

Hochschule RheinMain
Fachbereich Design Informatik Medien
Studiengang Medieninformatik

Bachelor-Thesis
zur Erlangung des akademischen Grades
Bachelor of Science – B.Sc.

Konzeption und Entwurf einer Anwendung zur Verwaltung von Texten für Informations- und Kommunikationsmedien

vorgelegt von Markus Tacker
am 19. Juni 2012

Referent Prof. Dr. Jörg Berdux
Korreferent Prof. Thomas Steffen

Erklärung gem. ABPO, Ziff. 6.4.3

Ich versichere, dass ich die Bachelor-Thesis selbständig verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe.

Offenbach am Main, 19. Juni 2012

Markus Tacker

Verbreitung

Hiermit erkläre ich mein Einverständnis mit den im folgenden aufgeführten Verbreitungsformen dieser Bachelor-Thesis:

Einstellung der Arbeit in die Hochschulbibliothek mit Datenträger:	nein
Einstellung der Arbeit in die Hochschulbibliothek ohne Datenträger:	nein
Veröffentlichung des Titels der Arbeit im Internet:	ja
Veröffentlichung der Arbeit im Internet:	nein

Offenbach am Main, 19. Juni 2012

Markus Tacker

Satz

Gesetzt mit \LaTeX / XeTeX

Schrift: Museo Sans¹ und Vollkorn²

Diagramme

Erstellt mit Graphviz³ und Google Docs⁴

¹<http://www.exljbris.com/museosans.html>

²http://friedrichalthausen.de/?page_id=411

³<http://www.graphviz.org/>

⁴<https://docs.google.com/>

Ich danke

Prof. Dr. Jörg Berdux für die Betreuung dieser Bachelor-Thesis und meinem Korreferenten Prof. Thomas Steffen.

Meinen Interviewpartnern Arthur Blozyk, Carsten Fischer, Masanori Fujita, Jorinde Gessner, Sandra-Charlotte Hildebrandt, Eva Kümml, Jan Lochner, Sebastian Nell, Tobias Rudolphi, Markus Rüb, Torsten Schölzel und Marc Stenzel.

Den Professoren am Fachbereich Medieninformatik der Hochschule RheinMain, besonders Prof. Dr. Peter Barth, Prof. Dr. Jörg Berdux und Prof. Dr. Wolfgang Weitz für ihr tägliches Engagement für Lehre und Studenten.

Der Bundesrepublik Deutschland für das Aufstiegsstipendium.

Widmung

Ich widme diese Bachelor-Thesis meiner Frau Tanja die mich ermutigt hat, mit dem Studium zu beginnen.

Zusammenfassung

TODO

Inhaltsverzeichnis

1	Einleitung	I
2	Definition	2
3	Problem-Analyse	4
3.1	Die besondere Rolle von Text in Informations- und Kommunikationsmedien	4
3.2	Das Werkzeug der Wahl zur Verwaltung von Text: <i>Word</i> und <i>Excel</i>	6
3.3	Probleme bei der Verwendung von Textverarbeitungs- und Tabellenkalkulationsprogrammen im Verlauf eines Projektes	10
3.4	Beispiele aus der Praxis	15
3.4.1	Internetseite EnBW Transportnetze AG	15
3.4.2	Banner-Kampagne Nintendo	16
3.4.3	EA Phenomic: BattleForge	16
3.4.4	MAN Truck & Bus AG: Neufahrzeug-Konfigurator	16
3.5	Schlussfolgerung	17
4	Personas	18
4.1	Eva, Konzepterin	21
4.2	Lotte, Designerin	22
4.3	Torsten, Texter	23
4.4	Markus, Kunde	24
4.5	Jorinde, Übersetzerin	25
4.6	Jan, Produzent	26
4.7	Arthur, Projektleiter	27
5	Konzeption einer an den spezifischen Workflow angepassten Anwendung	28
5.1	Abzubildender Workflow	28
5.1.1	Inhalt	30
5.1.2	Attribute	30
5.1.3	Status	32
5.2	Anforderungen an die Anwendung	32
5.3	Art der Anwendung	34

5.4	Schnittstellen	36
5.5	Zugang	36
5.6	Überblick über den Aufbau des gesamten Systems	38
5.7	Zusammenfassung, Nachteile & Risiken des Konzepts	39
6	Entwurf des zentralen Systems	40
6.1	Entwurf eines browserbasierten GUI	41
6.1.1	Aufbau des GUIs	43
6.1.2	Definieren des Produktes	44
6.1.3	Texte erstellen	45
6.1.4	Texte übersetzen	46
6.1.5	Prüfen	47
6.1.6	Kontext-Informationen	48
6.2	Anbindung der GUI an den Server	49
6.3	Implementierung des GUI	50
6.3.1	Klassendiagramm	52
6.4	Entwurf des Anwendungsservers	53
6.4.1	API, Cronjobs, CLI, ...	53
6.4.2	Dependency-Injection-Container	53
6.4.3	Controller	53
6.4.4	Models	54
6.4.5	Persistenz, ORM	54
6.4.6	Import/Export & Benachrichtigung	54
6.4.7	Jobs	55
6.4.8	Workflow	55
6.4.9	Service-Adapter	55
6.5	Implementierung des Anwendungsservers	55
6.5.1	API	56
6.5.2	Persistenz	58
6.5.3	Klassendiagramm	59
6.6	Domänenmodell	60
6.7	Implementierter Workflow	62

6.8	Umsetzung am Beispiel des Studiengangsflyers	63
6.9	Zusammenfassung	63
7	Fazit	64
A	Screenshots	65
B	API-Endpunkte	69
B.1	Benutzer	69
B.2	Projekt	69
B.3	Elemente	69
B.4	Container	69
B.5	Texte	70
B.6	Text-Typen	70
B.7	Export	70

Abbildungsverzeichnis

1	Bei der Erstellung von Texten beteiligte Personen	5
2	Word-Dokument mit Texten für eine Internetseite	8
3	Excel-Dokument mit Texten für eine Internetseite	9
4	Übersicht über die Personas und den idealisierten Workflow	18
5	Einfluss auf den Status eines Textbausteines	33
6	Aufbau des System in stark vereinfachter Darstellung	35
7	Aufbau des gesamten Systems im Überblick	38
8	Komponenten des zentralen Systems	40
9	Aufbau des browserbasierten GUIs	43
10	GUI zum Bearbeiten der Produktstruktur	44
11	GUI zum Erstellen der Texte	45
12	GUI zum Übersetzen der Texte	46
13	GUI zum Überprüfen der Texte	47
14	Kontext-Informationen	48
15	GUI des Prototypen	51

16	GUI-Klassendiagramm des Prototypen	52
17	IO-Komponente	54
18	Persistierte Objekte im Prototypen	58
19	Anwendungsserver-Klassendiagramm des Prototypen	59
20	Domänenmodell	60
21	Registrierung für neue Benutzer	65
22	Login	65
23	Projekt anlegen	66
24	Projekt anzeigen, mit Darstellung des Projekt-Fortschritts	66
25	Produktstruktur definieren, vgl. 6.1.2 / S.44	67
26	Texte erstellen, vgl. 6.1.3 / S.45	67
27	Qualitätssicherung, vgl. 6.1.5 / S.47	68
28	Darstellung im Vollbild-Modus mit ausgeblendeten Seitenleisten, vgl. 6.1.1 / S.43	68

Tabellenverzeichnis

1	Kriterien von Textbausteinen und verantwortliche Personen	6
2	Interviewte Personen	19
3	Stärke des Einfluss, die Mitarbeiter in einem Projekt haben	29
4	Einfluss mit Gewichtung auf den Status eines Textbausteines	32
5	Umfang und Häufigkeit der Benutzung des browserbasierten GUIs bezogen auf die durchgeführten Operation	42

1 Einleitung

Die Herstellung von Informations- und Kommunikationsmedien ist heute ein zentraler Wirtschaftsbereich in Deutschland. Hunderttausende Unternehmen und Abteilungen haben es sich zur Aufgabe gemacht, Produkte zu erschaffen, die Werbebotschaften transportieren, oder Information zugänglich zu machen – dank des Siegeszug des Internets kann diese heute überall und zu jeder Zeit geschehen. Die technischen Möglichkeiten sind vielfältig wie nie. Es ist inzwischen normal, Videos in High-Definition auf Tablet-PCs über mobile Datenverbindungen zu streamen. Trotzdem können auch die modernsten Kommunikationsformen auf einen wichtigen Bestandteil nicht verzichten: *Text*. Denn gerade die Medienvielfalt, mit der Möglichkeit durch Multimedia-Inhalte, Interaktion und persönlich Ansprache den Rezipienten auf emotionaler Ebene anzusprechen schafft Raum für Interpretation – die einer Klärung bedarf, um Mißverständnisse zwischen Sender und Empfänger auszuschließen. Diese Funktion übernehmen »Sternchentexte« und das sprichwörtlichen »Kleingedruckte«. Text ist also weiterhin Kernbestandteil aller Medien, von der Beschriftung eines Buttons in einer SmartPhone-App bis hin zur Einladung für die nächste Aktionärsversammlung. Als universeller und verbindlicher Informationsträger ist es aber gerade dieser Bestandteil, der von den meisten Personen beeinflusst wird. Da die Erstellung von Medien-Produkten heute hohes technisches und fachliches Know-How erfordert, sind an diesen Projekten viele Personen beteiligt. Neben den beteiligten Mitarbeitern im Unternehmen des Auftraggebers haben auch die Mitarbeiter der beauftragten Werbeagenturen und Softwarehäusern ihre eigenen speziellen Anforderungen an die Texte des Produktes. Die stetig zunehmende Komplexität der Medien-Produkte und der Wunsch nach Kostenreduktion trägt ihren Teil dazu bei, dass der Kreis derer, die an der Erstellung eines Medien-Produktes beteiligt sind weiter wächst: Spezialisten sind für bestimmte Aspekte des Produkts zuständig und verstärken bei Bedarf als freie Mitarbeiter das Team. Der Austausch über die durch die Texte zu vermittelnden Inhalte und diese Anforderungen findet fast ausschließlich durch das Versenden von *Word*- und *Excel*-Dateien als Anhang von E-Mails statt – ein Prozess, der fehleranfällig, langwierig und chaotisch ist.

Die Suche nach einer Möglichkeit, diesen Prozess zu verbessern ist die Motivation für diese Bachelor-Thesis.

2 Definition

In dieser Bachelor-Thesis werden bestimmte allgemeine Begriffe und deren Synonyme verwendet, deren konkrete Bedeutung im Kontext dieser Arbeit wie folgt definiert ist:

Workflow (Ablauf) Die Automatisierung eines Business-Prozesses, als Ganzes oder in Teilen, in welchem Dokumente, Informationen oder Aufgaben entsprechend einer Menge von prozeduralen Regeln von einem zum anderen Teilnehmer zur Bearbeitung weitergegeben werden [15, S.8]. Allgemein lässt sich sagen, dass ein Workflow aus den zum Erreichen eines Zieles nötigen Arbeitsschritten besteht.

Text (Textbaustein) Damit sind die kleinsten sinnvoll identifizierbaren Bestandteile gemeint, aus denen sich der Text eines Produktes zusammensetzt. Dies sind in der Regel einzelne Sätze bei Druckmedien, können aber auch einzelne Worte sein, wie z.B. die Beschriftung einer Schaltfläche in einer Anwendung.

Medium (Medien-Produkt, Produkt) Medien sind physische oder elektronische Informationsträger. Diese Bachelor-Thesis beschäftigt sich vor allem mit Informations- und Kommunikationsmedien und hierbei vor allem mit Massenkommunikationsmitteln, sowohl physischer als auch technischer Natur (vgl. [13, S.199–201]). Dies können z.B. Marketingmedien wie Broschüren oder Fernsehwerbespots sein aber auch softwarebasierte Produkte wie eine Smartphone-Anwendung oder eine Internetseite.

Agentur Ein Unternehmen das Medien erstellt. In der Regel sind dies Werbeagenturen, Medien-Produktionsfirmen oder Software-Systemhäuser.

Projekt Die Erstellung von Medien erfolgt in Agenturen in Projektarbeit. Projekte sind zeitlich begrenzt und vereinen zielgerichtet die zur Erstellung des Produktes beteiligten Mitarbeiter und Ressourcen.

Kunde Ein Unternehmen das Agenturen mit der Erstellung von Medien beauftragt.

Nutzer Eine Person, die ein Medium konsumiert oder ein Produkt verwendet.

Werkzeug (Anwendung) Eine Software, die eine spezielle Funktion erfüllt. Adobe Photoshop ist ein Werkzeug zur Bearbeitung von Bildern.

Wireframes Wireframes dienen zur schnellen Skizzierung eines Produktes mit Hilfe von einfachen geometrischen Formen und Texten. Sie werden angefertigt um einen Überblick über den gesamten Umfang eines Produktes zu bekommen die wichtigsten Aufgaben zu definieren, die von dessen Nutzern durchgeführt werden.

Ansicht Bezeichnet eine einzelne Darstellungsvariante in einem Produkt die sich durch ihre Gestaltung und ihren Aufbau von anderen Ansichten unterscheidet.

3 Problem-Analyse

In diesem Kapitel werden die Probleme beschrieben, die bei der Erstellung von Informations- und Kommunikationsmedien in Zusammenhang mit den dargestellten Texten auftreten. Abschnitt 3.1 analysiert die besondere Rolle von Text, anschließend zeigt Abschnitt 3.3 / S.10 typische Probleme auf, die im Verlauf von Projekten entstehen. Abschnitt 3.4 / S.15 belegt dies mit Beispielen aus der Praxis.

Die Analyse des Problems basiert auf Interviews mit Personen, die in ihrem Arbeitsalltag regelmäßig mit Texten zu tun haben. Eine Liste der interviewten Personen findet sich in Tabelle 2 / S.19.

