative Messung dar, deuten jedoch auf eine grundsätzlich positive Aufnahme des Konzepts innerhalb der TYPO3-Community hin.

6.2 Evaluation aus UX-Perspektive

Aus UX-Sicht zeigte sich insgesamt ein vielschichtiges Bild. Die Rückmeldungen bezogen sich weniger auf einzelne Funktionen oder visuelle Details als auf das allgemeine Nutzungserlebnis, insbesondere auf Orientierung, Übersicht und das Gefühl, sicher durch die Oberfläche geführt zu werden.

Mehrere Teilnehmende beschrieben, dass Zusammenhänge zwischen Seiten, Inhaltselementen und einzelnen Bearbeitungsschritten klarer erkennbar sind als im bisherigen Backend. Strukturen, die zuvor nur über mehrere Klicks oder implizites Vorwissen verständlich wurden, sind direkter sichtbar. Dadurch wirkt die Oberfläche weniger technisch und eher wie eine Abfolge konkreter Arbeitsschritte. Besonders für weniger erfahrene Nutzer:innen erleichtert dies den Einstieg, da weniger Systemlogik mitgedacht werden muss.

Positiv aufgenommen wurden außerdem dauerhaft sichtbare Informationen zum aktuellen Kontext. Elemente wie Seitentitel, Navigationspfade oder klar platzierte Aktionen zum Speichern und Veröffentlichen sorgen dafür, dass jederzeit nachvollziehbar bleibt, wo man sich befindet und welche Schritte möglich sind. Gerade bei umfangreicheren Seiten oder längeren Bearbeitungssitzungen geben diese Orientierungspunkte Sicherheit und verhindern, dass man sich innerhalb der Struktur verliert.

In den Gesprächen wurde deutlich, wie stark reale Arbeitsabläufe die Wahrnehmung der Oberfläche prägen. Viele Einschätzungen entstanden aus konkreten Situationen, etwa beim schnellen Korrigieren einzelner Inhalte oder beim parallelen Arbeiten an mehreren Seiten. Lösungen, die unnötige Wechsel zwischen Ansichten vermeiden und relevante Informationen direkt zugänglich machen, wurden in diesen Momenten als besonders hilfreich empfunden. Entscheidend war weniger eine einzelne Funktion als ein möglichst unterbrechungsfreier Arbeitsfluss.

Neben den positiven Rückmeldungen wurden jedoch auch einige Punkte kritisch gesehen. Der Übergang vom "Page Tree" in die konkrete Seitenansicht wurde teilweise als zu subtil beschrieben. Einige Teilnehmende bemerkten erst verzögert, dass sie sich bereits in einer anderen Bearbeitungsebene befanden. Hier könnten deutlichere visuelle Signale oder klarere Zustandswechsel helfen, Missverständnisse zu vermeiden.

Besonders unterschiedlich wurde die visuelle Live-Vorschau bewertet. Die Arbeit direkt im Layoutkontext verkürzt zwar den Weg zwischen Bearbeitung und Ergebnis und erspart zusätzliche Prüfschritte im Frontend, gleichzeitig kann die Vielzahl sichtbarer Elemente, vor allem bei komplexeren Seiten, zunächst überfordern. Ohne klare Führung ist nicht immer ersichtlich, wo mit der Bearbeitung begonnen werden soll.

Insgesamt zeigte sich, dass beide Darstellungsformen unterschiedliche Stärken haben. Die strukturierte Listenansicht unterstützt vor allem Orientierung und ein systematisches Vorgehen, während die visuelle Vorschau das schnelle Einschätzen von Layout und Wirkung erleichtert. Die Rückmeldungen sprechen daher für eine sinnvolle Verbindung beider Ansätze, statt einen davon vollständig zu ersetzen.

6.3 Evaluation aus der Perspektive von Content-Editor:innen

Im Mittelpunkt der Rückmeldungen aus der Content-Editor-Perspektive standen vor allem Effizienz und Alltagstauglichkeit. Viele Kommentare bezogen sich weniger auf gestalterische Fragen als auf konkrete Arbeitsschritte: Wie schnell lässt sich eine Seite finden? Wie viele Klicks sind für kleine Änderungen nötig? Wo entstehen unnötige Unterbrechungen? Bewertet wurde das Konzept daher primär danach, ob typische Routinetätigkeiten schneller und unkomplizierter erledigt werden können. Selbst kleine zusätzliche Schritte wurden als störend wahrgenommen, wenn sie sich im Tagesgeschäft häufig wiederholen. Dies gilt insbesondere in Situationen mit parallelen Aufgaben oder häufigen Kontextwechseln, in denen jede zusätzliche Navigation den Arbeitsfluss unterbricht.

Besonders positiv hervorgehoben wurden Quick Actions, Shortcuts sowie direkt erreichbare Bearbeitungsfunktionen. Mehrere Teilnehmende berichteten, dass sie im Arbeitsalltag zahlreiche kleine Anpassungen hintereinander vornehmen, etwa Texte korrigieren, Bilder austauschen oder einzelne Inhalte aktualisieren. In solchen Situationen summieren sich zusätzliche Klicks schnell zu spürbarem Mehraufwand. Wenn Bearbeitungsschritte ohne Modulwechsel oder lange Navigationswege möglich sind, wird dies daher als deutliche Entlastung empfunden. Die im Prototyp vorgesehenen Schnellzugriffe wurden entsprechend als praxisnah und zeitsparend beschrieben, da sie direkt an bestehende Arbeitsroutinen anknüpfen und typische Abläufe verkürzen. Gleichzeitig reduzieren sie die Notwendigkeit, wiederholt zwischen verschiedenen Bereichen des Backends zu wechseln.

Auch das Dashboard wurde als sinnvoller Einstiegspunkt bewertet. Übersichten zu zuletzt bearbeiteten Seiten oder offenen Aufgaben erleichtern den Wiedereinstieg und helfen, Prioritäten schneller zu setzen. Dadurch reduziert sich der Aufwand für die initiale Orientierung, sodass schneller mit der eigentlichen Bearbeitung begonnen werden kann.

Gleichzeitig zeigte sich, dass die Arbeit von Content-Editor:innen häufig unter Zeitdruck stattfindet und stark von wiederkehrenden Abläufen geprägt ist. Viele Tätigkeiten folgen festen Mustern und werden routiniert abgearbeitet. Im Kontext der Cognitive Load Theory (CLT) bedeutet dies, dass erfahrene Nutzer:innen durch Automatisierung ihre kognitive Belastung reduziert haben und Arbeitsschritte nahezu unbewusst ausführen. Entsprechend wurden vor allem solche Elemente positiv bewertet, die Entscheidungen reduzieren und den Ablauf möglichst geradlinig halten, während zusätzliche Komplexität oder neue Interaktionslogiken eher zurückhaltend aufgenommen wurden. Veränderungen an vertrauten Oberflächen können dabei selbst dann Irritationen auslösen, wenn sie funktional sinnvoll sind, da sich Nutzer:innen über Jahre an bestimmte Positionen, Icons oder visuelle Hinweise gewöhnt haben und diese nahezu automatisch ansteuern. Solche Veränderungen in eingespielten Arbeitsabläufen führen häufig zu Reibung und Akzeptanzproblemen, da Nutzer:innen Widerstand gegen veränderte Interaktionslogiken entwickeln. Neue Konzepte erfordern zunächst eine Umgewöhnung und werden kurzfristig als zusätzlicher Aufwand wahrgenommen, der die Akzeptanz beeinträchtigt [53].

Die visuelle Live-Vorschau im zweiten Konzept wurde vor diesem Hintergrund unterschiedlich wahrgenommen. Einerseits ermöglicht sie eine unmittelbare Rückmeldung über Layout, Bildwirkung und Textlängen und reduziert typische Schleifen aus Bearbeiten, Speichern und Prüfen im Frontend. Andererseits empfanden einzelne Teilnehmende die Vielzahl an sichtbaren Elementen als unübersichtlich, insbesondere bei komplexeren Seiten. Ohne klare Führung ist nicht immer ersichtlich, wo mit der Bearbeitung begonnen werden soll. Dadurch entsteht gerade zu Beginn zusätzlicher Orientierungsaufwand.

Insgesamt zeigen die Rückmeldungen, dass für Content-Editor:innen vor allem reibungslose Abläufe und kurze Wege entscheidend sind. Maßgeblich ist weniger das konkrete Interaktionsmodell als die Frage, ob sich typische Aufgaben schnell und ohne zusätzliche Hürden erledigen lassen. Verbesserungen werden vor allem dann akzeptiert, wenn sie sich nahtlos in bestehende Routinen einfügen und den Arbeitsalltag spürbar erleichtern.

6.4 Evaluation aus Entwicklungs- und Systemperspektive

Aus technischer Sicht wurde der Ansatz des Dashboards und der ersten Prototyp-Variante insgesamt als vergleichsweise unkompliziert umsetzbar eingeschätzt. Er bleibt nah am bestehenden TYPO3-Backend und baut auf bekannten Modulen, Datenstrukturen und Workflows auf. Die erforderlichen Anpassungen betreffen vor allem die Benutzeroberfläche und die Anordnung bestehender Backend-Komponenten. Die Änderungen erfolgen überwiegend innerhalb etablierter Strukturen und lassen sich entsprechend vergleichsweise einfach integrieren und warten. Gerade in gewachsenen Installationen, die über Jahre individuell konfiguriert wurden, gilt dieser Ansatz folglich als risikoarm und gut umsetzbar.