3.1 Die besondere Rolle von Text in Informations- und Kommunikationsmedien

Es existieren nahezu keine Medien, die ohne Texte auskommen, denn Text ist im Gegensatz zu Grafiken, Fotos oder Animationen ein eindeutiger Informationsträger und unterliegt viel weniger stark einer Interpretation durch den Rezipienten eines Mediums als die symbolisierte oder stilisierte Darstellung von Informationen in audiovisuellen Medien. Text wird in der Marketing-Kommunikation als Unterstützung der zu übermittelnden Information verwendet. Hat man die Aufmerksamkeit des Betrachter eines Produkts erlangt, liefert Text weitere Informationen zum Produkt, er dient dazu, die emotionale Botschaft zu erläutern und zu präzisieren. Auch aus rechtlichen Aspekten ist Text aus den genannten Gründen der einzige verbindliche Informationsträger – bestes Beispiel hierfür ist das sogenannte »Kleingedruckte«, dass sich gerade bei inhaltlich sehr stark komprimierten Werbeformen, wie z.B. Plakat- oder Fernsehwerbung, findet. Ist die Textmenge, die in der Marketing-Kommunikation zum Einsatz kommt, noch überschaubar, gibt es doch Medien die hauptsächlich aus Text bestehen. Hierunter fallen klassische Druckerzeugnisse wie Broschüren und Kataloge oder Produkte der Unternehmenskommunikation wie Jahresberichte und Pressemeldungen. Besonders digitale Medien werden oft mit großen Textmengen versehen – von der einfachen Produkt-Microsite, über Werbemittel wie Newsletter bis zur Unternehmenswebsite – die Möglichkeit Inhalte hierarchisch zu strukturieren und sogar über eine Suche zugänglich zu machen hebt eine Limitierung des Umfangs, wie bei Druckprodukten, praktisch auf.

Betrachtet man die Abläufe von Projekten, in deren Verlauf Medien erstellt werden, lassen sich bezüglich der Textbestandteile dieser Produkte immer wieder sehr ähnliche Vorgehensweisen und Besonderheiten beobachten. Aufgrund der verbindlichen Natur von Text sind an der Erstellung der Texte für das Medium mehr Personen beteiligt, als es z.B. für die Gestaltung, der Auswahl von Bildmaterial oder für die Programmierung der Fall ist, da Text sehr viele verschiedenen Kriterien erfüllen muss. Tabelle 1 / S.6 listet exemplarisch eine typische Gruppe von Personen auf, die im Verlauf des Projektes Einfluss auf den Text ei-

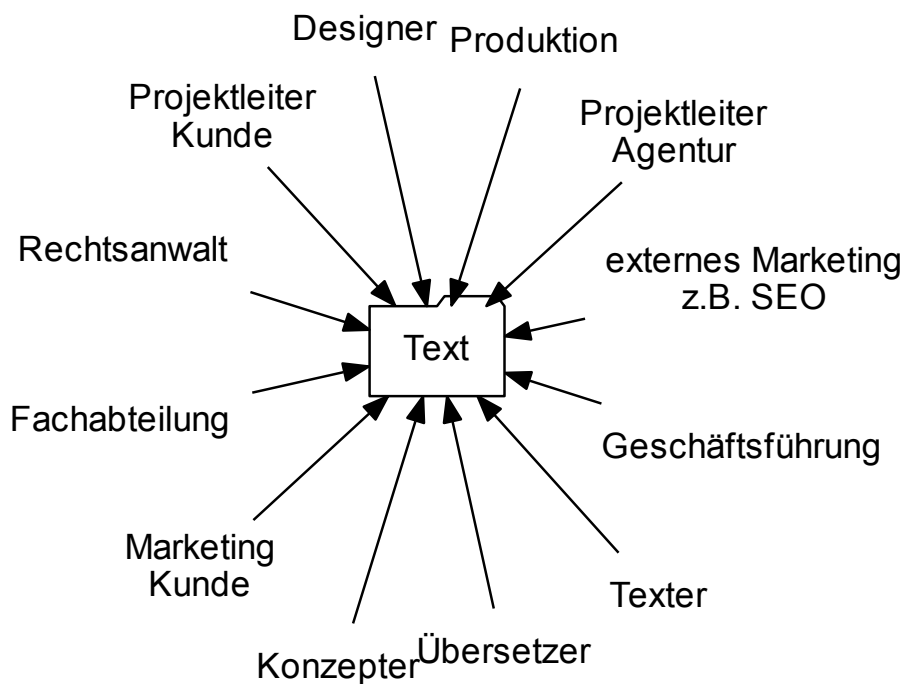


Abbildung 1: Bei der Erstellung von Texten beteiligte Personen

nes Produktes haben. Dieser Einfluss wird dabei in der Regel nicht in einer sinnvollen Reihenfolge und im Sinne des geplanten Projektverlaufes ausgeübt. Gerade auf die Mitarbeiter auf Kundenseite haben Agenturen keinen Einfluss; in Projektplänen lassen sich zwar verbindliche Termine für die Lieferung von Texten des Kunden festlegen, dies verhindert aber keinesfalls, dass zu einem späteren Zeitpunkt Änderungen notwendig werden – Hinweise von Anwälten sollten im besten Fall *vor* einer Übersetzung vorliegen, richtet sich aber nach ihren eigenen Terminplänen. Auch die Kriterien wie Text beeinflusst wird, sind sehr vielfältig: Im Entwurf und in der Umsetzung der Produkte legen Designer, Architekten und Produzenten die Struktur von Text wie Art der Ansprache, maximale Wortlänge, Anzahl der Wörter einer Überschrift fest oder diese werden durch das verwendete Medium vorgegeben, Texter legen die Inhalte fest, die wiederum durch Wünsche des Kunden beeinflusst werden; das Lektorat, Fachabteilungen und Anwälte begutachten die Texte dann bezüglich der jeweils erforderlichen Korrektheit.

Kriterium	Art	Verantwortlich
Aufgabenverteilung	Mitarbeiter	Projektleiter, Agentur
Zielgruppe	Struktur	Informationsarchitektur
Umfang, Satzlänge	Struktur	Art-Direktion
Länge einzelner Wörter	Struktur	Programmierer
Information	Inhalt	Texter
Orthographie	Korrektheit	Lektorat
Übersetzung	Sprache	Übersetzungsbüro
Suchmaschinen-Optimierung	Inhalt	SEO-Experte
Aufgabenverteilung	Mitarbeiter	Projektleiter, Kunde
Fachliche Aspekte	Korrektheit	Fachabteilung
Rechtliche Aspekte	Korrektheit	Rechtsanwalt
Werbeaussagen	Inhalt	Marketingabteilung
...

Tabelle 1: Kriterien von Textbausteinen und verantwortliche Personen

Wie man Tabelle 1 entnehmen kann, existieren vielfältige Einflussmöglichkeiten auf die Gestaltung von Texten für Medien die sich auf viele Verantwortliche verteilen. Der Grund dafür ist, dass alle Beteiligten jeweils spezifisches Fachwissen in den Text einfließen lassen, seien es gestalterische Aspekte, die Einfluss auf die Struktur haben, oder das Wissen über exakte technische Abläufe, die nur Spezialisten in den Fachabteilungen auf Kunden-seite bekannt sind. Dieses Expertenwissen kann nicht für die meist kurze Projektlaufzeit an die umsetzenden Agentur vermittelt werden. Es ist also unvermeidlich, dass Text während des gesamten Projektverlauf geändert werden kann. Neben den Einflüssen durch Experten gibt es auch projektbedingte Einflüsse auf Text in letzter Minute. Sind in Texten Informationen enthalten sind, die einen zeitlichen Aspekt abbilden, ergeben sich durch Verzögerungen im Projekt automatisch Änderungsanforderungen. Ein Beispiel sind Gewinnspiele: Verschiebt sich durch Probleme während dem Projekt der Zeitpunkt, ab dem ein Produkt beim Rezipienten vorliegt, müssen auch eventuell knapp kalkulierte Gewinnspieltermine angepasst werden. Ein weiterer Grund für vielfältige Textänderungen im Verlauf eines Projektes ist die Erwartungshaltung des Kunden – da es Kunden aus ihrem eigenen Arbeitsalltag gewöhnt sind, mit Textverarbeitungsprogramme zu arbeiten, und sie so aus eigener Erfahrung vermeintlich wissen dass Texte schnell geändert sind, erwarten sie auch, dass die Texte im Produkt bis zum Schluss geändert werden können; ihnen ist nicht bewusst, dass vom ursprünglichen Text im Quelldokument bis zur Darstellung im fertigen Produkt viele aufwändige Arbeitsschritte nötig sein können.

3.2 Das Werkzeug der Wahl zur Verwaltung von Text: *Word* und *Excel*

Zur Abbildung der komplexen Abläufe bei der Erstellung von Informations- und Kommunikationsmedien liefern etablierte Software-Hersteller passende Lösungen auch speziell

für Texte: Mit *InCopy* liefert *Adobe* eine „Lösung für Texterstellung und -bearbeitung, die aufgrund der engen Integration mit *Adobe InDesign*® CS5.5 effektivere Zusammenarbeit zwischen Redakteuren und Layoutern ermöglicht“ [1] und die *Content Station* von *Woodwing* „ist [...] eine einzige Oberfläche für alle Schritte des Publishing-Prozesses. [...] Unter Nutzung der Desktop- oder der Web-Version können die Team-Mitglieder unabhängig ihres Aufenthaltsorts mitarbeiten“ [14] – um nur zwei Beispiele zu nennen. Doch obwohl spezialisierte Werkzeuge existieren findet man diese in Agenturen nur selten – das Werkzeug der Wahl zur Verwaltung der Texte ist in der Regel eine in der Agentur vorhandene Textverarbeitungs- oder Tabellenkalkulationssoftware, in den allermeisten Fällen handelt es sich dabei um den Marktführer in diesem Bereich: *Microsoft Word* oder *Excel*. Auf die Probleme, die durch deren Einsatz entstehen wird im nachfolgenden Abschnitt 3.3 / S.10 eingegangen. Zu erst muss jedoch erst untersucht werden, warum statt spezieller Werkzeuge, die für den komplizierten Workflow in Projekten entwickelt wurden, *Word* oder *Excel* eingesetzt werden.

Oberflächlich betrachtet, bieten Textverarbeitungsprogramme die notwendigen Funktionen, um Texte zu verwalten und sind damit scheinbar die natürliche Wahl. Die verwendeten Funktionen sind dabei nachfolgend beschrieben.

Strukturierung von Texten Die Möglichkeit, Texte hierarchisch in Dokumente, Seiten, Kapitel, Abschnitte oder Absätze zu unterteilen ermöglicht es die Textbausteine für ein Produkt geordnet zu Erfassen. Neben den eigentlichen Texten lassen sich dazu auch Zusatzinformationen wie die Klasse des Textes dort zuzuordnen. Abbildung 2 / S.8 zeigt beispielhaft ein *Word*-Dokument, in dem die Texte für eine Website definiert werden. Im Dokument existiert pro Seite der Internetpräsenz jeweils ein Abschnitt, der alle Texte auf der Seite beschreibt. Dort finden sich die Texte zu den Platzhaltern, die in der Website verwendet werden, die dann an deren Stelle eingefügt werden. Über die Formatierung der Überschriften wird die Hierarchie der Texte definiert. Die Verwendung von Tabellen statt Dokumenten ist eine weitere Möglichkeit die verwendeten Texte zu erfassen. Abbildung 3 / S.9 zeigt beispielhaft ein *Excel*-Dokument, in dem pro Zeile ein Text definiert wird. In den Spalten findet sich neben dem eigentlichen Text Zusatzinformationen wie z.B. die Text-Klasse. Tabellarische Dokumente werden oft bei umfangreichen Projekten oder um mehrere Sprachversionen zu verwalten verwendet.

Rechtschreibkorrektur In Textverarbeitungsprogrammen sind ausgefeilte Funktionen zur Rechtschreibkorrektur enthalten, die bereits während der Eingabe auf Fehler aufmerksam machen und für viele Sprachen verfügbar sind. So ist sichergestellt, dass bereits die erste Version eines Textes relativ wenige Fehler enthält.

Kommentare Es ist möglich in *Word*- und *Excel*-Dokumenten Kommentare zu hinterlassen. Diese werden gesondert hervorgehoben und können zum Austausch über den Text

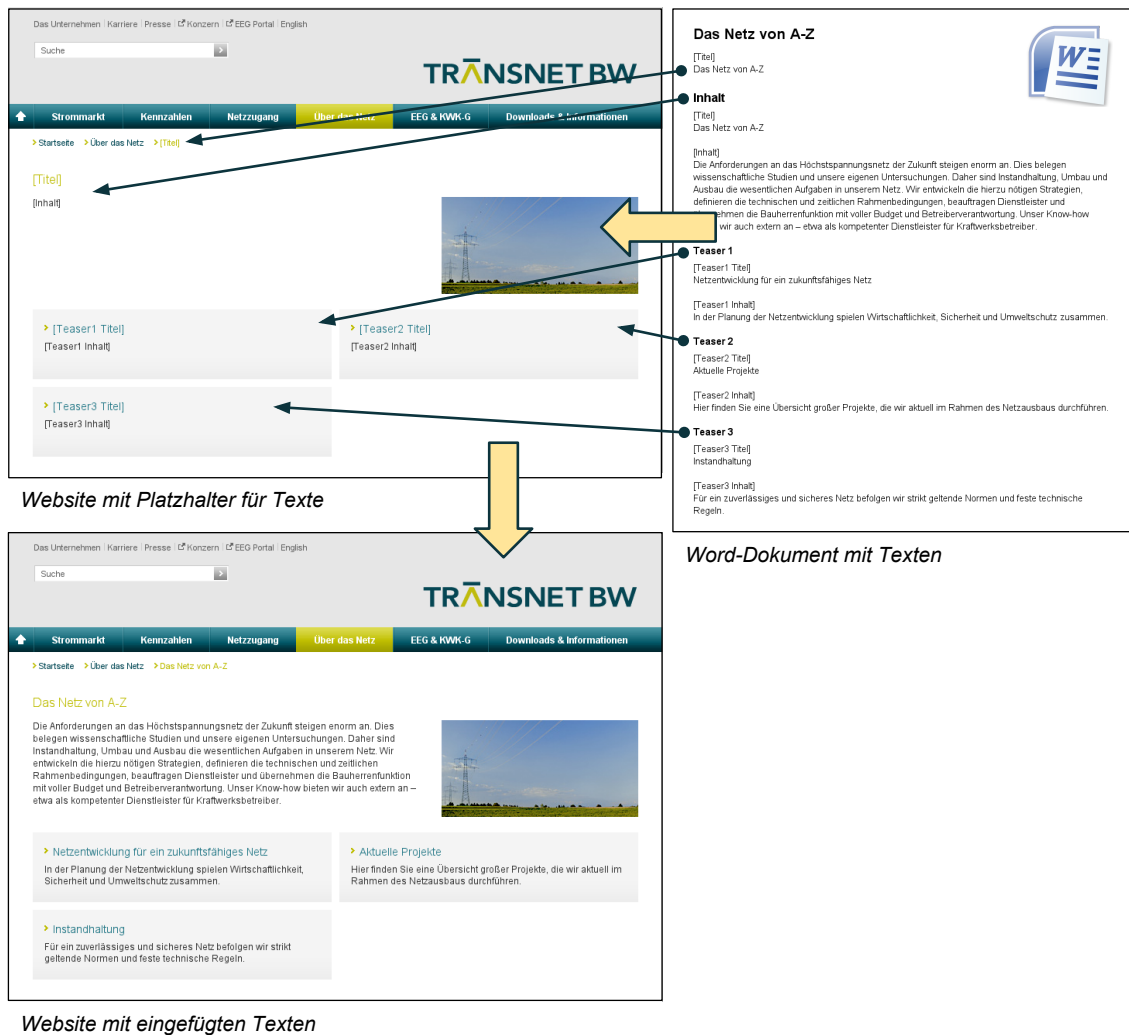


Abbildung 2: *Word-Dokument mit Texten für eine Internetseite*

oder für Hinweise zur Verwendung hinterlegt werden und von allen Bearbeitern eingesehen werden.

Änderungsverfolgung Wenn die Änderungsverfolgung aktiviert ist, werden alle Änderungen an einem Dokument aufgezeichnet. Diese Information kann dabei helfen, mehrere Versionen eines Dokumentes manuell zusammenzuführen oder Änderungen an Inhalten vorzuschlagen, zu prüfen und selektiv zu übernehmen.

Verzeichnisse In *Word*-Dokumenten ist es möglich, Verzeichnisse wie z.B. ein Inhaltsverzeichnis anzulegen. Dies hilft dabei, bei größeren Projekten einen Überblick über den

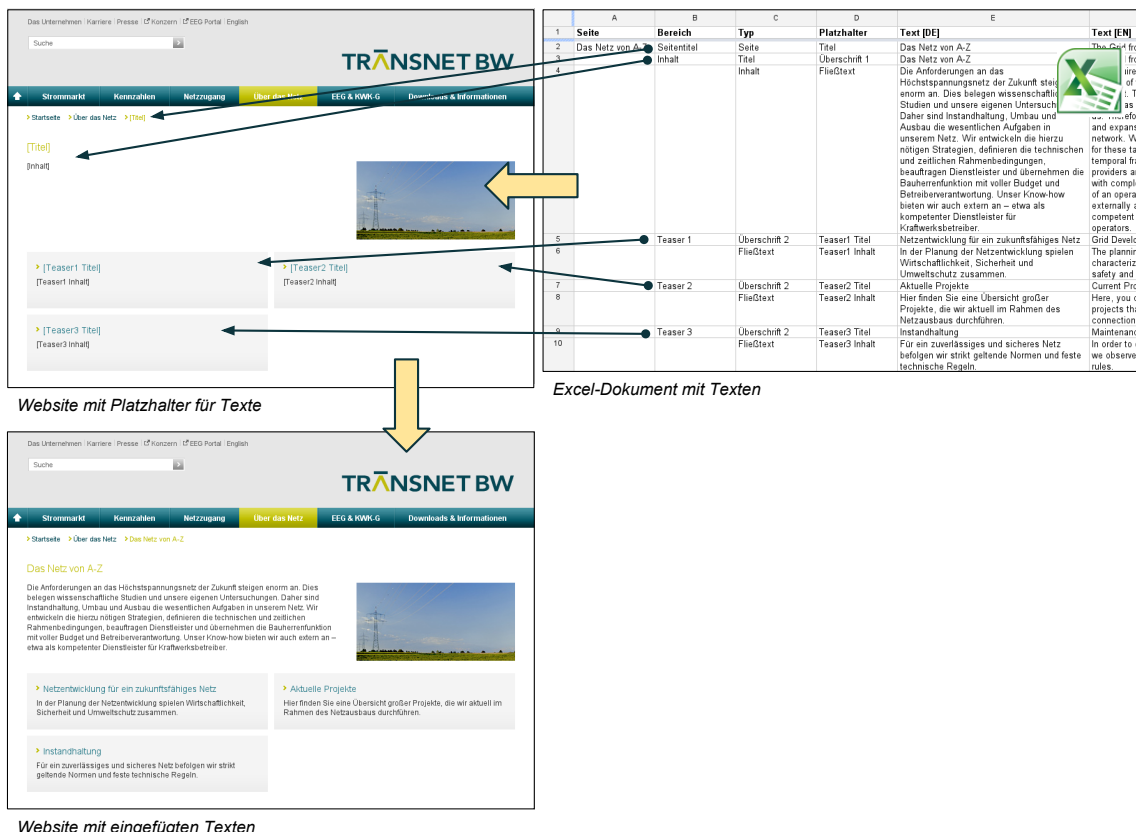


Abbildung 3: Excel-Dokument mit Texten für eine Internetseite

Aufbau des Produktes zu erhalten, sofern die Inhalte mit den passenden Formatvorlagen versehen wurden.

Suchen & Ersetzen Da sich die Texte in einem großen Dokument befinden können mit den Funktionen zum Suchen & Ersetzen schnell bestimmte Inhalte gefunden und angepasst werden.

Export Die *Office*-Programme verfügen über die Möglichkeit des Exports in verschiedene Formate. Verwendet werden vor allem PDF bei *Word*-Dokument zur Abstimmung, unter andere auch deswegen, weil es in *Adobe Acrobat* umfangreiche Korrekturfunktionen gibt, und CSV in *Excel*, das möglich macht die Text mit Hilfe von speziellen Programmen in andere Systeme zu importieren.

Formatierungsfunktionen Mit umfangreiche Formatierungsfunktion ermöglichen zusätzliche Informationen zu Texten zu hinterlegen. Oft werden durch farbige Markierung Passagen markiert, die zu überarbeiten sind, entfallen oder inhaltlich überarbeitet werden.

den müssen. Auch können Formatieren so angelegt sein, dass sie in das Produkt übernommen werden sollen – in der Regel werden dann Teile des oder einzelne Wörter Textes fett oder kursiv formatiert. Funktionen zum Setzen von Hyperlinks werden gerade bei Web-Projekten verwendet um Links zu definieren, die im Produkt verwendet werden sollen.

Auch im Hinblick auf nicht-funktionale Aspekte bieten Textverarbeitungsprogramme einige Vorteile, sind sie doch in den allermeisten Unternehmen der Standard zur Textverarbeitung und sogar plattformunabhängig verfügbar – zumindest existiert die Möglichkeit das *Microsoft Office*-Dateiformat auf allen Plattformen zu bearbeiten. Da bei allen Projektbeteiligten eine Installation von *Microsoft Office* vorausgesetzt werden kann, werden *Word* und *Excel* zu »leichtgewichtigen« Werkzeugen, das vom Anwender keine zusätzlichen Aufwände, z.B. bei der Installation oder Eingewöhnung, abverlangen. Selbst auf Plattformen die von *Microsoft Office* nicht offiziell unterstützt werden, wie z.B. Linux, existieren Programme mit denen das Office-Dokumenten-Format geöffnet und bearbeitet werden kann, unter Linux ist dies z.B. *Libre Office*. Da *Office*-Dokumente in nur einer Datei gespeichert werden, sind diese einfach auszutauschen – in Agenturen werden die Dateien in der Regel auf einem Netzwerk-Laufwerk gespeichert, unternehmensfremde Mitarbeiter erhalten die Dateien via E-Mail, FTP-Server oder Filesharing-Anbieter, z.B. Dropbox. So wird das gemeinsame Arbeiten an den Texten, zumindest nacheinander, möglich.

Wie in diesem Abschnitt gezeigt wurde, sind Textverarbeitungs- und Tabellenkalkulationsprogramme wie *Microsoft Word* und *Excel* nominell für den Einsatz zur Verwaltung von Texten für Medien geeignet. Dies erklärt, warum sie zu Beginn eines Projektes als geeignet angesehen und in Agenturen immer wieder als Werkzeug für die Erfassung, Definition und Übersetzung der Texte eines Projektes ausgewählt werden. Im alltäglichen Gebrauch treten jedoch Probleme gerade im Bereich des gemeinsamen Bearbeitens, paralleler oder nachträglicher Änderungen und der Übertragung der fertigen Texte in den Produktionsprozess auf, die im folgenden Abschnitt erläutert werden.

3.3 Probleme bei der Verwendung von Textverarbeitungs- und Tabellenkalkulationsprogrammen im Verlauf eines Projektes

Wie im vorangegangenen Abschnitt gezeigt wurde, sind Textverarbeitungs- und Tabellenkalkulationsprogramme wie *Microsoft Word* und *Excel* der Standard für die Verwaltung von Texten in Projekten zur Erstellung für Informations- und Kommunikationsmedien. In den für diese Bachelor-Thesis geführten Interviews haben jedoch alle Personen von vielfältigen Problemen in Zusammenhang mit diesen Programmen berichtet. Dies belegt zum einen, dass am gängigen Workflow viele Möglichkeiten zur Verbesserung existieren und liefert zum andere auch Hinweise, wie der verbesserte Workflow im Detail gestaltet werden muss. In diesem Abschnitt werden die beobachteten Probleme beschrieben.

Serielles Bearbeitungskonzept Das grundsätzliche Bearbeitungskonzept, das in *Word* und *Excel* zum Einsatz kommt ist seriell, das bedeutet, dass eine Dokument gleichzeitig nur von einer Person bearbeitet werden kann. Soll mit mehreren Personen an einem Dokument gearbeitet werden, muss dieses zwischen allen Beteiligten ausgetauscht werden. Dies geschieht, in dem die Dokumenten-Datei entweder per E-Mail jeweils zum nächsten Bearbeiter verschickt wird oder die Datei auf ein, durch alle Beteiligten erreichbaren Speicherplatz verschoben werden. In Agenturen handelt es sich hierbei meistens um eine Netzwerklaufwerk – hierauf haben aber nur die im lokalen Netzwerk integrierten Mitarbeiter Zugriff. Soll die Datei auch externen Mitarbeitern oder dem Kunden zur Verfügung gestellt werden, werden diese wieder per E-Mail verschickt oder in extern erreichbare Speicherort kopiert, wie z.B. FTP-Server, Wikis und Extranet-Portale oder es kommen spezielle Programme zum Dateiaustausch zum Einsatz, wie z.B. *Dropbox*. Die Organisation dieses Austausches ist besonders dann aufwändig, wenn Dateien sich nicht mehr unter Kontrolle der Agentur befinden, weil sie z.B. zum Kunden zur Kontrolle geschickt wurden. Dann kommt es dazu, dass mehrere Versionen des Dokumentes parallel existieren: eine Version beim Kunden, die dort mit Änderungen und Ergänzungen versehen wird und eine Version in der Agentur in der sich aufgrund von Änderungen im Verlauf des Projekts Text ändern. Um anschließend alle Beteiligten auf den aktuellen Stand zu müssen die verschiedenen Versionen des Dokumentes manuell zusammengeführt werden – automatisiert ist das mit *Word* und *Excel* nicht möglich. Neben dem zeitlichen Aufwand birgt das manuelle Zusammenführen weitere Fehlerquellen, da Änderungen an Texten durch Copy&Paste übertragen werden, kann es gerade bei großen Dokumenten passieren, dass man die Änderungen an der falschen Stelle einarbeitet, sofern im Dokument sich ähnelnde Textabschnitte existieren. Aus Kostengründen und weil es sich dabei um eine repetitive Arbeit handelt ist es nicht selten der Fall, dass diese Änderungen von Praktikanten oder studentischen Aushilfen durchgeführt werden, die mangels inhaltlicher Kenntnis den Zusammenhang der Text nicht kennen, was ebenfalls die irrtümliche Änderungen am Text durch fehlerhaftes Copy&Paste begünstigt. Auch ist die Eindeutigkeit der Dateiversionen nicht gewährleistet, allein aufgrund des Zeitstempels kann keine genaue Aussage darüber getroffen werden, welche Datei die neueste ist. So muss man sich auf ein Benamungsschema für Dateien einigen, das im besten Fall klar erkennen lässt welches Dokument das neueste ist. Üblich sind dabei Ergänzungen des Dateinamens mit Datumsinformationen, wie z.B. Text_Online_2012-04-13.docx. Folgen nicht alle Beteiligten diesem Schema, weil diese z.B. nicht ausreichend informiert sind, oder kommt es zu gleichzeitiger Änderungen von zwei Personen kann es so auch zu zwei verschiedenen Dateien mit dem gleichen Dateinamen kommen. Durch die Verteilung der Dokumente an verschiedene Speicherorte kommt es zu Situationen, in denen nicht klar ist, wer aktuell die *neueste* Dateiversion hat.

Tatsächlich existiert mit dem *Microsoft SharePoint*, einer Software für Intra-, Extra- und Internetportale, eine Lösung, die diese Probleme behebt [8]. Mit Hilfe von *Shared Documents* lassen sich Dokument zentral ablegen. Sollen diese editiert werden, müssen Sie von der jeweiligen Person »ausgecheckt« werden, dies sperrt den Zugriff auf das Dokument

durch andere Mitarbeiter. Sobald das Bearbeiten abgeschlossen wurde, wird die Datei wieder »eingespeichert«. So wird sichergestellt, dass es nie zwei Versionen der Datei mit unterschiedlichen Änderungen gibt. Diese Funktion behebt aber nicht den Umstand, dass an einem Dokument immer nur eine Person arbeiten kann. Des Weiteren ist *SharePoint* im Agentur-Umfeld kaum anzutreffen, was dem Umstand geschuldet ist, dass der Betrieb einer *SharePoint*-Instanz mit hohen Lizenz- und Personalkosten verbunden ist. Zum anderen setzen die Funktionen zum gemeinsamen Bearbeiten von *Word* oder *Excel* voraus, dass alle Mitarbeiter über die neuesten Programm-Version verfügen [5]. Dies kann aber auf Kundenseite nicht vorausgesetzt werden. Aufgrund der unterschiedlichen Programm-Versionen aber auch unterschiedlicher Betriebssystemen, kommt es beim Austausch der Dateien zu verschiedenen Kompatibilitätsproblemen, da in Agenturen meistens *Mac OS* verwendet wird, auf Kundenseite jedoch *Windows* verbreitet ist.