Deutlich anspruchsvoller zeigte sich hingegen eine visuelle Live-Vorschau beziehungsweise ein stärker frontendnahes Editing. Zwar existieren bereits einzelne Funktionen in diese Richtung, etwa feldbasiertes WYSIWYG-Editing über den Rich-Text-Editor (z. B. CKEditor). Diese Ansätze unterstützen vor allem einzelne Bearbeitungsschritte, stellen jedoch keine durchgängige Bearbeitung im realen Seitenkontext bereit. Für eine realitätsnahe Darstellung müsste das Backend Teile der tatsächlichen Frontend-Ausgabe nachbilden, insbesondere die Rendering-Logik mit Fluid-Templates, TypoScript und projektspezifischen Erweiterungen. Da sich diese Strukturen je nach Installation deutlich unterscheiden, ist eine einheitliche Vorschau technisch nur schwer generisch umzusetzen. Zusätzliche Anforderungen an Performance, Caching und Stabilität erhöhen den Integrationsaufwand weiter.

Aus technischer Perspektive bringt eine solche Lösung daher spürbar mehr Komplexität in Entwicklung und Betrieb. Entsprechend erscheint sie weniger als kurzfristiges Feature, sondern eher als langfristige Perspektive, die nur schrittweise weiterentwickelt werden kann.

Kapitel 7

Fazit und Ausblick

Ziel dieser Arbeit war es, die Nutzerführung des TYPO3-Backends aus der Perspektive von Content-Editor:innen zu untersuchen und darauf aufbauend konkrete Optimierungsansätze zu entwickeln. Ausgangspunkt war die Beobachtung, dass TYPO3 zwar einen großen Funktionsumfang bietet, für redaktionelle Nutzer:innen durch seine gewachsene Struktur und hohe Komplexität jedoch nicht immer intuitiv zugänglich ist. Im Zentrum stand daher die Frage, wie redaktionelle Arbeitsabläufe verständlicher, effizienter und näher an realen Nutzungssituationen gestaltet werden können. Entsprechend wurde untersucht, wie sich die Nutzerführung im TYPO3-Backend durch umsetzbare Optimierungsmaßnahmen und weiterführende Neugestaltungsideen so verbessern lässt, dass Arbeitsprozesse für Content-Editor:innen intuitiver und effizienter werden.

Im Rahmen einer heuristischen Analyse, eines systematischen Vergleichs mit anderen Content-Management-Systemen sowie der Entwicklung eines klickbaren Figma-Prototyps konnten mehrere zentrale Problemfelder identifiziert werden. Dazu zählen insbesondere Orientierungsprobleme in komplexen Strukturen, unnötige Klickwege, häufige Kontextwechsel zwischen unterschiedlichen Bereichen des Systems sowie fehlende unmittelbare Rückmeldungen zum aktuellen Status.

Das entwickelte Optimierungskonzept greift diese Punkte auf und kombiniert eine klarere Informationsarchitektur mit sichtbaren Status- und Steuerungselementen sowie verkürzten Interaktionswegen. Ergänzend wird das Dashboard als zentraler Einstiegspunkt stärker als arbeitsnahe „Einganglobby“ gedacht, die relevante Informationen bündelt und redaktionelle Aufgaben direkt unterstützt. Hinzu kommen Ansätze wie visuelles Editing, das die Distanz zwischen Bearbeitung und Ergebnis reduziert. Die qualitative Evaluation zeigt, dass diese Maßnahmen von unterschiedlichen Nutzergruppen als sinnvoll und praxisnah wahrgenommen werden. Besonders positiv bewertet wurden die verbesserte Orientierung sowie die stärkere Nähe zwischen Bearbeitung und Ergebnisdarstellung.

Ein zentraler Bestandteil der Evaluation war der Austausch mit der TYPO3-Community sowie ein Experteninterview mit Laura Heine, die das Backend-Redesign von TYPO3 Version 14 koordiniert. Die Gespräche verdeutlichten, dass viele der im Prototyp entwickelten Ansätze auch innerhalb der offiziellen Weiterentwicklung als relevant gelten. Einzelne Ideen, insbesondere die Quick Actions im Dashboard, wurden nicht nur positiv aufgenommen, sondern konkret in die laufende Entwicklung integriert und zur Umsetzung eingeplant. Je nach zeitlichen und technischen Rahmenbedingungen ist eine Integration bereits in Version 14 vorgesehen, andernfalls in einer der folgenden Versionen. Gerade das Dashboard bietet hierfür Spielraum, da dieser Bereich technisch vergleichsweise entkoppelt ist und sich leichter weiterentwickeln lässt.

Gleichzeitig zeigte sich, dass Veränderungen am TYPO3-Backend stets im Spannungsfeld technischer Komplexität, begrenzter Ressourcen und langfristiger Release-Zyklen stattfinden. TYPO3 folgt einem festen Entwicklungsrhythmus von etwa anderthalb Jahren pro Major-Version, während nur wenige Personen kontinuierlich am Core arbeiten. Vor diesem Hintergrund kön-

nen selbst sinnvolle UX-Verbesserungen meist nur schrittweise umgesetzt und über mehrere Versionen verteilt werden. Hinzu kommt, dass TYPO3 über viele Jahre zu einem sehr komplexen Ökosystem gewachsen ist. Zahlreiche Extensions, individuelle Konfigurationen und projektspezifische Anpassungen führen dazu, dass sich Bedienkonzepte je nach Installation deutlich unterscheiden. Maßnahmen lassen sich daher nur eingeschränkt verallgemeinern, was grundlegende Änderungen am System zusätzlich erschwert.

Die Ergebnisse dieser Arbeit sind deshalb bewusst als qualitative und konzeptionelle Beiträge zu verstehen. Ziel war nicht die Entwicklung einer universell gültigen Lösung, sondern das Aufzeigen konkreter Gestaltungsmöglichkeiten und das Sichtbarmachen redaktioneller Anforderungen. Gerade in einer historisch stark entwicklungsgetriebenen Systemlandschaft wird deutlich, wie wichtig es ist, UX-Aspekte und die Perspektive von Content-Editor:innen stärker einzubinden.

Zusammenfassend lässt sich die Forschungsfrage dahingehend beantworten, dass sich die Nutzerführung im TYPO3-Backend vor allem durch eine klar strukturierte Informationsarchitektur, kürzere Interaktionswege und arbeitsnahe Unterstützungsfunktionen spürbar verbessern lässt. Maßnahmen wie ein aufgabenorientiertes Dashboard mit integrierten Schnellzugriffen lassen sich vergleichsweise einfach ergänzen und verbessern den Arbeitsalltag unmittelbar. Eingriffe in die Navigation und die grundlegende Informationsstruktur greifen hingegen deutlich tiefer in bestehende Systemlogiken und Nutzungsmuster ein und sind entsprechend aufwendiger umzusetzen. Eine nachhaltige Verbesserung der Bedienbarkeit erfordert daher sowohl kleinere, gezielte Anpassungen als auch langfristige strukturelle Veränderungen. In ihrem Zusammenspiel ermöglichen sie es, TYPO3 Schritt für Schritt benutzerfreundlicher und näher an den tatsächlichen Arbeitsprozessen von Content-Editor:innen zu gestalten.

Literaturverzeichnis

- [1] K. Majrashi, M. Hamilton, and A. L. Uitdenbogerd, "Multiple user interfaces and cross-platform user experience: Theoretical foundations," in *Proceedings of the International Conference on Computer Science, Engineering and Applications*, CS & IT-CSCP, 2015. doi:10.5121/csit.2015.50205.
 - [2] S. K. Ramya, B. Sushma, and H. D. Sona, "Web Content Management System: A Comprehensive Review," *International Journal for Research Trends and Innovation*, vol. 10, no. 6, pp. 556–560, 2025.
 - [3] Regionales Rechenzentrum Erlangen (RRZE). Webauftritte der Hochschulen in Deutschland, <https://statistiken.rrze.fau.de/webauftritte/hochschulen/?monjahr=12.25&traeger=&land=&cms=>. abgerufen am 08.12.2025. [Online].
 - [4] J. Dyck. Content Editor UX: Why CMS Usability Is Tough, <https://evolvingweb.com/blog/content-editor-ux-why-cms-usability-tough>. abgerufen am 08.12.2025. [Online].
 - [5] TYPO3 Association. Roadmap for TYPO3 v14 — the High-Level Objectives and Strategic Goals. <https://news.typo3.com/archive/roadmap-for-typo3-v14-the-high-level-objectives-and-strategic-goals>. abgerufen am 09.12.2025. [Online].
 - [6] Informationstechnikzentrum Bund (ITZBund). Government Site Builder 11 (GSB 11), <https://produkt.gsb.bund.de/gsb11>. abgerufen am 28.01.2026. [Online].
 - [7] G. Baxter and I. Sommerville, "Socio-technical systems: From design methods to systems engineering," *Interacting with Computers*, vol. 23, no. 1, pp. 4–17, 2011. doi:10.1016/j.intcom.2010.07.003.
 - [8] T. Geis und G. Tesch, *Basiswissen Usability und User Experience: Aus- und Weiterbildung zum UXQB® Certified Professional for Usability and User Experience (CPUX) - Foundation Level (CPUX-F)*. dpunkt.verlag, 2023. 2., überarbeitete und aktualisierte Auflage.
 - [9] International Organization for Standardization. ISO 9241-11:2018 Ergonomics of human-system interaction – Part 11: Usability: Definitions and concepts, <https://www.iso.org/obp/ui/en/#iso:std:iso:9241:-11:ed-2:v1:en>. abgerufen am 09.12.2025. [Online].
 - [10] J. Nielsen. 10 Usability Heuristics for User Interface Design, <https://www.nngroup.com/articles/ten-usability-heuristics/>. abgerufen am 10.12.2025. [Online].
 - [11] Massachusetts Institute of Technology. Reading 17: Heuristic Evaluation, https://web.mit.edu/6.813/www/sp18/classes/17-heuristic-evaluation/#heuristic_evaluation. abgerufen am 10.12.2025. [Online].
 - [12] N. A. R. Amato, "Information architecture: Web design for usability," 2024.
-