Monolithische Dokumente Das Zusammenführen aller Textbausteine eines Produktes in einem Dokument hat den Nachteil, dass diese nur als Ganzes weitergeben werden können. In bestimmten Konstellationen ist es aber notwendig, sicherzustellen, dass bestimmte Inhalte nicht von allen Projektbeteiligten einsehbar sind. Zum einen kann es sich dabei um Informationen handeln, die der Geheimhaltung unterliegen oder sensibel sind, so dass sie nur bestimmten Personen zugänglich sein dürfen. Zum anderen kann es aus Kostengründen sinnvoll sein, die Prüfung von Texten durch Anwälte, oder die Übersetzung von Texten auf bestimmte Bereiche einzuschränken. In diesen Fällen wird es notwendig, verschiedene Versionen des Dokumentes anzulegen, die an den jeweiligen Personenkreis angepasst sind. Dies erzeugt die Problem, die im vorangegangenen Abschnitt beschrieben wurden.

Feedback Durch das Verteilen der Dokumente auf verschiedene Speicherorte wird eine parallele Kommunikation des Arbeitsstandes mittels E-Mail nötig, bei der jeweils dem nächsten Bearbeiter mitgeteilt wird, dass er mit seiner Aufgabe weiter fortfahren kann. Der Ablauf und die Reihenfolge der Kommunikation ergibt sich durch die Aufgaben der beteiligten Personen, aber auch durch informelle Absprachen. Gerade zwischen Agentur und Kunden gibt es häufig »Flaschenhälse«, die zu Verzögerungen führen. Dies sind in den meisten Fällen die jeweiligen Projektleiter und Ansprechpartner, die auch bei technischen oder inhaltlichen Fragen jeweils der alleinige Empfänger sind, die Anfrage entgegen nehmen in in ihrem Unternehmen an die zuständige Person weiterleiten, auf deren Antwort warten um dann die Antwort zurück zu spielen. Hierdurch bilden sich umfangreiche und langlebige E-Mail-Kommunikationsketten, an denen viele, meistens zu viele, Personen beteiligt sind und in ungeordneter Reihenfolge Feedback liefern.

Strukturierung von Dokumenten Eine der Gründe, warum *Word* und *Excel* zur Standardausstattung auf jedem Büro-Computer gehören, ist der, dass sie für einen sehr breiten Anwendungsbereich entwickelt wurden. Dies hat jedoch zur Folge, dass es mit einigem Aufwand verbunden ist, die passende Struktur für die Inhalte eines Produktes in einem Text-

oder Tabellen-Dokument anzulegen. Hierbei wird meistens eine hierarchische Struktur mit Hilfe von Abschnitten angelegt (vgl. Abb. 2 / S.8). In *Excel* wird im Hinblick auf die kompaktere Darstellung meisten aus besonderer Formatierungen verzichtet und mit sich wiederholenden Zellen gearbeitet (vgl. Abb. 3 / S.9), die eine hierarchische Struktur simulieren – die zweidimensionale Tabellendarstellung ist für komplexere Hierarchien nicht ausgelegt. Dies aufwändige Strukturierung des Dokuments muss auch geschehen, damit sich alle Anwender in den Dokumenten zurecht finden und eine eindeutige Zuordnung zwischen Textbausteinen im Dokument und den dafür vorgesehen Platzhaltern im fertigen Produkt möglich ist. Da beide Programme für diese Aufgabe keinerlei Vorlage und Unterstützung liefern, muss hier viel Arbeit investiert werden, die zudem noch vorausschauend genug sein muss, damit es im späteren Verlauf des Projektes durch nicht berücksichtigte Fälle nicht notwendig wird, das Dokument komplett zu überarbeiten.

Formatierung Die Formatierung der Textbausteine nach gestalterischen Aspekten, also das Hinzufügen von z.B. Hervorhebungen, Unterstreichungen und Absätzen wird zum Teil schon während der Erstellung der Texte vorgenommen. Hierbei werden die jeweiligen Funktionen der von *Word* und *Excel* verwendet. Ist dies in *Word* komfortabel möglich, sind die Möglichkeiten in *Excel* deutlich eingeschränkt. Hier lassen sich Zeichen-Formatierungen wie Hervorhebung, Farbe o.ä. nicht auf einzelne Worte oder Zeichen anwenden, sondern nur auf eine ganze Zelle. Auch Zeilenumbrüche stellen ein Problem dar. Diese sind zwar grundsätzlich möglich, jedoch kann es mangels Wissen dazu kommen, dass ein Bearbeiter einen Zeilenumbruch nicht innerhalb einer Zelle einfügt, sondern statt dessen eine neue Zeile einfügt. Werden die Zeilennummern oder bestimmte Spalten als Referenz-Schlüssel für den Text verwendet, führt das dazu, dass die zweite Zeile des Textes, nicht mehr zugeordnet werden kann um im Produkt fehlt. Werden Tabellen in *Word*-Dokumenten verwendet um z.B. tabellarische Inhalte in einem Produkt zu beschreiben, führt die aufgrund der beschränkten Seitengröße eines Textdokuments dazu, dass Tabellen mit vielen Spalten nur mit sehr kleinem Text dargestellt werden können, was das Bearbeiten der Texte schwierig macht. Üblich ist auch das Einfügen von Bildern, dies ist nötig, um die Zuordnung der Texte zu Produkten zu erleichtern oder um Untertitel Fotos zu definieren, hierbei treten dann zusätzliche Formatierungsproblem auf, da das platzieren von Bildern nur in beschränktem Maße beeinflusst werden kann. Problematisch ist auch der Austausch der Formatierungen zwischen verschiedenen Versionen von *Word* oder *Excel*, besonders wenn Dokumente von neueren Versionen in älteren Versionen angezeigt und bearbeitet werden, und vor allem dann, wenn die Dokumente in anderen Textverarbeitungs- und Tabellenkalkulationsprogramm wie z.B. *Apple iWorks* oder *LibreOffice* bearbeitet werden. Diese Programme unterstützen zwar *per se* den Ex- und Import von *Microsoft*-Dateiformaten, bei der Konvertierung entstehen gerade bei Übertragen dieser Formatierungen Unsauberkeiten. Sollen die Formatierungen dann in das Produkt übernommen werden kann dies nicht automatisch geschehen, da es für die verwendete Auszeichnungen keinen Standard gibt, der in allen Umgebungen verwendet werden kann. Deswegen müssen alle Formatierung manuell im

Produkt angelegt werden, dies kann, ähnlich wie bei Copy&Paste, zu Übertragungsfehlern führen.

Feedback und Kommunikation Neben den eigentlichen Textbausteinen werden in *Word*- und *Excel*-Dokumenten auch Hinweise und Kommentare zu den Texten mit Hilfe von Notizen oder direkt in das Dokument platzierten, besonders formatierten Texten hinterlegt. Werden Notizen verwendet besteht zum einen das Problem, dass diese an einer spezifischen Stelle im Text platziert werden, wird diese Stelle gelöscht, wird damit auch die Notiz ohne einen Warnhinweis gelöscht. Bei der gleichzeitigen Darstellung der Notizen in Kombination mit der Anzeige von Änderungen durch andere Bearbeiter kann das Dokument sehr unübersichtlich werden. Werden Hinweise als Text im Dokument hinterlegt kann diese dazu führen, dass diese Hinweise übersehen werden, oder beim Copy&Paste von großen Abschnitten unbeabsichtigt in das Produkt übernommen werden. Diese Unzulänglichkeiten führen dazu dass Feedback auch parallel zu den Dokumenten ausgetauscht wird, meistens mit Hilfe von E-Mails in denen die Anmerkungen bzw. Änderungswünsche aufgezählt werden. Bei der Übersetzung von Texten kommt es mitunter vor, dass Hinweistexte, die zur z.B. Abschnitte kennzeichnen, übersetzt werden, da extern beauftragte Übersetzer sich mit den verwendeten Dokumentenvorlagen nicht auskennen. So werden dann auch beschreibende Texte wie „*Überschrift:*“ übersetzt, was zur Folge hat, dass das Dokument gänzlich unleserlich wird. Übersetzer erhalten auch oft nicht wichtige Zusatzinformationen zu Texten, wie z.B. Angaben über maximale Satzlängen, da sie die fertigen abgenommenen Texte in der Ausgangssprache in einem eigenen Dokument erhalten und bei der Erstellung des Dokuments diese, vermeintlich *internen* Hinweise, oft nicht mit übernommen werden. Informationen darüber, welche Teile zuletzt geändert wurden, können in *Word* und *Excel* zwar aufgezeichnet werden, diese Funktion muss aber Explizit vom Überarbeiter aktiviert werden. Wird dies versäumt kommt es dazu, dass Änderungen manuell durch Vergleichen von zwei Dokumenten-Version identifiziert werden müssen, wenn es nicht sinnvoll ist, alle Texte eines Produktes noch einmal zu ersetzen.

Usability-, technische und typografische Probleme Die bisher genannten Punkte beschreiben Probleme die im Workflow entstehen, wenn *Word* und *Excel* eingesetzt werden. Aber auch in der Verwendung dieser Programme existieren weitere Probleme, die sich negativ auf die Arbeitsgeschwindigkeit auswirken. Zu den wichtigsten Usability-Probleme zählt, dass die Programme nicht für das gleichzeitige Arbeiten in mehreren Dokumenten ausgelegt sind, dies kommt besonders dann zum Tragen, wenn zwei Dokumente miteinander verglichen werden sollen; was z.B. beim Kontrollieren einer Übersetzung der Fall ist. Dann müssen das Originaldokument und die Übersetzung nebeneinander in zwei Fenstern geöffnet werden. Zum Vergleichen der Texten muss dann abwechselnd in diesen beiden Fenstern gescrollt werden. Kleinere Usability-Probleme ist etwa, dass das Anzeigen der Anzahl der Zeichen in einem Satz nur via Kontextmenü zu erreichen ist, diese Funktion aber häufig verwendet wird. In den Office-Produkten ist es auch nicht möglich, weiche Zei-

len Umbrüche fest zu legen. Diese bestimmen in einem Satz die Wortzwischenräume, an denen der Text umgebrochen werden *kann*, sollte er über mehrere Zeilen laufen. Da Umbrüche sprachspezifischen Regeln folgen müssen, diese Informationen bereits während der Übersetzung hinterlegt werden, später in der Produktion werden sonst im Zweifelsfall die Umbrüche willkürlich gesetzt. Ein Problem technischer Art ist, dass sich bei Dokumenten ab etwa 200 Seiten die Reaktionszeit der Anwendung merkbar verringert.

Workflow Die Abwicklung des Workflows durch den Austausch von Dokumenten erfolgt nicht in einer geordneten Art und Weise. Die Reihenfolge, wer wann an den Dokumenten arbeitet ergibt sich organisch, da sich eine Reihenfolge nicht in den Dokumenten festlegen lässt. Soll ein gewisser Ablauf festgelegt werden, bei dem eine Vorbedingung erfüllt sein muss, bevor der nächste Mitarbeiter weiter arbeiten kann, muss dies parallel definiert und mit allen Beteiligten vereinbart werden. Die Kontrolle dieser Vereinbarung liegt auch außerhalb der Dokumente und muss durch die einzelnen Mitarbeiter sichergestellt werden. Dieser Umstand, und vor allem die Tatsache, dass mündliche oder schriftliche Vereinbarungen in der Praxis immer wieder umgangen werden, führt letztendlich dazu, dass im Verlauf eines Projektes immer wieder Schleifen entstehen, die Mehrarbeit erzeugen.

3.4 Beispiele aus der Praxis

Konkrete Beispiele für die Verwendung von *Word* und *Excel*, auch in größeren Projekten, liefert dieser Abschnitt.

3.4.1 Internetseite EnBW Transportnetze AG

Bei *Scholz & Volkmer* wurde im Rahmen der Auslagerung des Geschäftsbereiches Transportnetze der EnBW Energie Baden-Württemberg AG ein neues Informationskonzept für die Internetseite des neuen Unternehmens *EnBW Transportnetze AG*⁵ erarbeitet. Hierzu wurden die bestehenden Inhalte, die sich auf etwa 100 Seiten verteilten, analysiert und überarbeitet. Die überarbeiteten Texte wurden direkt in ein CMS übertragen und der neuen Struktur der Internetseite zugeordnet, die dann aus etwa 300 Einzel-Seiten bestand. Zur Abstimmung mit dem Kunden und den Fachabteilungen wurden aus dem CMS sogenannte »Content-Booklets« als *Word*-Dokument generiert (siehe Abbildung 2 / S.8), die dann dem Kunden zur Abstimmung via E-Mail zur Verfügung gestellt wurden. Da die Freigabe des Booklets nicht auf einmal sondern nur Stück für Stück freigegeben wurde, mussten auch das externe Übersetzungsbüro nacheinander mehrere Dokumente übersetzen. Die freigegebenen und übersetzten Texte wurden dann von studentischen Aushilfen wieder mit Copy&Paste im CMS korrigiert bzw. eingetragen. Die größten Probleme in diesem Projekt war der zusätzliche Arbeitsaufwand, um die *Word*-Dokumente zu erzeugen, der Zeitauf-

⁵<http://enbw-transportnetze.de/>

wand beim Einpflegen der neuen Texte und Übersetzungen und die Verwaltung der abgenommenen und nicht abgenommen Teile der Texte.

3.4.2 Banner-Kampagne Nintendo

Für ein Computerspiel lässt *Nintendo* Flash-Werbebanner in verschiedenen Formaten, mit unterschiedlichen Motiven und Sprachvarianten anfertigen. Die mit der Umsetzung beauftragten Mitarbeiter bekommen hierfür die Texte in Word-Dokumenten geliefert und müssen diese mittels Copy&Paste übertragen. Hierbei muss darauf geachtet werden, dass die jeweiligen Motive mit den passenden Texten versehen werden, entsprechend umfangreich sind die Hinweise im *Word*-Dokument. Das größte Problem sind die fehlenden Hinweise, wie man Texte korrekt umbricht, da es gerade bei Banner Texten mit nur wenigen Worten pro Zeile gibt. Diese Zusatzinformationen werden aber oft nicht vom Übersetzer hinterlegt.