- [13] P. M. Bach, R. DeLine, and J. M. Carroll, "Designers wanted: participation and the user experience in open source software development," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 985–994, 2009. doi:10.1145/1518701.1518852.
- [14] M. Hassenzahl, S. Diefenbach, and A. Göritz, "Needs, affect, and interactive products – facets of user experience," *Interacting with Computers*, vol. 22, no. 5, pp. 353–362, 2010. doi:10.1016/j.intcom.2010.04.002.
- [15] D. Norman, "Emotion & design: attractive things work better," *Interactions Magazine*, vol. 9, pp. 36–42, 2002. doi:10.1145/543434.543435.
- [16] TYPO3 Association. User Experience (UX) Team, <https://typo3.community/contribute/teams-committees/user-experience-ux>. abgerufen am 13.12.2025. [Online].
- [17] Interaction Design Foundation. User Centered Design (UCD), <https://www.interaction-design.org/literature/topics/user-centered-design>. abgerufen am 11.12.2025. [Online].
- [18] E. Lawrence, "User-centered design to enhance accessibility and usability in digital systems," 2024.
- [19] A. M. Heinecke und J. Gerken, *Mensch-Computer-Interaktion: Basiswissen für Entwickler und Gestalter*. Xpert.press, Wiesbaden: Springer Vieweg, 2024. 3. Auflage, doi:10.1007/978-3-662-67569-4.
- [20] M. Koch, J. Ziegler, C. Reuter, T. Schlegel und M. Prilla, "Mensch-Computer-Interaktion als zentrales Gebiet der Informatik – Bestandsaufnahme, Trends und Herausforderungen," *Informatik Spektrum*, vol. 43, no. 6, p. 381–387, 2020. doi:10.1007/s00287-020-01299-8.
- [21] J. Langan-Fox, S. Code, and K. Langfield-Smith, "Team mental models: Techniques, methods, and analytic approaches," *Human factors*, vol. 42, no. 2, pp. 242–271, 2000. doi:10.1518/001872000779656534.
- [22] N. Hollender, C. Hofmann, M. Deneke, and B. Schmitz, "Integrating cognitive load theory and concepts of human–computer interaction," *Computers in human behavior*, vol. 26, no. 6, pp. 1278–1288, 2010. doi:10.1016/j.chb.2010.05.031.
- [23] P. M. Salmon, M. G. Lenné, N. A. Stanton, D. P. Jenkins, and G. H. Walker, "Managing error on the open road: The contribution of human error models and methods," *Safety science*, vol. 48, no. 10, pp. 1225–1235, 2010. doi:10.1016/j.ssci.2010.04.004.
- [24] TYPO3 GmbH. Was ist TYPO3?, <https://typo3.com/de/typo3-cms/was-ist-typo3>. abgerufen am 13.12.2025. [Online].
- [25] J. A. Rohn, J. Spool, M. Ektare, S. Koyani, M. Muller, and J. Redish, "Usability in practice: alternatives to formative evaluations-evolution and revolution," in *CHI'02 extended abstracts on Human factors in computing systems*, pp. 891–897, 2002. doi:10.1145/506443.506648.
- [26] A. Altun, H. Yurdugül, and Y. Gülbahar, "Investigating usability constructs in a content management system," *Education and Science*, vol. 34, no. 152, pp. 160–173, 2009.
- [27] TYPO3 Association. Official TYPO3 Documentation, <https://docs.typo3.org/>. abgerufen am 13.12.2025. [Online].
-

- [28] S. Reinhardt. Accessible Web Design with TYPO3: A Comprehensive Guide, <https://nitsantech.de/en/blog/accessible-web-design-with-typo3>. abgerufen am 23.12.2025. [Online].
- [29] TYPO3 Association. Granular Access Rights in TYPO3, <https://typo3.com/typo3-cms/why-typo3/granular-access-rights>. abgerufen am 13.12.2025. [Online].
- [30] D. Lochner. Creating TYPO3 Backend Usergroups Your Clients Will Love, <https://b13.com/blog/creating-typo3-backend-usergroups-your-clients-will-love>. abgerufen am 13.12.2025. [Online].
- [31] D. Steeb. Optimizing Collaboration and Version Control in TYPO3: A Guide to Workspaces, <https://b13.com/blog/optimizing-collaboration-and-version-control-in-typo3-a-guide-to-workspaces>. abgerufen am 13.12.2025. [Online].
- [32] TYPO3 Association. UX and TYPO3—the Challenges for the Next Five Years, <https://news.typo3.com/archive/ux-and-typo3-the-challenges-for-the-next-five-years>. abgerufen am 13.12.2025. [Online].
- [33] S. Gasson, “Human-centered vs. user-centered approaches to information system design,” *Journal of Information Technology Theory and Application (JITTA)*, vol. 5, no. 2, p. 5, 2003.
- [34] L. Kuijter and A. De Jong, “A practice oriented approach to user centered sustainable design,” in *Ecodesign 2009 Conference, Sapporo*, 2009.
- [35] E. Insfran and A. Fernandez, “A systematic review of usability evaluation in web development,” in *International Conference on Web Information Systems Engineering*, pp. 81–91, Springer, 2008. doi:10.1007/978-3-540-85200-1_10.
- [36] J. Segal, “Some problems of professional end user developers,” in *IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC 2007)*, pp. 111–118, IEEE, 2007. doi:10.1109/VLHCC.2007.17.
- [37] J. Grudin, “The organizational contexts of development and use,” *ACM Computing Surveys (CSUR)*, vol. 28, no. 1, pp. 169–171, 1996. doi:10.1145/234313.234384.
- [38] J. Dumez, M. Bernaert, and G. Poels, “Development of software tool support for enterprise architecture in small and medium-sized enterprises,” in *International Conference on Advanced Information Systems Engineering*, pp. 87–98, Springer, 2013. doi:10.1007/978-3-642-38490-5_7.
- [39] TYPO3 Association. TYPO3 Demoprojekt, <https://demo.typo3.org/de/>. abgerufen am 19.12.2025. [Online].
- [40] O. Bartsch. Optimierte deinen täglichen Workflow mit dem TYPO3 Dashboard, <https://b13.com/de/blog/optimierte-deinen-taeglichen-workflow-mit-dem-typo3-dashboard>. abgerufen am 19.12.2025. [Online].
- [41] O. Bausewein and B. Franzke. TYPO3 Extension *custom_dashboard_widgets*, https://github.com/o-ba/custom_dashboard_widgets. abgerufen am 19.12.2025. [Online].
- [42] J. Sauer, A. Sonderegger, and S. Schmutz, “Usability, user experience and accessibility: towards an integrative model,” *Ergonomics*, vol. 63, pp. 1–23, 2020. doi:10.1080/00140139.2020.1774080.
-

- [43] W3C WAI. Web Content Accessibility Guidelines (WCAG) 2.2, <https://www.w3.org/TR/WCAG22/>. abgerufen am 23.12.2025. [Online].
- [44] European Commission. European accessibility act, https://commission.europa.eu/strategy-and-policy/policies/justice-and-fundamental-rights/disability/union-equality-strategy-rights-persons-disabilities-2021-2030/european-accessibility-act_en. abgerufen am 23.12.2025. [Online].
- [45] S. Schmutz, A. Sonderegger, and J. Sauer, "Implementing recommendations from web accessibility guidelines: A comparative study of nondisabled users and users with visual impairments," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 59, no. 6, pp. 956–972, 2017. doi:10.1177/0018720817708397.
- [46] A. Kreßmann. TYPO3 and its Accessibility in the Backend, <https://news.typo3.com/archive/typo3-and-its-accessibility-in-the-backend>. abgerufen am 23.12.2025. [Online].
- [47] S. Walsh. CMS Market Share Trends: Top Content Management Systems (Oct 2025), <https://www.searchenginejournal.com/cms-market-share/454039/>. abgerufen am 30.12.2025. [Online].
- [48] Drupal, "Overview of Drupal." <https://www.drupal.org/docs/getting-started/understanding-drupal/overview-of-drupal>. abgerufen am 31.12.2025. [Online].
- [49] webit! Gesellschaft für neue Medien mbH, "Storyblok CMS." <https://www.webit.de/storyblok/technologie/>. abgerufen am 02.01.2026. [Online].
- [50] A. Almani and O. Alrwais, "The role of wireframes in enhancing user interface design," *International Research Journal of Innovations in Engineering and Technology (IRJIET)*, vol. 8, pp. 134–140, December 2024. doi:10.47001/IRJIET/2024.812020.
- [51] Figma, Inc. Outfit Font, <https://www.figma.com/fonts/outfit/>. abgerufen am 18.01.2026. [Online].
- [52] L. Heine. Designing TYPO3 14's Backend: Our Vision for a Modern, Expandable Foundation, <https://b13.com/blog/designing-typo3-14s-backend-our-vision-for-a-modern-expandable-foundation>. abgerufen am 04.02.2026. [Online].
- [53] J. Grudin, "Groupware and social dynamics: Eight challenges for developers," *Communications of the ACM*, vol. 37, no. 1, pp. 92–105, 1994. doi:10.1145/175222.175230.
-

Anhang A

Prototyp

Der interaktive Prototyp wurde mit Figma erstellt und umfasst 16 Screens zur Darstellung der optimierten Benutzeroberfläche. Der vollständige Prototyp ist unter folgender URL verfügbar:

<https://www.figma.com/proto/p7hr6R60adNLLGmiD3hhPr/TYP03-CMS--bachelor-thesis-?node-id=9667-22986&starting-point-node-id=9667%3A22986&t=A0cjIEflkQdP09jK-8>

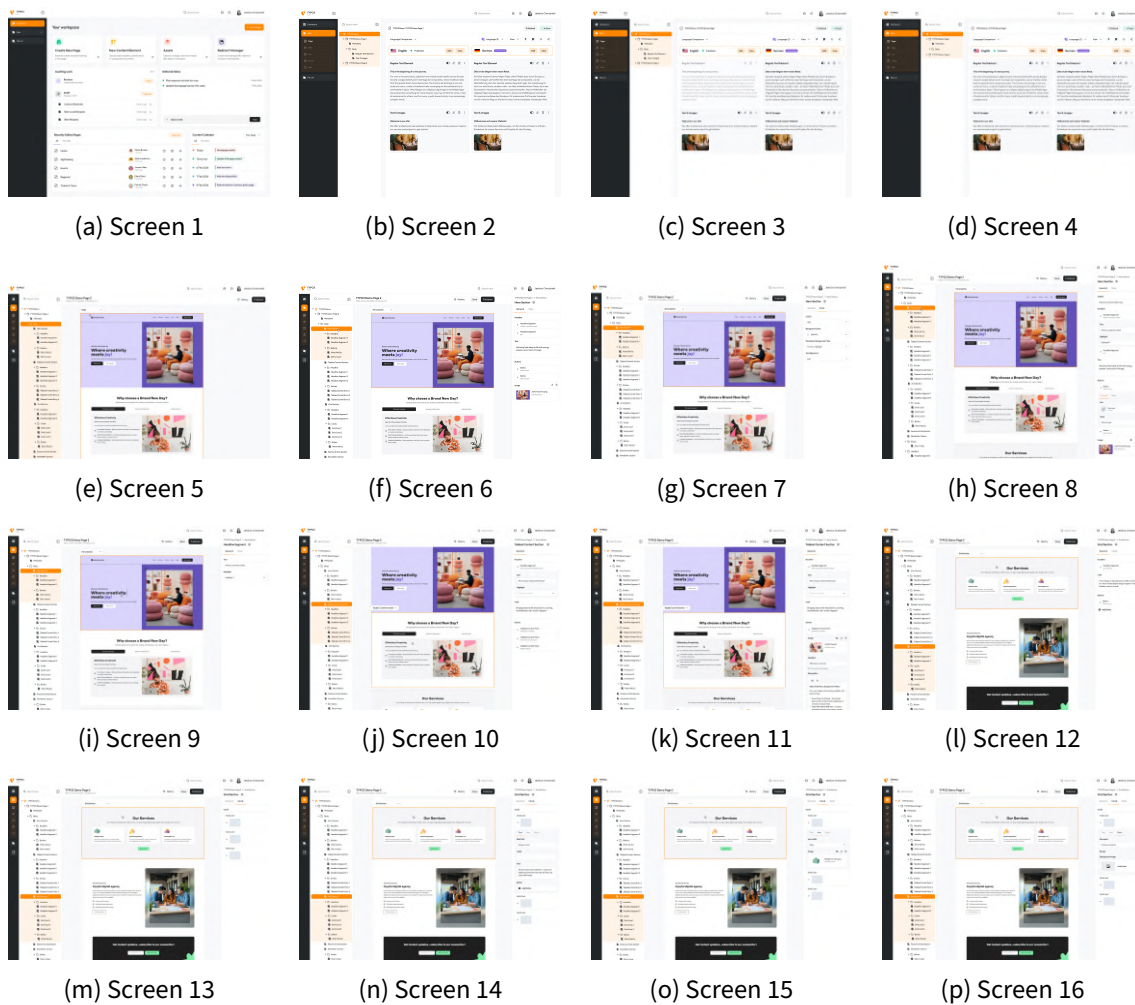


Abbildung A.1: Übersicht aller Prototyp-Screens der optimierten TYPO3-Backend-Oberfläche

Anhang B

Prototyp-Feedback

Das Feedback zum Prototyp wurde über den #cig-ux-Slack-Channel der TYPO3-Community (vgl. Abbildung B.1) sowie im erweiterten Fachkreis eingeholt und in einem Miro-Board (vgl. Abbildung B.2) dokumentiert. Zur Wahrung des Datenschutzes wurden in den dargestellten Slack-Ausschnitten Namen und Profilbilder anonymisiert.

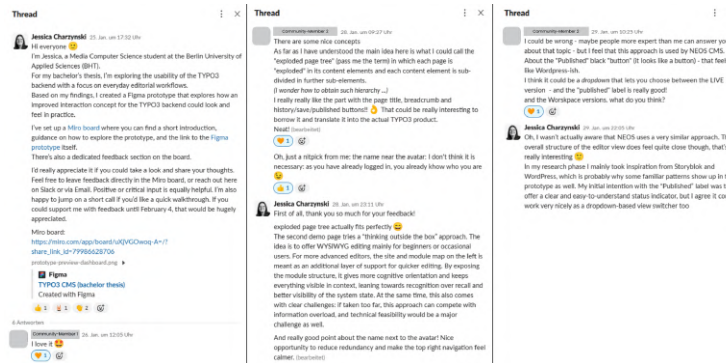


Abbildung B.1: Slack-Thread mit Feedback von TYPO3-Community-Mitgliedern

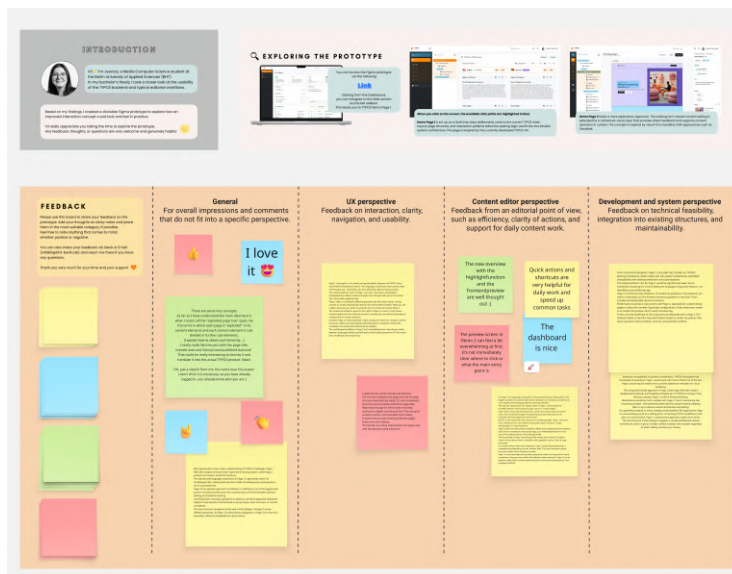


Abbildung B.2: Dokumentation und Strukturierung des Feedbacks im Miro-Board

Anhang C

Experteninterview

Interviewpartnerin:	Laura Heine (Leiterin des TYPO3 Version 14 Backend-Redesigns)
Datum:	3. Februar 2026
Dauer:	45 Minuten
Format:	Remote-Interview (Video-Call)
Art:	Semistrukturiertes Experteninterview

Methodische Einordnung

Das Interview wurde als semistrukturiertes Experteninterview durchgeführt. Ein vorbereiteter Leitfaden diente der thematischen Orientierung, wurde jedoch nicht starr befolgt. Stattdessen entwickelte sich ein offenes, dialogisches Gespräch, in dem einzelne Aspekte flexibel vertieft und durch spontane Nachfragen ergänzt wurden. Ziel war es, praxisnahe Einblicke in das Backend-Redesign von TYPO3 zu gewinnen. Die gewonnenen Einblicke dienten anschließend als qualitative Grundlage zur Einordnung der eigenen Analyse sowie zur Reflexion und Validierung der im Prototyp entwickelten Gestaltungskonzepte.

Gesprächsleitfaden

Der im Gespräch verwendete Leitfaden ist im Folgenden aufgeführt. Er diente als Orientierung und wurde im Verlauf des Interviews flexibel angepasst und stellenweise vertieft.

Einleitung und Rahmen

- Einverständnis zur Aufnahme und Klärung der namentlichen Nennung
- Kurze Vorstellung der Interviewerin und des Forschungsvorhabens (Bachelorarbeit, Zielsetzung, Prototyp und Evaluation)

Discovery-Phase

- Wie gestaltete sich die Discovery-Phase zu Beginn des Backend-Redesigns im Rahmen von Version 14?
 - Welche Usability- und Research-Methoden kamen dabei zum Einsatz (z. B. Interviews, Tests, Workshops, Feedback aus der Community)?
-

Strukturelle und technische Herausforderungen

- Welche besonderen Herausforderungen ergaben sich durch die Komplexität der bestehenden CMS-Architektur?
- Wie wurde mit historisch gewachsenen Strukturen und unterschiedlichen Nutzergruppen umgegangen?