3.4.3 EA Phenomic: BattleForge

Mehrere tausend Texte und deren Übersetzung für das auf Spielkarten basierende Echtzeitstrategiespiel *BattleForge* von *EA Phenomic* wurde bei diesem Projekt mit Hilfe einer Excel-Tabellen verwaltet. Hierbei wurden in der ersten Spalte Identifier für alle Texte vergeben. In den weiteren Spalten wurden die Übersetzungen eingetragen. Übersetzungsbüros haben die Texte mit Hilfe einer Kopie der Datei übersetzt. Praktikanten haben die übersetzten Texte dann wieder in die Master-Datei eingepflegt. Hierbei kam es regelmäßig zu Fehler, zum einen, wurde Zeilen und damit Identifier gelöscht und zum anderen wurden mehrzeilige Texte manchmal versehentlich in mehrere Zeile geschrieben, dabei wurden neue Zeilen eingefügt, die dann aber keinen Identifier mehr besaßen. Parallel zu den Arbeiten des Übersetzungsbüros kam es aber auch immer wieder zu Änderungen an den Ursprungstexten und Identifiern. Diese Änderungen mussten dann in allen Sprachversionen nachträglich überarbeitet werden. Zusätzlich wurden die Texte durch Markenanwälte begutachtet, auch deren Feedback musste wieder in das Excel-Dokument eingepflegt werden und hatte ggfs. Einfluss auf die Texte im Spiel und die Übersetzung.

3.4.4 MAN Truck & Bus AG: Neufahrzeug-Konfigurator

Im Neufahrzeug-Konfigurator der *MAN Truck & Bus AG* existieren über 20.000 Texte in 18 Sprachen. Diese Texte werden zentral in einem Content-Management-System verwaltet. Müssen neue Texte eingepflegt oder aktualisiert werden erhalten die Landesniederlassungen ein Delta als *Excel*-Dokument, das in der Regel alle 4 Monate jeweils 1.000 bis 1.500 Texte enthält. Nachdem die Texte von der Landesniederlassung übersetzt wurden, werden diese von einer Übersetzungsagentur geprüft und mittels Copy&Paste wieder in das CMS eingepflegt. Probleme entstehen hier, wenn sich Landesniederlassungen nicht an das Dokumenten-Format halten und zusätzliche Texte, z.B. Übersetzungsalternativen, ein-

fügen. Der Versand der Dokumente erfolgt durch einen zentralen Datenverantwortlichen, der sicherstellen muss, dass alle Landesniederlassungen ihre Dokumente bekommen, wieder zurücksenden und sie dem Übersetzungsbüro anschließend zur Verfügung gestellt werden.

3.5 Schlussfolgerung

In diesem Kapitel wurde erläutert, warum Text eine besondere Rolle bei der Erstellung von Informations- und Kommunikationsmedien spielt. Es wurde gezeigt, dass *Word* und *Excel* ausgewählt werden, um die Texten für ein Produkt zu Verwalten den den vielen Projektbeteiligten zugänglich zu machen. Der Grund, warum diese Programme für die Erfassung, Erstellung, Übersetzung und Übertragung von Texten in das Produkt verwendet werden ist der, dass keine dedizierten Lösungen existieren, die explizit die genannten Abläufe in der Textverarbeitung abbildet. Statt dessen wird Software verwendet, die bei allen Beteiligten vorhanden ist und mit denen diese bereits vertraut sind, wodurch sie mit Texten in einem gewohnten Umfeld arbeiten können. Die Verwendung von Dateien ermöglicht den Austausch unter den Beteiligten im Netzwerk oder per E-Mail. In diesem Kapitel wurde jedoch gezeigt, dass diese Wahl mit vielen Nachteilen verbunden ist und im Projektverlauf viele Stellen eröffnet, an denen es zu Problemen kommen kann oder es wegen der schlechten Eignung zu einer langsamen Arbeitsgeschwindigkeit führt. Die Probleme dieses Workflows werden jedoch erst im Verlauf des Projektes sichtbar und betreffen vor allem Agenturen, die als *Dienstleister* in einem Abhängigkeitsverhältnis zu stehen. Auf deren Seite werden die Missstände durch Mehrarbeit und aufwändige, sich wiederholende Arbeitsschritte ausgeglichen, die aufgrund ihrer Natur fehleranfällig sind.

Bevor in Kapitel 5 / S.28 eine Lösung für diese Probleme konzipiert wird, werden im nächsten Kapitel jedoch zuerst Personas vorgestellt, die die typischen Nutzer dieser Lösung repräsentieren.

4 Personas

Im vorigen Kapitel wurden die Probleme geschildert, die in Projekten auftreten, in denen Informations- und Kommunikationsmedien erstellt werden und daraus gefolgert, welches die wichtigsten Aspekte sind, die eine mögliche Lösung berücksichtigen muss. In diesem Kapitel werden Personas vorgestellt, die die typischen Benutzergruppen des Systems repräsentieren und ihre Aufgaben und Erwartungen zusammengefasst.

„Personas describe a site’s target users, giving a clear picture of how they’re likely to use the system, and what they’ll expect from it, among other things. [...] Without personas, there is no common language for talking about what users want.“ [2, S.15 ff.]

Personas sind ein wichtiger Baustein für die Konzeption eines Systems. Sie ermöglichen es Bestandteile des Konzepts schon während des Entwurfs zu verifizieren, in dem deren Auswirkungen mit dem Nutzungsverhalten und den Erwartungen der Personas verglichen werden. Personas bilden nicht nur im Konzept eine wichtige Entscheidungshilfe sondern werden auch während des Entwurfs und der Umsetzung immer wieder zu Rate gezogen, in dem Funktionalitäten auf deren Relevanz und mögliche Probleme für bestimmte Personas hin überprüft werden. [4, S.38 ff.]

Die Personas basieren auf im April 2012 geführten Interviews mit den in Tabelle 2 / S.19 aufgezählten Personen. Die Namen und Fotos der Personas basieren zwar auf den interviewten Personen, dienen aber lediglich dazu, das Merken der Personas zu erleichtern.

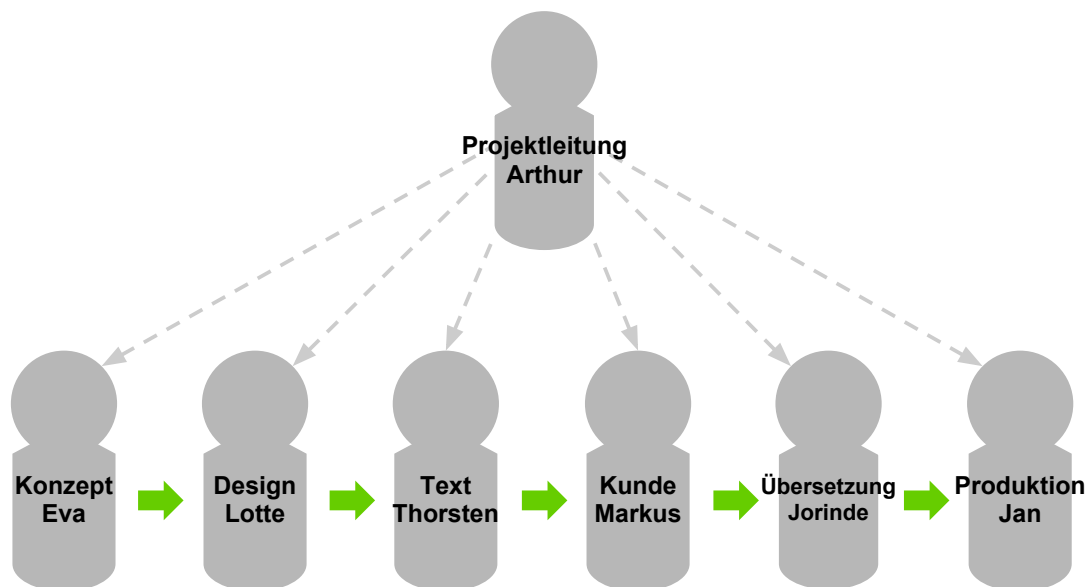


Abbildung 4: Übersicht über die Personas und den idealisierten Workflow

Projektleiter

Arthur Blozyk <i>Dipl. Ingenieur, Dipl. Wirtschaftsingenieur</i>	<i>Sales Information and Communication MAN Truck & Bus AG</i>
Sebastian Nell	<i>Director of USE // Connected Products Scholz & Volkmer GmbH</i>
Tobias Rudolphi <i>Dipl. Informatik</i>	<i>Lead Software Architect Zühlke Engineering GmbH</i>

Konzept, Design

Carsten Fischer <i>M.A. Sozialwissenschaften, Anglistik</i>	<i>UX Designer & Informationsarchitekt triplesense GmbH</i>
Eva Kümml <i>Dipl. Medienwirtin</i>	<i>Senior Konzept / User Experience SinnerSchrader Deutschland GmbH</i>
Sandra-Charlotte Hildebrandt <i>Dipl. Designerin</i>	<i>freie Art-Direktorin</i>

Produktion

Sebastian Beyer <i>Informatik, B.Sc.</i>	<i>Developer Scholz & Volkmer GmbH</i>
Jan Lochner <i>Dipl. Multimedia Producer</i>	<i>Freelancer</i>

Texter

Marc Stenzel	<i>Fachjournalist, freier Projektleiter</i>
Torsten Schölzel <i>Dipl. Soziologe</i>	<i>freier Texter</i>

Übersetzer

Jorinde Gessner <i>German and Politics, B.A. Hons</i>	<i>Information Manager Ogilvy & Mather Deutschland GmbH</i>
--	---

Kunde

Markus Rüb <i>Dipl. Wirtschaftsingenieurwesen</i>	<i>Sales Information and Communication MAN Truck & Bus AG</i>
--	---

Tabelle 2: Interviewte Personen

Die Auswahl der Personas orientiert sich an dem vorherrschenden Workflow innerhalb der Projekte. Wie im vorangegangenen Kapitel gezeigt wurde, gibt es in der Praxis keinen linearen Ablauf, sondern es ergeben sich vielzählige Feedback-Schleifen. Vernachlässigt man diese Feedback- und Korrekturschleifen kann man die beteiligten Personen in eine, in Abbildung 4 / S.18 gezeigte, lineare Reihenfolge bringen:

1. Konzepterin *Eva* entwickelt das Produkt, wobei sie die Rahmenbedingungen wie Aufbau, Umfang, Zielgruppe, Ansprache festlegt.
2. Designerin *Lotte* gestaltet das Produkt, wobei sie bestimmt, wie Texte dargestellt werden (Satz, Länge, Schriftart, Farben, Hervorhebungen)
3. Texter *Torsten* erstellt die Texte für das Produkt in der Ausgangssprache
4. Kunde *Markus* nimmt die Texte ab
5. Übersetzerin *Jorinde* übersetzt die Texte
6. Produzent *Jan* übernimmt die Texte in das Produkt

Eine wichtige Rolle fehlt in dieser Auflistung: Projektleiter *Arthur* koordiniert den Ablauf des Projektes, hat aber keinen Einfluss den Text. Er darf jedoch als wichtiges Bindeglied zwischen allen Beteiligten nicht fehlen.

Bei der Formulierung der Personas wurde bewusst darauf verzichtet, persönliche Daten wie Alter und Bildung zu verwenden, da diese keinen Einfluss auf die Konzeption des Systems haben. Die Personas enthalten dementsprechend nur

- den wichtigsten Nutzen aus der Sicht der Person als Zitat
- eine Beschreibung der Aufgabe der Rolle, die diese Persona repräsentiert
- Angaben zu den Beteiligten, mit denen sich die Person im Verlauf des Projektes *über Texte* austauschen wird
- die verwendeten Werkzeug zur Erfüllung der Aufgabe und die Erfahrung im Umgang mit Anwendungen im Allgemeinen
- die wichtigsten Szenarien, die die Persona mit Hilfe des Systems durchführen wird
- allgemeine Anforderungen an das System
- und Angaben darüber, wie der Zugriff auf das System erfolgt.

Diese Information zusammen mit der Problemanalyse im vorangegangenen Kapitel bilden die Basis für die Konzeption einer Lösung im nächsten Kapitel (5 / S.28).

4.1 Eva, Konzepterin



„Ich möchte, dass alle Beteiligten einen guten Überblick über das Produkt haben.“

Eva konzipiert als Informationsarchitektin das Produkt. Dabei legt sie entsprechend der Zielsetzung fest, wie das Produkt aufgebaut ist um die Erwartungen des Nutzers zu erfüllen und ihn das gewünschte Ergebnis im Sinne des Produktes leicht erreichen zu lassen. Hierzu erstellt sie einen Überblick über das Produkt mit Hilfe von Wireframes und macht dabei Vorgaben über die Platzierung von Texten und deren Funktion.

Organisation, Abstimmung

Eva arbeitet auf Seite der Agentur und stimmt sich mit *Markus* (4.4) sowie *Lotte* (4.2) über das Produkt ab. Sie gibt Feedback zu den Texten von *Torsten* (4.3) und deren Integration in das Produkt durch *Jan* (4.6).

Werkzeuge und Erfahrung

Evas wichtigstes Werkzeug ist *OmniGraffle* von *Omni Group* mit dem sie die Wireframes des Produktes erstellt. Sie ist versiert im Umgang mit vielfältigen Anwendung und steht neuen Werkzeugen offen gegenüber.

Szenarien

Eva definiert die einzelnen logischen Bestandteile des Produktes (z.B. Seiten, Abschnitte) und definiert darin, welche einzelnen Textbausteine verwendet werden.

Eva legt Rahmenbedingungen für den Text fest. Zum einen bestimmt sie die Ansprache, d.h. welche Zielgruppe soll mit den Texten angesprochen werden und welches Ziel verfolgen die Nutzer. Zum anderen macht sie Vorgaben über den Aufbau einzelner Klassen von Texten wie z.B. Überschriften, Schaltflächen, Fließtext bei denen sie z.B. die Textlänge, Spaltenbreite oder Zeilenanzahl festlegt. Diese Rahmenbedingungen kann sie zu den jeweiligen Textbausteinen hinterlegen.