Visuelles Design und User Experience

- Welche Rolle spielt eine visuelle Modernisierung (z. B. Typografie, Farben, Designsystem) im Kontext von Usability- und UX-Verbesserungen?
- Inwiefern beeinflusst ein konsistentes Erscheinungsbild Orientierung, Vertrauen und wahrgenommene Qualität?

Interaktionskonzepte und Effizienz

- Welche Rolle spielen modulübergreifende Funktionen wie „Quick Edit“ oder Quick Actions für effizienteres Arbeiten?
- Ist eine Ausweitung solcher Konzepte, etwa auf Dashboard-Ebene, geplant?
- Welche Bedeutung hat Frontend-Editing im aktuellen oder zukünftigen Bedienkonzept?

Aktueller Stand und Ausblick

- Wie ist der aktuelle Stand der Arbeiten an der kommenden LTS-Version?
- Welche organisatorischen oder technischen Schwerpunkte prägen diese Phase besonders?

Zukunftsperspektive

- Wenn Zeit und Ressourcen keine Rolle spielen würden: Welche drei Verbesserungen würdest du als Nächstes im Backend angehen oder gern noch in das v14-Release im April 2026 aufnehmen?

Interviewtranskript

Hinweis: Das folgende Transkript wurde für eine bessere Lesbarkeit leicht sprachlich geglättet. Füllwörter, Versprecher und redundante Wiederholungen wurden entfernt oder angepasst, ohne den Inhalt oder die Bedeutung der Aussagen zu verändern.

Zu Beginn erfolgte eine kurze Begrüßung. Die Interviewpartnerin gab ihr Einverständnis zur Audioaufzeichnung des Gesprächs für Transkriptionzwecke sowie zur namentlichen Nennung in der Bachelorarbeit.

-
- [JC]** Dann vielleicht erst einmal kurz zu mir: Ich bin Jessica, wir hatten ja schon kurz geschrieben. Ich schreibe aktuell meine Bachelorarbeit an der Berliner Hochschule für Technik im Studiengang Medieninformatik und mache das Ganze berufsbegleitend. Ich arbeite in Vollzeit als Content Managerin bei der Digitalagentur SYZYGY. Leider nicht direkt mit TYPO3, aber ich bringe auf jeden Fall die Content-Editor-Perspektive mit ein. Mit TYPO3 bin ich im Studium in Kontakt gekommen und zusammen mit meinem Betreuer Sebastian Kreideweiß, der auch sehr aktiv in der TYPO3-Community ist, auf das Thema gestoßen. Da ich viel Leidenschaft und Interesse für den Bereich Usability und Sichtbarkeit von Content-Management mitbringe, hat das einfach sehr gut gepasst. In meiner Bachelorarbeit habe ich mich vor allem aus der Brille der Content-Editor:innen an das TYPO3-Backend herangetastet und geschaut, wie der aktuelle Stand ist. Ich habe zunächst eine Literaturrecherche durchgeführt, danach eine Ist-Analyse vom Backend sowie eine Benchmarking-Analyse. Anschließend habe ich versucht, meine Erkenntnisse greifbar und erlebbar zu machen, indem ich sie in einem Prototypen visualisiert habe. Diesen habe ich im Slack-UX-Channel geteilt und dort den Tipp bekommen, mich mit dir zu vernetzen. Ich habe dann auch deinen Blogartikel gelesen. Ich glaube, wir teilen auf jeden Fall die gleiche Vision für TYPO3, da ging mir wirklich das Herz auf. Ich finde das Ganze total spannend. Ich bin noch ein kompletter Newbie in der TYPO3-Community, aber ich mag den Spirit sehr. Und für euch ist das ja gerade auch eine sehr arbeitsintensive Zeit rund um Version 14, deshalb weiß ich es umso mehr zu schätzen, dass du dir die Zeit nimmst, gemeinsam mit mir darauf zu schauen.
- [LH]** Voll gerne. Das Bottleneck ist meistens die Programmierung, da bin ich nicht direkt drin. Ich habe meinen Teil im Grunde schon abgegeben, und jetzt geht es darum, so viel wie möglich umzusetzen. Das ist gerade die eigentliche Herausforderung.
- [JC]** Oh stimmt, wie viel man da noch reinbekommt, kapazitätsmäßig, stelle ich mir echt herausfordernd vor.
- [LH]** Ja, das ist übel. Brutal, ehrlich gesagt.
- [JC]** Ich dachte, wir sprechen kurz über den Prototypen, und danach hätte ich auf jeden Fall noch einige Fragen an dich, zum Prozess bei euch und zur aktuellen Entwicklung. Ich kann dich gerne einmal kurz durchführen, falls du keine Zeit hattest, schon genauer reinzuschauen.
- [LH]** Ich habe ihn mir schon angeschaut und auch ein bisschen durchgeklickt. Erklär gern nochmal mit deinen eigenen Worten, wie dein Ansatz dahinter ist.
- [JC]** Sehr gerne. Ich share einmal meinen kompletten Bildschirm. Du solltest gerade mein Miro-Board sehen, und ich springe direkt ins Figma-Board. Konzeptionell bin ich so rangegangen, zuerst zu überlegen, wie man das Dashboard als Vision gestalten könnte, quasi wie eine Eingangslobby für Content-Editor:innen. Aktuell ist es ja eher funktional und stark aus einer Admin-Perspektive gedacht. Ich habe mir deshalb die Frage gestellt: Was wäre im Daily Business wirklich hilfreich? Zum Beispiel Shortcuts als direkte Einstiegspunkte, ähnlich wie ihr es in Version 14 schon im Content Editing verfolgt. Dann ein Überblick über anstehende Aufgaben: Welche Seiten sind im Draft, wo muss ich noch ran, gibt es Inhalte von Kolleg:innen, die geprüft werden müssen. Ergänzend ein kleines Notiztool als persönliche Gedankenstütze, zuletzt bearbeitete Seiten und ein Content-Kalender für geplante Go-Lives oder Updates.
-

- [JC]** Von dort gelangt man in eine Demo-Seite, die sehr nah an der aktuell entwickelten Version ist. Das Layout bleibt vertraut, ist aber gestalterisch leicht aufgefrischt. Statusanzeigen helfen dabei, jederzeit zu verstehen, wo man sich gerade im Workflow befindet. Shortcuts sind nicht an der Seite, sondern direkt im Arbeitsbereich platziert. Im zweiten Schritt bin ich dann bewusst losgelöst vom Status quo vorgegangen und habe versucht, etwas freier zu denken, auch mit Blick auf andere CMS. Die Idee war, mehr Kontext zu geben. In der Sitemap sieht man immer, wo man sich gerade befindet, und der aktuelle Bereich wird hervorgehoben. Daraus entstand eine Dreiteilung: links die Sitemap, in der Mitte eine visuelle Vorschau als Frontend-Editing und rechts weiterhin die klassische formularbasierte Bearbeitung. So hätten erfahrene Nutzer:innen ihre gewohnte Arbeitsweise, während Neue direkt in der Preview arbeiten könnten, ähnlich wie man es vom Gutenberg-Editor kennt. Man könnte Elemente entweder über die Sitemap oder direkt in der Vorschau auswählen. Der aktuelle Bereich wird visuell markiert, und bei Bedarf kann man jederzeit zurück in die Formularansicht wechseln. Die Idee ist, Kontextwechsel zu reduzieren und die Bearbeitung möglichst nah am eigentlichen Inhalt stattfinden zu lassen. Hättest du dazu erst mal Fragen?
- [LH]** Nee, ich finde deinen Ansatz auf jeden Fall richtig gut. Frontend-Editing ist ja immer wieder ein Thema, das wird ständig diskutiert, wie und wann man das sinnvoll integrieren kann. Ich weiß gar nicht, wie viel sie davon in Version 14 noch unterbringen, aber es bewegt sich was, auch wenn es gefühlt im Schneckentempo vorangeht.
- [JC]** Versteh ich total. Wenn man so nah dran ist und richtig dafür brennt, merkt man erst, wie komplex dieses ganze Universum ist, dieser Organismus TYPO3. Veränderungen dort umzusetzen, stelle ich mir wirklich herausfordernd vor.
- [LH]** Ist es auch. Es gibt unglaublich viele Constraints, mit denen man vorher gar nicht rechnet. Man zeigt ein Design und denkt, das passt. Und dann heißt es: Gute Idee, aber ... Und ab da hängen plötzlich Dinge zusammen, von denen wir als Designer gar nicht wussten, dass sie technisch miteinander verknüpft sind. Gerade in TYPO3 gibt es viele spezielle Edge Cases. Deshalb muss man immer schauen, wo man ansetzen kann, mit kleinen, machbaren Änderungen, die langfristig auf ein größeres Ganzes einzahlen. Wenn ich Schneckentempo sage, meine ich das wirklich wörtlich. Alles dauert lange, vor allem wenn man es sauber machen will, ohne an anderer Stelle etwas kaputtzumachen. Dafür braucht man vor allem Geduld.
- [JC]** Das glaube ich sofort. Wann habt ihr eigentlich mit den Vorarbeiten im UX-Team für Version 14 begonnen? Und wie sah euer Discovery-Prozess aus?
- [LH]** Es gibt ja das UX-Team, also auch den Channel, in dem du unterwegs bist. Das ist das offizielle TYPO3-UX-Team, da bin ich selbst gar nicht fest drin. Ich arbeite ganz normal in einer Agentur und mache Webseiten, ich bin also nicht das ganze Jahr über im TYPO3-Kosmos aktiv. Das war bei mir eher ein Sonderfall. Dieses UX-Team arbeitet aber schon länger an Strategie und Visionen. Ich glaube, vor drei oder vier Jahren ging das los. Letztes Jahr kam dann die Entscheidung, dass sich das Backend für die nächste LTS grundlegend ändern soll. TYPO3 läuft ja in festen Release-Zyklen von anderthalb Jahren. In dieser Zeit muss alles rein, was man für die nächste Version schaffen will. Danach ist erst mal wieder Ruhe, bis der nächste Zyklus startet. Wenn die LTS 14 draußen ist, ändert sich am Backend für anderthalb Jahre praktisch nichts mehr. Erst danach beginnt wieder das nächste Zeitfenster für größere Anpassungen. Und genau so war es jetzt auch. Diese anderthalb Jahre wurden vor allem dafür genutzt, die Grundlage zu schaffen.
-