Eva kann sich eine Übersicht ausdrucken, die alle Bestandteile des Produktes enthält. So kann sie leicht den Überblick behalten.

Anforderungen

Eva verwendet das System häufig, deswegen müssen Funktionen zur Definition des Produktes, der Texte und der Rahmenbedingungen einfach zu bedienen sein. Sie will nie die Übersicht verlieren und leicht Elemente verändern können, da sich in der Konzeptionsphase häufig Änderungen ergeben.

Zugang

Eva greift auf das System mit ihrem *MacBook Pro* zu, sie verfügt über einen zusätzlichen großen Bildschirm und eine schnelle Internetverbindung.

4.2 Lotte, Designerin



„Ich möchte, dass meine Vorgaben zur Gestaltung der Texte von allen berücksichtigt werden.“

Lotte gestaltet als Art-Direktorin das Produkt. Sie entwirft dazu das Fein-Layout auf Basis der Wireframes, die von Eva (4.1) erstellt wurden, indem sie für alle Darstellungs-Varianten präzise Entwürfe anfertigt. Hierbei legt sie auch genaue Vorgaben für die Formatierung der Texte im Produkt fest.

Organisation, Abstimmung

Lotte arbeitet auf Seiten der Agentur und stimmt sich mit Eva (4.1) bei der Gestaltung des Produktes ab. Mit Torsten (4.3) spricht sie über ihre Vorgaben zu Texten und überprüft deren Einhaltung in der Umsetzung durch Jan (4.6).

Werkzeuge und Erfahrung

Lotte arbeitet mit den Produkten der *Adobe Creative Suite*. Sie ist nur im Umgang mit wenigen anderen Werkzeugen versiert, steht neuen Anwendungen aber offen gegenüber.

Zugang

Lotte greift auf das System mit ihrem *iMac* zu, sie verfügt über einen großen Bildschirm und eine schnelle Internetverbindung.

Szenarien

Lotte erstellt einen Styleguide und legt dabei fest, welche Text-Klassen im Produkt verwendet werden. Text-Klassen sind z.B. Überschriften, Untertitel und Fließtext. Dazu macht sie für jede Klasse Angaben zu der verwendeten Schriftart, die Einschränkungen, wie z.B. die maximalen Zeilenanzahl oder Menge der Zeichen pro Zeile.

Lotte erstellt Layouts in Form von Screenshots oder Beispielseiten. Hierbei verwendet Sie für die Texte die noch nicht durch Eva (4.1) festgelegt wurden, Blindtexte. Um ein besseres Gefühl für die Inhalte zu bekommen und die Layouts besser mit dem Kunden abstimmen zu können möchte Lotte gerne jetzt schon einige Texte von Torsten (4.3) in den Layouts verwenden.

Während der Umsetzung des Produktes ergeben sich Änderungen am Styleguide. Sie möchte, dass Jan (4.6) alle Stellen im Produkt anpasst, die von dieser Änderungen betroffen sind.

Anforderungen

Lotte möchte das Anlegen von Text-Vorgaben unkompliziert und schnell erledigen können. Änderungen müssen jederzeit und ohne großen Aufwand möglich sein. Das System muss leicht verständlich sein, da sie nie viel Zeit damit verbringt. Komplizierte Abläufe würden Sie abschrecken und sie würde statt dessen eine E-Mail schreiben.

Lotte hat sehr hohe Ansprüche an die Gestaltung des Systems.

4.3 Torsten, Texter



„Ich möchte den Texteditor an meine Bedürfnisse anpassen können und beim Schreiben in meinem »Flow« nicht unterbrochen werden.“

Torsten erstellt auf Basis der vom Kunden gelieferten Materialien oder bereits bestehender Produkte die Texte für das Produkt. Hierbei muss er Vorgaben aus dem Konzept und dem Design berücksichtigen, sowie Wünschen und möglicherweise verbindliche Richtlinien des Kunden beachten. Hierzu erstellt Torsten für alle Textbausteine des Produktes die Texte in der endgültigen Fassung in seiner Muttersprache.

Organisation, Abstimmung

Torsten ist selbständig. Er stimmt sich mit *Eva* (4.1) und *Lotte* (4.2) bezüglich deren Vorgaben ab. Er berücksichtigt inhaltliche Vorgaben, die er von *Markus* (4.4) erhält. Er steht *Jorinde* (4.5) für Rückfragen zur Verfügung.

Werkzeuge und Erfahrung

Torsten arbeitet mit *iWorks Pages* von Apple, da er die vielen Funktionen von *Word* als störend empfindet und diese nicht benötigt. Er ist nur im Umgang mit wenigen Programmen vertraut.

Szenarien

Torsten verfasst die Texten zum Produkt, dazu befüllt der die vom Design vorgegeben Textbaustein. Dabei bekommt kann er die Vorgaben sehen, die für den jeweiligen Text gelten (z.B. maximale Textlänge).

Torsten benötigt Kontext-Information zum aktuellen Text und kann im System die vom Kunden zur Verfügung gestellten Materialien aufrufen. Auch die Hinweise zur Zielgruppe und Funktion des Textes aus dem Konzept kann er sich ansehen, ohne die aktuelle Ansicht des Systems verlassen zu müssen.

Der Kunde wünscht Änderungen an den Texten. Diese sind im System bei den jeweiligen Textbausteinen hinterlegt, so das Torsten die Texte schnell anpassen kann, ohne sie erst aufwändig suchen zu müssen.

Torsten arbeitet in einem Team mit anderen Textern und führt die Qualitätskontrolle an den Texten seiner Kollegen durch.

Anforderungen

Das Editor zum Erstellen von Texten muss im System an Torsten Bedürfnisse anpassbar sein. Er möchte genaue Kontrolle darüber haben, *wann* andere Projektmitarbeiter seine Texte sehen können. Feedback zu Texten soll zu den jeweiligen Textbausteinen zugeordnet werden können.

Zugang

Torsten arbeitet in seinem eigenen Büro oder von unterwegs, da er an mehreren Projekten gleichzeitig arbeitet. Er greift auf das System mit seinem *MacBook Pro* zu und je nach Standort kann seine Internetverbindung auch nur mittel schnell sein.

4.4 Markus, Kunde



„Ich möchte, dass die Vorgaben aus unserer Marketing-, Fach- und Rechtsabteilung genau berücksichtigt werden. Änderungswünsche und Korrekturen sollen exakt befolgt werden.“

Markus ist der Auftraggeber des Produktes. Er ist der erste Ansprechpartner für die Agentur, steht aber auch stellvertretend für weitere Unternehmensmitarbeiter aus verschiedenen Abteilungen seines Unternehmens. Er liefert Materialien und Vorgaben die als Basis für die zu erstellenden Texte dienen. Er führt auch die finale Abnahme der Texte und des Produktes durch.

Organisation, Abstimmung

Markus stimmt sich über den Zweck des Produktes mit *Eva* (4.1) ab. *Torsten* (4.3) gibt er Feedback und Änderungswünschen zu dessen Texten. Auch die Übersetzungen von *Jorinde* (4.5) kontrolliert und verbessert er. Wurden die Texte in das Produkt übernommen, gibt er Änderungswünsche an den Inhalten und der Darstellung an *Jan* (4.6) weiter.

Werkzeuge und Erfahrung

Markus arbeitet vor allem mit *Word*, *Excel* und *PowerPoint*, ansonsten hat er keine weiteren Erfahrungen im Umgang mit Werkzeugen.

Szenarien

Markus nimmt die fertigen Texte ab. Hierzu kann er einzelnen Textbausteine freigeben oder Änderungen anfordern, die er mit Hilfe von Kommentaren beschreibt.

Um den Projektablauf nicht zu verzögern beantwortet Markus Rückfragen zu Texten von unterwegs.

Materialien können in das System eingestellt und einzelnen Textbausteinen oder ganzen Abschnitten zugeordnet werden.

Markus kann sich alle Texte als *Word*-Dokument exportieren um Korrekturen darin vorzunehmen. Anschließend importiert er das Dokument. Dabei werden die Änderungen automatisch im System übernommen.

Anforderungen

Das System „muss einfach funktionieren“, da er nicht gezwungen werden will, ein neues System zu erlernen. Er erwartet, dass die wichtigen Funktionen auch mobil verfügbar sind. Die Daten im System und der Zugriff darauf müssen sicher sein. Er will festlegen, wer Zugriff erhält, mit der Möglichkeit, dies auch für einzelne Bereiche des Projektes definieren zu können.

Zugang

Markus arbeitet von seinem Büro-PC aus und greift auf das System mit einer schnellen Internetverbindung zu. Er ist viel unterwegs und verwendet dann sein *iPhone* oder seinen *iPad*.

4.5 Jorinde, Übersetzerin



„Zum Übersetzen brauche ich eine praktische Darstellung der Original-Texte. Der Zugriff auf Kontext-Informationen muss leicht möglich sein.“

Jorinde übersetzt die Texte des Projektes in ihre Muttersprache. Sie berücksichtigt dabei bestehende Materialien des Kunden, sowie die Vorgaben aus Konzept und Design.

Organisation, Abstimmung

Jorinde arbeitet in einem Übersetzungsbüro als Teil eines Teams von Übersetzern, die das gesamte Projekt übersetzen. Sie stimmt sich mit *Torsten* (4.3) bei inhaltlichen Fragen zu Texten ab. Für die Übersetzung spezieller Begriffe stimmt sie sich mit *Markus* (4.4) ab.

Werkzeuge und Erfahrung

Jorinde verwendet zum Erstellen der Übersetzung *Word*, da ihr dort Komfortfunktionen wie Rechtschreibkorrektur und Synonyme zur Verfügung stehen. Jorinde hat wenig Erfahrung mit anderen Werkzeugen und braucht im allgemeinen länger, um sich an neue Systeme zu gewöhnen.

Zugang

Jorinde greift auf das System nur von ihrem Firmen-PC aus zu und verfügt über eine schnelle Internetverbindung.

Szenarien

Jorinde übersetzt die Texte des Projekts, dazu werden die Originaltexte direkt in der Übersichtsansicht dargestellt. Die Zusatzinformationen zu den Textbausteinen stehen ihr, wie bei *Torsten* (4.3), in der Ansicht zur Verfügung.

Jorinde ist sich bei der Übersetzung eines bestimmten Begriffes unsicher. Sie verwendet das zur Verfügung gestellte Ausgangsmaterial des Kunden um die Übersetzung nachzuschlagen, die in Publikationen des Kunden üblicherweise verwendet wird. Für zukünftige Verwendung hinterlegt sie dies im Glossar des Projektes.

Das Projekt enthält an mehreren Stellen die gleichen Formulierungen. Jorinde erhält an diesen Stellen die Übersetzung vorgeschlagen, die sie bereits angelegt hat. Sie kann diese direkt übernehmen oder eine Variante anlegen.

Jorinde sucht für einen Begriff ein Synonym. Sie kann sich direkt in der Ansicht zu dem Begriff Synonyme aus einem globalen Wörterbuch anzeigen lassen. Zusätzlich kann sie sich für das Wort in der Originalsprache die Übersetzungen anzeigen lassen, die bereits im Projekt verwendet wurden.

Jorinde führt die Qualitätskontrolle an übersetzten Texten ihrer Kollegen durch.

Anforderungen

Jorinde benötigt zum Übersetzen im System die Hilfsmittel, die ihr auch in *Word* zur Verfügung stehen: Rechtschreibkorrektur, Synonyme sowie die nahtlose Integration von Wörterbüchern in mehreren Sprachen.

4.6 Jan, Produzent



„Ich muss exakt wissen, welche Texte an welche Stelle im Produkt gehören. Bei Änderungen am Text möchte ich nicht jedes mal die Texte per Copy&Paste übertragen müssen.“

Jan ist für die Erstellung des Produktes verantwortlich. Er hat aber auch während der Entwurfsphase Einfluss auf die Rahmenbedingungen für Texte, vor allem wenn es um technische Parameter geht (z.B. maximale Zeilenlänge).

Organisation, Abstimmung

Jan arbeitet auf Seiten der Agentur und stimmt sich mit Eva (4.1) über den Aufbau und mit Lotte (4.2) über die Gestaltung des Produktes ab. Von Markus (4.4) bekommt er letzte Änderungen am Text mitgeteilt, die erst bei der Darstellung im fertigen Produkt auffallen.

Anforderungen

Für Jan ist es sehr wichtig, dass er die Texte am besten automatisiert in seine Werkzeuge übernehmen kann, so dass Texte, die bereits in das Produkt integriert wurden bei Änderungen automatisch aktualisiert werden können. Für Software-Produkte erwartet er, dass der Zugriff auf die Texte mit einer API möglich ist.

Szenarien

Jan hat eine Broschüre in *Adobe InDesign* erstellt. Er verknüpft die Texten aus dem System mit den Texten im Dokument. Nachdem sich bereits verwendete Texte geändert haben, öffnet Jan das Dokument erneut und kann mit Hilfe eines Dialoges die geänderten Stellen anspringen. Er muss diese nur noch auf gestalterische Probleme hin kontrollieren.

Jan hat eine *Android*-App entwickelt und verwendet die Identifier der vom Konzept vorgegebenen Texte. Beim kompilieren der App lädt das build-Script die aktuellen Texte für die App über die Schnittstelle des Systems und erzeugt automatisch die Sprachdateien.

Jan entdeckt ein Problem mit der Textlänge einer Überschrift. Er meldet dieses Problem im System. Lottes (4.2) Änderungen betreffen allen Überschriften. Jan kann sich im System alle betroffenen Stellen im Produkt anzeigen lassen.

Werkzeuge und Erfahrung

Jan arbeitet je nach Produkt mit DTP-Produkten oder IDEs (für diese Persona ist es unerheblich, ob ein Medium oder Software entsteht), er ist versiert im Umgang mit vielen Werkzeugen und kann sich sehr schnell in neuen Anwendungen eingewöhnen.

Zugang

Jan greift von seinem Laptop aus auf das System zu. Er verfügt über eine schnelle Internetverbindung.

4.7 Arthur, Projektleiter



„Ich möchte steuern, wer welche Aufgabe im Projekt übernimmt und über Problemen informiert sein. Ich möchte jederzeit einsehen können, welcher Anteil der Texte bereits fertig ist.“

Arthur koordiniert als Projektleiter mit allen Beteiligten den Ablauf des Projektes, hat jedoch keinen Einfluss auf die eigentlichen Texte.

Organisation, Abstimmung

Arthur arbeitet auf Seiten der Agentur und stimmt sich über organisatorische Fragen mit allen Beteiligten ab.

Werkzeuge und Erfahrung

Arthur arbeitet vor allem mit *Word*, *Excel* und *PowerPoint*, findet sich aber leicht in anderen Anwendungen zurecht.

Anforderungen

Für Arthur muss das System vor allem immer verfügbar sein, Unterbrechungen im Projektverlauf durch einen Systemausfall sind nicht akzeptabel. Auch ein Datenverlust muss ausgeschlossen sein. Es muss möglich sein alle Daten des Projektes zu exportieren.

Zugang

Arthur greift von seinem Laptop auf das System zu und verfügt über eine schnelle Internetverbindung.

Szenarien

Arthur legt ein neues Projekt an und fügt Mitarbeiter mit Hilfe ihrer E-Mail-Adressen hinzu. Er kann Mitarbeitern Rollen zuweisen, damit klar ist, welche Aufgabe sie haben.

Die Rechtsabteilung des Kunden muss die AGB, das Impressum und die Datenschutzbedingungen einer Website abnehmen. Arthur konfiguriert den Workflow im System so, dass diese Texte von allen Mitarbeitern dieser Abteilung freigegeben werden müssen.

Arthur beauftragt ein Lektorat mit der Kontrolle aller Texte. Er konfiguriert den Workflow so, dass Texte erst durch den Kunden einsehbar sind, wenn die Mitarbeiter des Lektorats alle geprüft haben.

Anhand der Aufzeichnungen seit Projektbeginn kann das System für Arthur einen voraussichtlichen Abschlusstermin berechnen. Es ist ersichtlich, dass der Texter für die Erstellung seiner Texte zu lange braucht. Arthur zieht einen Texter aus einem anderen Projekt ab und weist ihn diesem Projekt zu. Das System berechnet mit der zusätzlichen mittleren Arbeitsleistung des neuen Texters ein neues Enddatum, das jetzt Arthurs Vorstellung entspricht.

5 Konzeption einer an den spezifischen Workflow angepassten Anwendung

Im Kapitel 3 wurde analysiert, welche Probleme in Zusammenhang mit der Produktion von Informations- und Kommunikationsmedien bezüglich den verwendeten Texten entstehen, wenn Textverarbeitungs- und Tabellenkalkulationsprogramme wie *Microsoft Word* und *Excel* verwendet werden. In diesem Kapitel wird eine Anwendung konzipiert, die den Anforderungen der im vorangegangenen Kapitel vorgestellten Personas entspricht und die beschriebenen Probleme beseitigt. Hierzu werden zunächst in Abschnitt 5.1 und 5.2 die Anforderungen an die Lösung beschrieben, bevor beginnend ab Abschnitt 5.3 / S.34 die Anwendung konzipiert wird, die diese Anforderungen erfüllt.

In Kapitel 6 / S.40 wird auf Basis dieses Konzepts eine Anwendung entworfen und in Form eines Prototypen umgesetzt.

5.1 Abzubildender Workflow

Hauptaufgabe der Anwendung ist es, den Workflow von der Definition bis zur Übernahmen der fertigen Texte in das Produkt zu übernehmen und dabei nicht nur Funktionen zur *Definieren* und *Speichern* und *Exportieren* zu bieten, sondern auch die *Kommunikation über die Texte* zu integrieren. Für die Umsetzung des Workflows in einer Anwendung müssen zunächst alle Ausprägungen der speziellen Abläufe beschrieben werden. In Abschnitt 3.1 wurde bereits beschrieben, wie umfangreich die Anzahl der Personen ist, die Einfluss auf die Texte eines Produktes haben. Die Rollenverteilung ist dabei von Projekt zu Projekt unterschiedlich, mit den in Kapitel 4 vorgestellten Personas würde eine Übersicht über die typische Rollenverteilung geschaffen. Betrachtet man die von Projektmitarbeitern durchgeführten Operationen in Zusammenhang mit Text lassen sich diese in sechs eigenständige Operationen unterteilen:

1. Durch **Definieren eines Textbausteines** werden dessen *Attribute* bestimmt. Dadurch wird festgelegt, wie der benötigte Text beschaffen sein muss. Die Aussage „*Wir brauchen an dieser Stelle eine Überschrift*“ ist ein Beispiel für diese Operation. Sie legt fest, wie der Textbaustein gestaltet werden muss, um die ihm zugeordnete Aufgabe zu erfüllen. Neben der Angabe zur Platzierung auf dem Medium durch »an dieser Stelle« wird implizit durch »eine Überschrift« eine Angabe zur inhaltlichen und visuellen Gestaltung getroffen; Überschriften sollen kurz und knapp sein und ihre visuelle Gestaltung wird durch den Styleguide des Projektes festgelegt.
2. Das **Schreiben eines Textes** erzeugt den Inhalt eines Textbausteins in einer Sprache. Bei diesem Vorgang wird der Text entsprechend der Vorgabe aus der Beschreibung als Original erstellt oder aus Quellen außerhalb des Projektes kopiert und eingefügt.

Persona	Inhalt	Attribute	Status
Agentur			
Eva, Konzept	◐	●	○
Lotte, Art-Direktion	○	◐	○
Jan, Produktion	○	◐	○
Arthur, Projektleitung	○	○	◐
Extern			
Torsten, Text	◐	○	○
Jorinde, Übersetzung	◐	○	○
Kunde			
Markus	◐	◐	●

Tabelle 3: Stärke des Einfluss, die Mitarbeiter in einem Projekt haben

3. In der **Korrektur** wird der Text inhaltlich und grammatikalisch überprüft und entsprechend angepasst. Der Korrektor muss dabei für eine grammatikalische Überprüfung des Textes kein Fachwissen bezogen auf das Projekt haben. Ist dieses Fachwissen vorhanden, kann eine inhaltliche Korrektur vorgenommen werden.
4. In der **Qualitätskontrolle** wird der Text dahingehend überprüft, ob er den Anforderungen gemäß der Beschreibung und inhaltlichen Vorgaben, auch hinsichtlich des gesamten Projektes entspricht.
5. Durch die **Freigabe** wird der Text abgenommen und kann nun in das Endprodukt übernommen werden. Die Freigabe unterscheidet sich von der Qualitätskontrolle durch ihren autoritativen Charakter. Qualitätskontrolle können prinzipiell von allen Mitarbeitern durchgeführt werden. Freigaben werden nur von Mitarbeitern mit Management-Berechtigungen erteilt.
6. Durch die **Veröffentlichung** wird der Text in das Endprodukt eingebracht.

Verallgemeinert man diesen Ablauf, erkennt man, dass sich der Einfluss in drei grundlegenden Eigenschaften der Texte unterteilen lässt:

1. den **Inhalt** des Textes,
2. die **Attribute** wie z.B. »maximale Textlänge« oder »Position im Medium«
3. und den **Status** wie z.B. »neu« und »freigegeben«.

Das Gewicht des Einfluss der Mitarbeiter ist je nach Rolle unterschiedlich, Tabelle 3 / S.29 zeigt dies in einer Übersicht.

Nachfolgend sind die Text-Eigenschaften näher beschrieben.

5.1.1 Inhalt

Personen die Einfluss auf den *Inhalt* haben, sind vor allem diejenigen die die Texte für das Produkt liefern. Neben den Mitarbeitern auf Kundenseite, Ausgangsmaterialien und Fachinformationen zur Verfügung stellen sind die Texter und Übersetzer, die diese Informationen aufbereiten. Texte müssen aber auch die spezifischen Gegebenheiten des Mediums angepasst werden, hierzu liefern Experten Rahmenbedingungen aber auch inhaltliche Anpassungen. Ein Beispiel hierfür ist die Suchmaschinenoptimierung (SEO) von Texten. Hierbei werden Texte auf das Vorhandensein von bestimmten Formulierungen und Stichwörter optimiert aber auch Vorgaben über die Länge und Aufbau von Texten gemacht.

Mit Inhalt ist der eigentliche Text gemeint, der auch im Produkt erscheint. Für Inhalte gibt es immer eine Original-Version für die im späteren Projektverlauf Übersetzungen in eine oder mehrere Sprachen angelegt werden können. Die Übersetzung basiert dabei auf der Original-Version, oder je nach Übersetzer auch auf einer anderen Übersetzung.

Vorgaben für Inhalte werden in Form von Richtlinien formuliert. Diese Richtlinien dienen dem Texter als Orientierungshilfe, wie er die Texte zu formulieren hat. Richtlinien werden von verschiedenen Mitarbeitern formuliert:

- Vom Konzept werden grundlegende Vorgaben geschaffen, wie z.B. Annahmen über die Zielgruppe und den Zweck des Produktes.
- Der Kunde hat Vorstellungen oder sogar Vorgaben, wie der Sprach-Stil der Texte sein soll, aus seinen Fachabteilungen und von Beratern oder Anwälten werden weitere Vorgaben über erwünschte oder zu vermeidende Aussagen erstellt.

Es existieren aber auch implizite Vorgaben, die sich aus der Art des Mediums ergeben: Lange Texte sind für Fernsehspots ungeeignet, Informationsbroschüren haben Raum für ausführliche Erläuterungen.

Nicht selten werden weitere externe Experten zu Projekten hinzugezogen, die Texte auf bestimmte Aspekte hin anpassen. Besonders bei Internetseiten werden SEO-Konzepte erstellt, auf deren Basis die Texte von einzelnen Seiten und Abschnitten so angepasst werden, dass diese bestimmte Schlüsselwörter und Formulierungen enthalten, um von den Algorithmen der Suchmaschinenbetreiber bessere Positionierungen in Suchergebnissen zu erreichen.

5.1.2 Attribute

Attribute legen die Rahmenbedingungen von Text fest, diese werden vor allem in der Gestaltung des Produktes durch Designer, als auch in der Umsetzung durch produktbedingte Einschränkungen, z.B. Platzverhältnisse oder systembedingte Beschränkungen, bestimmt. Attribute sind in irgendeiner Form quantifizierbare Angaben zu Textbausteinen. Sie dienen zum einen dazu, den Textbaustein zu identifizieren und seine Rolle im Produkt festzulegen

und zum anderen werden damit Vorgaben zur Beschaffenheit des Textes festgelegt. Die Attribute lassen sich in vier Bereiche unterteilen:

Identifizier Dies sind eindeutige Bezeichner für Textbausteine, diese dienen dazu, Referenzen zwischen den Textbausteinen im System und im Produkt herzustellen, z.B. um automatische Aktualisierungen zu ermöglichen. Identifizier werden auch benötigt, damit Kommentare, Zusatzinformationen usw. diesen eindeutig zugeordnet werden können.

Klasse generell lassen sich die Texte in jedem Produkt in wenige, deutlich unterscheidbare Klassen unterteilen. Aus Gründen der Usability versucht man bei der Gestaltung von Medien ein einheitliches Gestaltungsbild zu erreichen, dazu gehört auch, dass Texte, die die gleiche Funktion haben, auch gleich gestaltet werden. In den meisten Fällen gibt es eine Unterscheidung zwischen Fließtext und Überschriften, die es auf mehreren Hierarchieebenen gibt. Bei interaktiven Produkten finden sich dann häufig Texte für Navigations-Elemente wie Buttons und Links oder für die Verwendung in Formularen. Bei der Klassifizierung von Texten werden die Schriftart, Schriftgröße, Schreibweise (z.B. nur GROSSBUCHSTABEN) und weitere Angaben zu Formatierung (z.B. Abstände zum nächsten Textbaustein) bestimmt, die für alle Texte dieser Klasse verwendet werden. Zusätzlich werden üblicherweise auch noch weitere Regeln für die Verwendung der Klassen festgelegt, z.B. dass auf eine Headline immer vor einem Fließtext stehen muss. Teil der Klassifizierung können auch Angaben zur Zeichenlänge enthalten, da dies aber nicht immer zwingend der Fall ist und es auch Textlängenbeschränkungen ohne Bezug zur Klasse eines Textes geben kann (z.B. bei Formularen) werden diese separat betrachtet.

Textlänge In vielen Fällen ist es aus gestalterischer, inhaltlicher oder technischer Sicht gewünscht, dass die Textlänge eines Textbausteines in einem gewissen Rahmen liegt. Üblich sind Vorgaben zur minimalen, maximalen und gewünschten Anzahl von Zeichen, Wörtern, Sätzen, Zeilen und Absätzen

Position Zu jedem Textbaustein wird festgelegt, wo dieser im Produkt erscheint. Diese Information ist besonders für die Produktion wichtig, aber auch schon vorher wird diese Information immer wieder benötigt, z.B. um vor Fertigstellung des Produktes eine Vorschau oder einen Dummy einzelner Bestandteile des Produktes anzufertigen. Positionsangaben bestehen meistens aus zwei Komponenten. Zum einen wird eine hierarchische Position definiert, z.B. auf „im zweiten Kapitel“, oder „unter der Überschrift auf der Seite »Über uns.«“ Zum anderen kann es exakte Positionsangaben und Größenangaben geben. Diese haben zwar keinen direkten Einfluss auf den Text, können aber für Produkte mit festen Gestaltungsrahmen und den Informationen zur Textklasse wichtige Hinweise darüber liefern, wie lang der Text maximal sein darf, ohne das Layout »zu sprengen«.