[LH] Um die Änderungen umzusetzen, die wir jetzt sehen oder die die Devs gerade fertigstellen, musste im Hintergrund unglaublich viel aufgeräumt werden. Das merkt man als User gar nicht. Aber im Code war wohl einiges fällig. Die nennen das immer Housekeeping. Funktionen entkoppeln, Abhängigkeiten auflösen, alte Strukturen entfernen, damit man später flexibler arbeiten kann. Die Codebase von TYPO3 ist schließlich rund 25 Jahre alt. So etwas findet man heute in kaum einem Projekt noch. Entsprechend viel Ballast sammelt sich da über die Zeit an.

[JC] Das ist ja echt historisch angewachsen.

[LH] Genau. Und deswegen musste erst mal richtig viel Vorarbeit stattfinden. Und dann ging es darum: Was schaffen wir noch bis zur 14 an optischen Änderungen. Damit haben wir eigentlich erst im Oktober angefangen, letztes Jahr. Wir haben uns mit den Devs zusammengesetzt und geguckt: Realistisch gesehen, was wäre eine Option, oder wie kriegen wir das hin, dass es einen Impact hat, und zwar so, dass man ihn sieht und nicht nur unter der Haube. Weil die User in den meisten Fällen keine Devs sind, sondern Redakteur:innen und Editors, und denen ist es relativ egal, was sich an der Codebase ändert, das sieht man ja nicht. Wir wollten schon hinkriegen, dass man den optischen Eindruck hat, dass sich etwas tut und dass es in eine neue Richtung geht. Und wir haben im Oktober angefangen mit dieser Vision, die wir auch im Blogpost noch drin hatten, da sind Screenshots drin. Nach der T3Con, wo Benni, also Benni ist mein Chef, das vorgestellt hat, haben wir dann geguckt: Wir haben noch so und so viele Wochen und nur begrenzt Dinge. Und dann ist es wieder so: Man denkt sich, ich ändere die Farben. Das ist bei TYPO3 krass aufwendig. Ich mache sonst Webseiten: Wenn ich die Website blau machen will, ist sie blau. Aber bei TYPO3, wenn du sagst, du willst das ganze Backend ein bisschen tinten und die Grautöne ein bisschen ändern, dann sagt dir der Dev erst mal: Das sind ja keine festen Farbwerte. Und ich so: Was? Ja, die werden dynamisch berechnet. Und das sind Dinge, die macht man nicht mal eben. Der Hintergrund ist, dass man das Ganze flexibel halten will, damit du später in der Zukunft dein eigenes Color-Theme im Backend haben kannst. Wenn du deine Corporate-Farben picken kannst und sagst, meine Farben sind blau, dann kannst du das einstellen und alles färbt sich dementsprechend um, aber immer mit dem Gedanken Barrierefreiheit, also Kontrastanforderungen. Das dynamisch zu berechnen, ist mega schwierig. Das geht im Moment noch nicht hundert Prozent, deswegen gibt es noch keinen Color-Picker. Aber die Farben müssen so vorbereitet sein, dass man sie in Zukunft einfach wechseln kann. Das heißt, ich kann meinem Dev nicht einfach sagen, hier ist meine Grayscale, nimm die, sondern wir müssen zusammen gucken, wie man die berechnet und wie die dann aussehen. Und allein das Farben aussuchen hat schon ewig gedauert. Dann so Dinge wie: Lass uns die Icons tauschen. Ich weiß nicht, ob du das verfolgt hast, aber das ist auch ein Thema.

[JC] Oh stimmt das habe ich mitgekomen, war auf jeden Fall sehr heiß diskutiert.

[LH] Jesus Christ, ja. Es war wichtig, dass wir das gemacht haben, und ich finde es auch gut, dass die jetzt schon live sind. Ich glaube, in 14.1 kamen sie direkt mit raus. Das hat sofort Impact, weil es auf den ersten Blick ganz anders aussieht. Aber klar, Veränderungen finden viele erst mal schwierig. Das ist ja fast immer so. Und bei einem Produkt wie TYPO3, das Leute seit 25 Jahren benutzen, kann das schnell kippen. Da wird jemand wirklich sauer, nur weil sich Icons ändern. Es gibt auch berechtigtes Feedback, zum Beispiel dass das Color-Coding fehlt. Früher wusstest du: Mein Modul war orange. Das fällt jetzt weg. Man muss sich erst mal neu orientieren. Wir haben deshalb auch schon einen zweiten Icon-Release gemacht und Dinge angepasst, weil man oft erst im Live-Betrieb merkt, wie etwas wirklich wirkt. Da haben wir direkt noch eine Runde nachgeschärft.

- [LH]** Und auch die Icon-Größe war mega der Struggle. Ich wollte die acht Pixel kleiner machen. Und Benni, mit dem ich das zusammen mache, meinte dann: Jetzt stellst du dir das einfach vor, das geht nicht. Und ich so: Wie kann es sein, dass ich die nicht acht Pixel kleiner machen kann? Dann erklärt er mir, auf welchen Größen die angelegt sind, wie die runtergeskaliert werden, dass alles von der 64er Base ausgeht, dann auf 32 Pixel runterskaliert wird, immer noch pixelgenau ausgerichtet ist und auf 16 Pixel auch funktionieren muss. Und dann stehe ich da und denke: Ja, okay, dann kann ich die wirklich nicht kleiner machen.
- [JC]** Krass, man muss wirklich jeden Edge-Case mitdenken.
- [LH]** Ja, da tauchen ständig neue Fälle auf. Und dann kommt noch dazu, dass jede:r contributen kann. Das ist einerseits der große Vorteil von TYPO3, macht das System aber auch unberechenbar. Icons kann zum Beispiel jede:r für die eigene Extension hochladen, wie er oder sie möchte, und dann sieht wieder alles anders aus. Man kann nie alles kontrollieren. Man kann nur eine solide Grundlage schaffen, mit der andere arbeiten können, sodass es trotzdem halbwegs konsistent bleibt. Da spielt so viel mit rein. Ich habe bei deinem Prototyp auch gesehen, dass du die Ecken abgerundet hast. Wir hatten anfangs Buttons, die komplett rund waren, so Pills. Sah super aus, aber braucht einfach zu viel Platz. Das sind dann diese Momente, wo man als Designer denkt: sieht toll aus. Und dann kommt ein Dev und erklärt dir die technischen Grenzen, und man merkt: okay, geht halt nicht. Am Ende sind es Baby Steps. Aber es bewegt sich. Langsam kommen wir dahin.
- [JC]** Total.
- [LH]** Und mit der 14 haben wir jetzt zumindest eine gute Basis. Dinge wie die Quick Actions rechts rutschen wahrscheinlich eher in die 15, außer es passiert noch ein kleines Wunder.
- [JC]** Aber dann habt ihr zumindest schon die nächste Ausbaustufe im Blick.
- [LH]** Ja. Ich würde mir nur wünschen, dass man schneller releasen könnte.
- [JC]** Dass man nicht immer anderthalb Jahre warten muss.
- [LH]** Genau. Aber die Rahmenbedingungen sind halt, wie sie sind. Für so ein großes System arbeiten erstaunlich wenige Leute Vollzeit daran und tragen wirklich viel Verantwortung. Klar kann jede:r contributen, aber irgendwer muss am Ende den Hut aufhaben, alles prüfen und sicherstellen, dass nichts kaputtgeht. Und das sind echt nicht viele.
- [JC]** Das ist schon heftig. Ich kann mir gut vorstellen, wie intensiv die Phase gerade ist. Für die Devs sowieso, aber für dich bestimmt auch. Ständig dieses Abwägen: Was schaffen wir noch in dieses Release rein?
- [LH]** Der größte Druck liegt definitiv bei den Devs, die das jetzt umsetzen. Die kämpfen echt. Und jedes Feature, das man anfasst, zieht irgendwo neue Baustellen nach sich. Deshalb sind wir am Ende einfach froh über alles, was wir sauber reinbekommen.
- [JC]** Ja, absolut.
- [LH]** Du hast in deinem Prototyp ja auch die Suche nach oben gezogen. Das ist gerade noch ein Thema. Mit etwas Glück schaffen wir das noch in die 14. Wahrscheinlich wird sie dann fest in der Top Bar als offene Suchleiste integriert.
- [JC]** Das klingt super! Schafft es der Page Wizard schon in die LTS-Version? Den fand ich auch sehr, sehr cool.
-

[LH] Vermutlich nicht. Ich glaube, die arbeiten beim Code-Sprint gerade dran. Aber das ist auch so ein Feature, das hätten wir richtig gern drin gehabt. Die Grundvision von dem, was wir jetzt gemacht haben, war ja: Der Look hat sich zwar ziemlich geändert, und dieses ganze Aufräumen im Hintergrund war wichtig. Aber eigentlich wollten wir vor allem bei Usability und User Experience ansetzen, damit man den Kontext nicht mehr so oft verliert. Ich glaube, das hattest du auch in deinem Prototyp. Du gehst auf ein Content-Element, bearbeitest es, alles lädt neu, und plötzlich bist du ganz woanders und hast keine Ahnung mehr, wo du herkommst. Bei den Seiteneigenschaften ist es genauso, das reißt einen komplett raus. Genau das wollen wir reduzieren, also diese ständigen Brüche. Deshalb auch die Quick Actions, die sich nur drüberlegen, und der Page Wizard, der einen Schritt für Schritt durchführt, bis alles erledigt ist, ohne dass im Hintergrund neu geladen wird. Das wären für die Experience richtig gute Verbesserungen, aber technisch ist das einfach mega komplex.

[JC] Total.

[LH] Wenn es nicht in die 14 kommt, dann hoffentlich in die 15. Eigentlich sind wir uns alle einig, dass dieser Page Wizard einfach dazugehört.

[JC] Das wäre echt ein Game Changer. Ich fände das auch richtig stark.

[LH] Ja, absolut. Und dann ist da noch der Page Tree. Der ist wahnsinnig kompliziert, da kann man gar nicht so viel ändern. In meiner ersten Version hatte ich zum Beispiel die Linien rausgenommen, weil es ruhiger und weniger überladen wirkt. Aber dann kommt sofort die Frage: Was ist, wenn du fünf Ordner untereinander hast und die Titel nicht mehr richtig lesen kannst? Dann brauchst du plötzlich einen horizontalen Scroll. Und ohne Linien fehlt dir die Orientierung. Dann merkt man schnell: Okay, die Linie muss doch bleiben. Solche Diskussionen hat man ständig. Aber genau dieser Austausch ist wichtig. Wir haben jeden Tag Stand-up gemacht, geschaut, wer woran arbeitet, wo es hängt und wer Unterstützung braucht. In diesen Modus muss man kommen. Und ich hatte im Blogartikel auch geschrieben: TYPO3 war in der Vergangenheit extrem von den Devs geprägt, und es wird auch in Zukunft so sein, weil sie es bauen. Aber UX mit reinzukriegen, ist wichtig. Und ich finde es gut, wenn du deine Stimme zeigst in der Community, weil dann hört man, dass es UX-Leute gibt, die Bock haben und Motivation, auch was zu machen, und an die man sich wenden kann. Default ist oft, dass ein Dev eine Idee hat und das baut. Und dann schaust du als Designer drauf und denkst: Hätte ich anders gemacht, auch vom Userflow her. Je lauter die UX-Community innerhalb der TYPO3-Community wird, desto eher wird nachgefragt.

[JC] Ja, das glaube ich auch.

[LH] Und je enger man mit den Devs zusammenarbeitet, desto eher etabliert sich das. Aber leicht ist es nicht.

[JC] Ich glaube auch, dass das genau der richtige Weg ist. Genau dadurch bekommt das Thema mehr Raum und bringt die Perspektive der Content-Editor:innen stärker rein. Habt ihr dafür eigentlich Usability-Methoden eingesetzt? Wart ihr im Austausch mit Editor:innen oder konnten sie schon testen? Oder plant ihr das eher nach dem LTS-Release im Live-Betrieb?

- [LH]** Das UX-Team hat auf jeden Fall Umfragen gemacht und User Research betrieben, ich glaube allerdings vor zwei Jahren oder so, also schon eine Weile her. Und dadurch, dass wir mit unseren Kunden fast nur in TYPO3 arbeiten, bekommen wir sowieso ständig Feedback. Das ist nicht quantitativ, keine Ahnung, vielleicht 20 Kunden oder so, aber eben qualitativ. Und da merkt man schon ziemlich gut, wie die Leute tatsächlich damit arbeiten. Das Ding ist nur: Qualitatives Feedback zum TYPO3-Backend ist super divers, weil jedes Backend anders aussieht. Je nachdem, welche Agentur es gebaut hat und welche Extensions drin sind, fühlt sich alles anders an. Jeder Workflow ist anders, Leute sind unterschiedliche Dinge gewohnt, Workspaces sind anders konfiguriert. Da einen gemeinsamen Nenner zu finden, ist echt schwierig. Man muss viel auf den Subtext hören. Und dieses ständige Kontext-Springen taucht zum Beispiel immer wieder auf, egal wie der Workflow aussieht. Aber insgesamt ist User Feedback durch diese Vielfalt echt eine Challenge. Man kann nicht nur einen Edge Case optimieren, weil es hundert andere gibt. Man muss eher an den Grundlagen schrauben und hoffen, dass Agenturen und Devs das dann sinnvoll konfigurieren.
- [JC]** Stimmt, das hattest du auch im Blogartikel geschrieben: Man gestaltet im Grunde den Rahmen, aber wie es am Ende aussieht, liegt gar nicht mehr komplett in der eigenen Hand. Man kann nur versuchen, möglichst viel mitzudenken. Das ist echt nicht ohne.
- [LH]** Ja, total. Was manche mit ihrem Backend bauen, da fällt dir echt nichts mehr ein. Das ist wild, komplett wild. Und dann bringst du ein Update raus und plötzlich bricht alles, weil irgendwo super spezielle Sonderfälle drinstecken. Dann sind die Leute natürlich erst mal frustriert, aber da kannst du manchmal einfach nichts machen.
- [JC]** Aber die Grundresonanz in der Community ist bisher eher positiv, oder? Wie würdest du das einschätzen?
- [LH]** Doch, auf jeden Fall. Auf das Grundsätzliche gab es richtig positives Feedback. Auf die Vision, die wir auf der T3Con vorgestellt haben, kam fast nur Zuspruch, ich glaube, gar nichts wirklich Negatives. Ich glaube, vielen ist einfach bewusst, dass TYPO3 im Vergleich zu anderen CMS ziemlich oldschool wirkt. Der Markt ist krass, ständig kommen neue Systeme dazu, und man sieht natürlich, was die Konkurrenz macht. Deshalb haben wir beim Design auch immer gesagt: Es geht nicht darum, etwas perfekt zu machen, sondern erst mal anders. Perfekt gibt es sowieso nicht. Design ist subjektiv, und irgendwas wird man später immer noch verbessern müssen. Aber zu zeigen, dass TYPO3 auch anders aussehen kann, dass da nochmal Bewegung reinkommt, das war vielen wichtig. Daran haben manche gar nicht mehr so richtig geglaubt. Deshalb war die Resonanz insgesamt echt positiv. Und jetzt schauen wir, wie es weitergeht und wo wir am Ende landen.
- [JC]** Da bin ich sehr gespannt. Habt ihr beim Page Wizard und bei den Shortcuts, die eher in die 15 rutschen, schon verschiedene Ansätze ausprobiert? Gibt es da noch Ideen in der Pipeline? Ich könnte mir zum Beispiel so ein Quick Edit als Overlay auch bei Modulen vorstellen, aber ich glaube, wenn man damit anfängt, macht man schnell die Büchse der Pandora auf und hat wieder tausend neue Baustellen.
- [LH]** Was wir auf jeden Fall noch haben, ist WYSIWYG Editing. Man kann Preview Links machen und dann den Draft auf der Website previewen. Der aktuelle Stand ist, dass man das nutzt, um Frontend Editing möglich zu machen.
-

[LH] Ähnlich wie du das hattest: rechts das Formular und links die Seite, wo die einzelnen Teile gehighlightet werden. Direkt auf der Seite zu bearbeiten ist aktuell noch schwierig. Ich weiß nicht, wie machbar das am Ende wirklich ist, weil da einfach extrem viel mit reinspielt. Aber das Formular direkt daneben zu haben und Inhalte dort einzugeben, wirkt deutlich realistischer. Ob das dann eher eine 15- oder 16-Geschichte wird, müssen wir schauen. Ein großes Thema sind auch Workspaces. Die können super viel, wenn man sie richtig konfiguriert. Damit deckst du fast jeden Edge Case ab, mit Translation-Workflows, Rechten und Rollen. Aber die Idee wäre zusätzlich eine Light-Version. Also ein einfacher Draft-Workflow, für den man keinen kompletten Workspace aufsetzen muss. So wie man es aus anderen CMS kennt: Ich bearbeite etwas, es ist erst mal ein Draft, und dann kann ich es publishen. Ganz ohne großen Overhead. Ich hoffe, dass das eher in der 15 kommt als erst in der 16. Und dann natürlich die Content-Stage generell. Da haben wir noch gar nicht so viel verändert. Klar, Styles, Farben und Rounded Corners ziehen sich durch, aber eigentlich geht es darum, dass man Content-Elemente direkt dort bearbeiten kann, ohne ständig rauszuspringen. Dass sich Dinge eher aufklappen oder inline editieren lassen. Wie genau das am Ende aussieht, ist aber noch offen.

[JC] Klingt auf jeden Fall spannend.

[LH] Ja, da haben wir noch viele Ideen in der Schublade. Das sind gerade so die großen Themen. Viele kleinere Verbesserungen nebenbei. Das Dashboard ist bei uns zum Beispiel schon ewig ein Thema, wie man das besser machen kann. Ich glaube, da kommt in der 14 auch noch ein bisschen was.

[JC] Oh, das wäre echt klasse.

[LH] Und einer von den Devs, mit denen ich viel zusammenarbeite, der auch bei uns in der Agentur ist, hat sich deinen Prototyp angeschaut und fand die Quick Actions richtig gut. Ich glaube, da ist bei ihm schon so ein kleiner Floh ins Ohr gesetzt worden. Mal schauen, was beim Dashboard noch passiert. Das sind eher dankbare Baustellen, weil der Bereich etwas abgekapselt ist und nicht so viele Abhängigkeiten hat. Ich meine, es gibt auch ein Bookmarks-Widget, aber da will ich nichts versprechen.

[JC] Oh, das freut mich total zu hören. Wenn davon wirklich etwas ins Dashboard oder generell ins Produkt wandert, wäre das richtig schön. Hättest du Sachen am Prototyp, wo du sagen würdest: besser nicht? Fühl dich frei, auch zu roasten. Das hilft mir total. Ich habe auch mit ein paar Content-Editor:innen gesprochen, und die meinten, weil sie eher formularbasiert arbeiten, dass so eine Frontend-Editing-View erst mal ziemlich überwältigend sein kann. Ich glaube, das ist bei euch ja auch immer die Frage: Wie holt man unterschiedliche User Groups ab ohne sie zu überfordern? Also ohne dieses Gefühl auszulösen: Wo klicke ich jetzt eigentlich?

[LH] Ja, das ist ein spannendes Thema. Es gibt User, die das seit gefühlt tausend Jahren benutzen, und andere sind einfach nicht besonders internet-affin. Viele Editoren pflegen nur die Website ihres Unternehmens, sind aber keine UX-Leute oder super techniknah. Wir kennen irgendwann jedes CMS, die eben nicht. Die fühlen sich dann total wohl in ihrem TYPO3. Und genau da liegt die Balance: Wie viel kann ich verändern, und wie viel sollte ich lieber lassen? Irgendwann muss man sich aber auch committen. Man kann nicht immer versuchen, es allen recht zu machen, das funktioniert einfach nicht.

- [LH] Solange es noch eine alternative Möglichkeit zum Bearbeiten gibt, so wie du es ja vorgesehen hast, und es nicht der einzige Weg ist, finde ich eine neue Ansicht völlig in Ordnung. Technisch kann ich nicht einschätzen, was alles machbar ist, aber das ist auch nicht dein Part. Du kannst dich erst einmal auf die Vision konzentrieren, die Technik kommt später. Wichtige Rahmenbedingungen sind vor allem Barrierefreiheit, gute Farbkontraste und dass möglichst viele Funktionen klar beschriftet sind. Natürlich geht das nicht überall, aber zumindest Tooltips oder Labels sollten vorhanden sein. Und bei solchen Ansichten, wo viel parallel passiert, muss man auch bedenken, dass viele auf kleineren Bildschirmen arbeiten.
- [JC] Stimmt, das alles unter einen Hut zu bekommen, ist wirklich nicht ohne.
- [LH] Ich denke mir zwar manchmal: Warum arbeitet man mit einem so kleinen Bildschirm im TYPO3-Backend? Nimm doch einen größeren. Aber gut, das ist eben die Realität. Das sind die wichtigsten Punkte, die man mitdenken sollte. Ansonsten gab es im Prototyp nichts, wo ich sagen würde: Das würde ich auf keinen Fall so machen. Nur der Wechsel zwischen Page Tree und Seitenaufbau war für mich etwas subtil. Am Anfang habe ich nicht direkt verstanden, dass ich damit schon in der Seitenansicht bin und dass dort plötzlich Header und Body stehen statt meiner Seiten. Vielleicht kann man diesen Übergang noch klarer machen und ermöglichen, schneller zwischen Seiten zu wechseln. Das ist mir aufgefallen. Und sonst fand ich viele Ansätze wirklich gut. Die Top Bar geht zum Beispiel genau in die Richtung, die wir auch verfolgen. Beim Modul-Menü links wird man allerdings akzeptieren müssen, dass es mehr als drei Module sind.
- [JC] Vielen Dank für dein Feedback! Oh ja stimmt, das ist dann einfach die Realität, mit der man arbeiten muss.
- [LH] Genau. Man kann trotzdem mit einer reduzierteren Ansicht arbeiten, denn viele Redakteur:innen sehen ja gar nicht alle Module. Aber unsere Screens müssen immer den Extremfall zeigen: Super-Admin, alle Rechte, alles sichtbar. Dann ist die Liste eben lang und man muss scrollen. Oder bei den Icons: Was passiert, wenn ich sechs Gruppen untereinander habe, wie unterscheide ich die noch sauber? Deshalb zeigen wir meistens alles. Für eine Idee oder ein Konzept darf man es aber ruhig vereinfachen. Ich verteidige immer gern, dass wir vor allem für die Editor:innen gestalten, weil die Devs diese Sicht oft nicht haben. Aber neulich meinte jemand zu mir: Wenn es den Devs nicht gefällt, lehnen sie es einfach ab und bauen es mit einem anderen Tool.
- [JC] Oh stimmt, die haben am Ende die Entscheidungsmacht.
- [LH] Ja, am Ende könnte es einfach passieren, dass Entwickler:innen keine Lust mehr haben, damit zu arbeiten. Und das ist wirklich kritisch, weil man als Kunde TYPO3 ja nicht einfach selbst aufsetzen oder betreiben kann. Man ist immer auf Devs angewiesen. Wenn die das System komplett ablehnen, wird es schwierig.
- [JC] Für mich fühlt sich das nach einer ständigen Suche nach Kompromissen an. Irgendwo versucht man immer, die Mitte zu finden.
- [LH] Ja, voll.
- [JC] Ich glaube, wir sind damit schon fast am Ende. Eine letzte Frage hätte ich noch: Angenommen, Zeit und Ressourcen spielen keine Rolle, es gibt keine Einschränkungen. Du könntest dir einfach sagen: Die drei Sachen will ich auf jeden Fall schon in der v14. Was wären deine Favorites?
-

-
- [LH] Page Wizard. Die Quick Actions natürlich. Und ich hätte super gern, dass man auf der Content-Stage direkt den Content bearbeiten kann. Ich glaube, das wären meine drei.
- [JC] Bin ich voll bei dir, klingt richtig gut. Ich glaube, ich bin mit meinen Fragen durch. Hast du noch etwas, das du mich fragen möchtest?
- [LH] Eigentlich nicht. Aber wenn du Lust hast, dich auch in Zukunft weiter einzubringen: Der anderthalbjährige Zyklus dauert, man denkt da immer ziemlich weit voraus. Ich behalte dich auf jeden Fall im Hinterkopf, gerade beim Dashboard oder solchen Themen. Oder wenn wir Live-Editing und WYSIWYG angehen. Es ist einfach gut, wenn man sich austauschen kann. Wir sind nicht so viele, die daran arbeiten, und je mehr Perspektiven man hat, desto besser wird das Ergebnis.
- [JC] Sehr gern, meld dich jederzeit. Und du hast mir wirklich sehr geholfen, danke dir dafür. Das war richtig wertvoller Input. Ich freue mich total auf die neue Version und alles, was danach kommt. Das ist ja kein abgeschlossener Punkt, sondern eher ein fortlaufender Prozess.
- [LH] Genau, und das macht es spannend. Und es ist echt schön, wenn Leute Lust haben, sich damit zu beschäftigen. TYPO3 ist ja nicht unbedingt everybody's darling. Ich weiß nicht, wie lange du schon damit arbeitest. Aber wenn man lange dabei ist, wächst es einem irgendwie ans Herz. So eine leicht toxische Beziehung.
- [JC] Ja, es hat auf jeden Fall seinen eigenen Charme.
- [LH] Total. Und es kann einfach extrem viel. Wenn du dir andere CMS anschaust: Da kommt ständig etwas Neues, dann verschwindet es wieder. TYPO3 ist einfach da.
- [JC] Es ist irgendwie eine Konstante.
- [LH] Genau. Und du kannst dich darauf verlassen: Wenn du damit eine Website baust, läuft die auch noch in zehn Jahren. Klar, WordPress gibt es auch lange, aber selbst da merkt man Schwankungen. Dieses Gefühl von Stabilität und Langlebigkeit ist bei TYPO3 schon besonders. Und wenn du die richtigen Entwickler:innen hast, kannst du dir im Grunde alles bauen.
- [JC] Das ist schon ziemlich stark.
- [LH] Ja, absolut. Deshalb ist es schön zu wissen, dass noch mehr Leute Lust haben, sich einzubringen. Ich melde mich auf jeden Fall, wenn wir wieder an solchen Themen dran sind.
- [JC] Oh ja, sehr gerne. Dann wünsche ich euch fürs Release in den nächsten Wochen starke Nerven. Das wird auf jeden Fall ein richtig gutes Ergebnis. Ich freue mich schon drauf.
- [LH] Danke dir. Und viel Erfolg noch mit der Bachelorarbeit.
- [JC] Danke, das ist lieb. Ich wünsch dir noch einen schönen Tag.
- [LH] Wünsche ich dir auch.
- [JC] Mach's gut, bis dann. Tschüss.
- [LH] Bis dann, wir hören uns. Tschüss.
-