	Eva	Lotte	Torsten	Jorinde	Jan	Arthur	Markus
Einfluss auf	Konz.	Des.	Texter	Übersetz.	Prod.	Projektl.	Kunde
Konzept	●	◐	○	○	◐	◐	●
Design	◐	●	○	○	◐	◐	●
Text	◐	◐	●	◐	◐	◐	◐
Übersetzung	○	○	◐	●	◐	◐	◐
Produktion	◐	◐	○	○	●	◐	●

Tabelle 4: Einfluss mit Gewichtung auf den Status eines Textbausteines

5.1.3 Status

Den Status von Texten, also ob ein Text dem nächsten Mitarbeiter im Workflow zugewiesen werden soll kann von bestimmten Mitarbeitern abhängen. Es ist üblich, dass Texte erst dann dem Kunden zur Abnahme vorgelegt werden, wenn sie als Gesamtes vorliegen. Auch externe Dienstleister bekommen aus Kostengründen meistens alle Text im Paket, damit eine zügige Abarbeitung des Auftrages gewährleistet wird. Praktisch ist der Status eines Textes dem aktuellen Bearbeiter gleich zu setzen. Eine Änderung des Status bedeutet, dass der aktuelle Bearbeiter seine Aufgabe abgeschlossen hat und der nächste Bearbeiter mit seiner Aufgabe weitermachen kann. Es kann auch eine Statusänderung in umgekehrter Richtung geben, wenn der die Vorbedingungen für den nächsten Mitarbeiter nicht erfüllt sind, oder bei einer Kontrolle ein Problem festgestellt wird. Je nach Mitarbeiter unterscheidet sich, zu welchen Zeitpunkt und zu welchen Status er Feedback gibt. Abbildung 5 / S.33 gibt einen Überblick über die Entscheidungsprozesse im Verlauf eines Projektes und wer auf diese Einfluss nimmt. Tabelle 4 / S.32 listet den Einfluss tabellarische und enthält zusätzlich Informationen, wie stark der jeweils ausgeübte Einfluss in der Regel ist.

Ereignisse Da sich die Texte des Produktes aus vielen einzelnen Bausteinen zusammensetzen, werden auch die Eigenschaften jeweils einzelnen gesetzt. Aus Sicht des Projektverlaufes ergeben sich aus der Gesamtheit alle Zustände bestimmte Ereignisse, die sich auf die Änderung der Stati der Texte zurückführen lassen. Immer dann, wenn alle Texte eines Produktes einen gewissen Status erreichen, werden bestimmte Ereignisse ausgelöst. Ein Beispiel: erst wenn alle Texte durch den Kunden abgenommen wurden, werden diese an die Übersetzung übergeben. Dies stellt sicher, dass die Übersetzung nur die Texte übersetzt, die auch benötigt werden und somit keine unnötigen Kosten verursacht.

5.2 Anforderungen an die Anwendung

Neben den im vorangegangenen Abschnitt beschriebenen Anforderungen an den Workflow, muss die Anwendung weitere Eigenschaften erfüllen. Die wichtigsten werden nachfolgend kurz beschrieben.

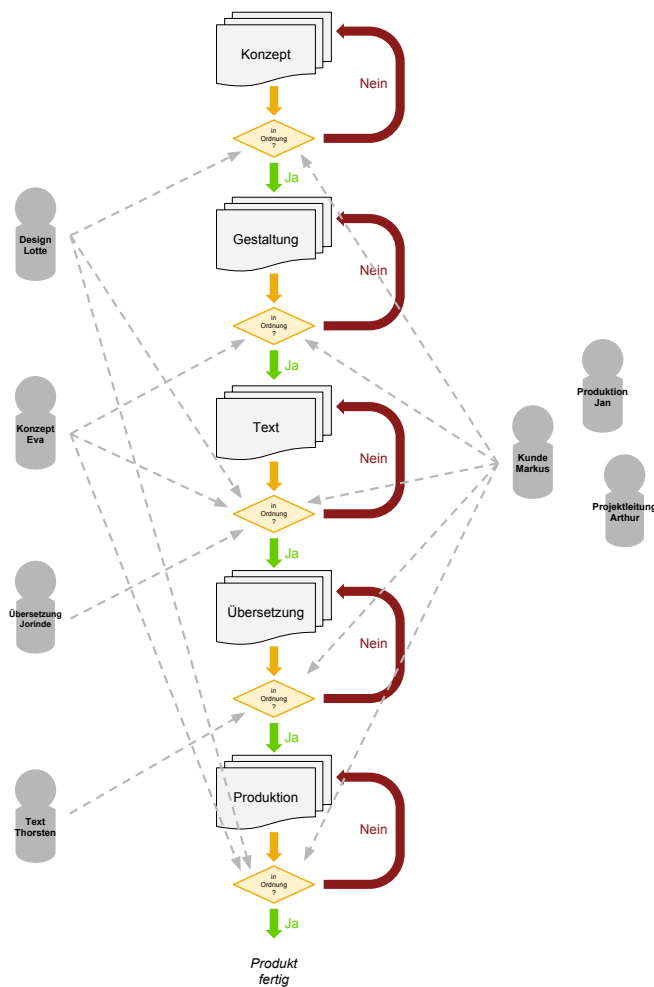


Abbildung 5: Einfluss auf den Status eines Textbausteines

Gleichzeitiges Bearbeiten Es ist essentiell, die Beschränkungen eines seriellen Bearbeitungskonzeptes, wie es durch *Word* und *Excel* vorgegeben wird, aufzuheben und das gleichzeitige Bearbeiten der Inhalte des Projektes zu ermöglichen.

Benutzerverwaltung und Nachvollziehbarkeit Die Abbildung eines Workflows setzt voraus, dass die Beteiligten am Projekt eindeutig identifiziert werden können und dass es möglich ist, die Aufgaben explizit zu verteilen. Das ist auch Voraussetzung für eine weitere Anforderungen nach der Nachvollziehbarkeit aller Änderungen.

Zugang von überall Da sich die Projektbeteiligten nicht alle an einem Ort befinden, muss der Zugang zur Anwendung über das Internet möglich sein.

Zugriff von verschiedenen Plattformen Es gibt unterschiedliche Anforderungen an den Zugang zur Plattform, dieser muss möglichst von vielen Endgeräten und dabei sowohl von stationären Systemen als auch von unterwegs aus möglich sein.

Integration in die Werkzeuge Für viele Arbeiten in Zusammenhang mit der Erstellung von Medien werden spezialisiert Werkzeuge verwendet, die durch die hier vorgestellte Lösung niemals ersetzt werden können. Es muss also eine Integration in diese Werkzeuge mit Hilfe von Plug-Ins oder ähnlichem existieren, mit denen es möglich wird, die Texte aus der Anwendung in die Produkte zu übernehmen. Zur Anbindung dieser Plug-Ins wird eine Schnittstelle (API) benötigt.

Umfangreiche Export-Funktionen Um die Texte in das Produkt zu integrieren, bedarf es umfangreicher Exportfunktionen in strukturierte Formate wie z.B. XML. Zur Kontrolle oder um eine Übersicht über das Produkt zu bekommen ist es erforderlich den Export in Dokumentenformate wie *PDF* und *Word* zu ermöglichen.

5.3 Art der Anwendung

Die im vorangehenden Abschnitt beschriebenen Anforderungen an die Anwendung, besonders im Hinblick auf den universellen, jederzeitigen Zugriff von überall sind die entscheidenden Anforderungen, die für die Konzeption des Systems als *browserbasierte Web-Anwendung mit vollständiger Schnittstellen-Abdeckung* sprechen.

Diese Klasse von Anwendung verwendet einen Webbrowser als Laufzeitumgebung. Dabei stellt der Browser das GUI der Anwendung mit Hilfe von HTML, CSS und JavaScript dar, die Businesslogik und die Datenhaltung wird auf einem Server ausgeführt, mit der die GUI mithilfe einer Schnittstelle kommuniziert. Abbildung 6 / S.35 zeigt den Aufbau des Systems in stark vereinfachter Darstellung. War es in den letzten Jahren noch üblich, dass Fragmente des GUIs mit serverseitigen Template-Sprachen erzeugt wurden (vgl. [6, S.48]) hat die zunehmende Verbreitung von mobilen Clients ein Umdenken zur Folge. Zum einen stellen Desktop-Clients, mobile Browser-Clients und native Apps zwar die gleichen Daten eines Systems dar, verwenden dafür aber nicht zwangsläufig die gleiche GUI-Technologie. Zum anderen werden Clients immer leistungstärker, selbst Einsteiger-SmartPhones haben inzwischen CPUs mit mindestens dreistelligem Megahertz-Wert. Diese Entwicklung führt gerade bei Web-Anwendungen, auch Rich Internet Applications (RIAs) genannt, zu der Idee, Architekturen zu entwickeln, bei denen serverseitig keine GUI-Komponenten mehr erzeugt werden (vgl. [10] und [3]). Clients kommunizieren über Schnittstellen mit dem Server und tauschen nur noch reine Daten aus. Dies hat mehrere Vorteile. Zum einen muss serverseitig kein Modell der clientseitigen Darstellung verwaltet werden, zum anderen verkleinert sich die Menge der transferierten Daten zwischen Client und Server erheblich. Dies hat besonders bei Benutzern mit langsamen oder schlechten Datenverbindungen im Mo-

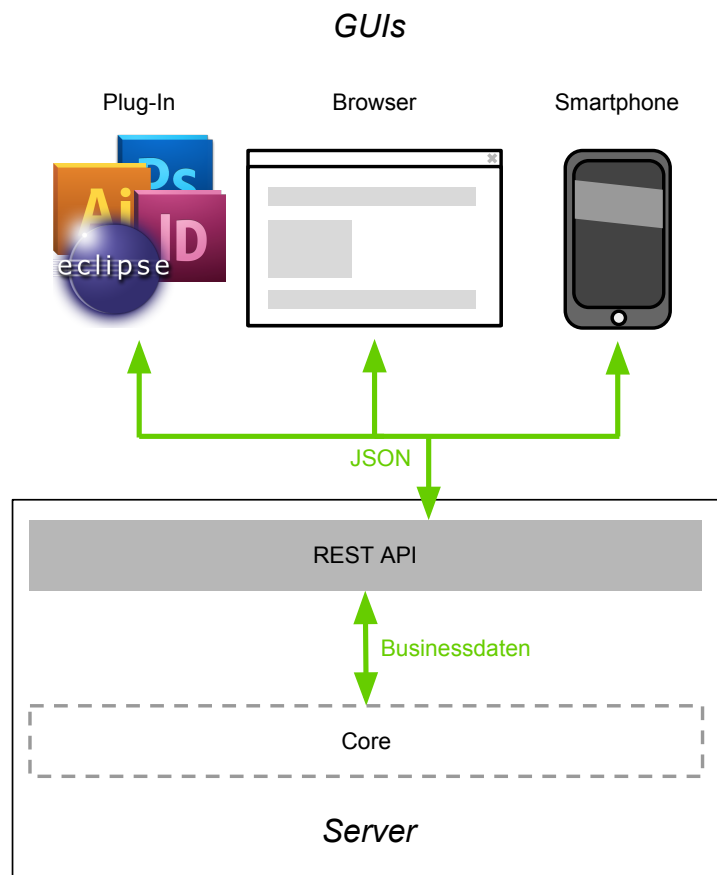


Abbildung 6: Aufbau des System in stark vereinfachter Darstellung

bilfunk-Netz große Vorteile. Für Web-Anwendungen bedeutet dass diese das zur Darstellung benötigte HTML mit Hilfe von JavaScript selber direkt im Client erzeugen. Beim ersten Besuch einer Internetseite müssen lediglich einmal die JavaScript-Dateien und benötigte statische Ressourcen wie CSS-Dateien, Bilder und ein statischer HTML-Grundaufbau geladen werden. Anschließend werden nur noch die für die jeweilige Aktion benötigten Daten mit Hilfe von JavaScript zwischen der Anwendung und dem Server ausgetauscht. Mobile Endgeräte, die über eigene GUI-Toolkits verfügen, oder Software von Drittanbietern können dann die selben Schnittstellen verwenden, ohne dass serverseitige Anpassungen vorgenommen werden müssen.

Web-Anwendung haben den Vorteil, dass sie ohne Installation auf dem Rechner des Benutzers lauffähig sind. Sie können als unmittelbar verwendet werden. Kompatibilitätsprobleme mit alten Browser-Versionen (z.B. dem *Internet Explorer 6*) können inzwischen mit Hilfe des *ChromeFrame*⁶ komfortable umgangen werden. Der Umfang an frei verfügbaren

⁶<https://developers.google.com/chrome/chrome-frame/>

Bibliotheken zur Erstellung attraktiver und angenehm benutzbarer Anwendungen auf Basis von HTML ist riesig. Web-Anwendungen können mit wenig Aufwand auch auf mobilen Endgeräten eingesetzt werden, da Technologien zur plattformabhängigen Anpassung der Darstellung (z.B. CSS-Mediaqueries) existierten. Insgesamt sind Webbrowser der aktuellen Generation mächtige Werkzeuge zur Erstellung von CRUD-Anwendungen. [12] Die allgemeinen Vorteile einer browserbasierten Software, auch als *Software as a Service* (SaaS) oder *Application Service Provider* (ASP) Modell bekannt, liegen auf der Hand und werden an dieser Stelle nicht detailliert ausgeführt. Um nur einen zu nennen: die Möglichkeit, die Software jederzeit und für alle Mitarbeiter gleichzeitig ohne deren Zutun aktualisieren zu können, eliminiert vielfältige Probleme, die sonst in Umgebungen entstehen, in denen unterschiedliche Programmversionen existieren.

5.4 Schnittstellen

Die Verwendung einer einheitlichen Schnittstelle durch alle Clients ermöglicht ein konsistentes Verhalten der Anwendungen über alle Zugangswege hinweg und ist besonders in Fall dieser Anwendung von Bedeutung, da die Benutzer des Systems wünschen, dass sich die Texte direkt innerhalb ihrer bevorzugten Werkzeuge abrufen und einbinden lassen. Dies ist nur mit Hilfe von Plugin-Ins möglich, die in der jeweiligen Umgebung der Software entwickelt werden müssen. Aus diesem Grund ist es unvermeidlich, dass für alle Funktionen des Systems eine öffentliche Schnittstelle existiert.

Als Protokoll zur Kommunikation zwischen Clients und Server hat sich REST bewährt. Die Struktur des Protokolls ist direkt mit dem HTTP-Protokoll verbunden, so ist die Verarbeitung von REST-Anfragen serverseitig leicht mit Web-Frameworks zu implementieren, da diese von sich aus bereits für diese Art von Anfragen ausgelegt sind. Clientseitig wird lediglich ein HTTP-Client benötigt sowie Module zum Parsen von JSON- oder XML-Datenstrukturen – Voraussetzungen, die von Browsern und SmartPhones erfüllt werden. JSON hat im Vergleich zu SOAP den Vorteil, dass es nicht versucht die Architektur der zugrundeliegenden Software nach außen abzubilden, so muss sich der Client nicht an bestimmte Reihenfolgen im Aufruf von Methoden halten. In der REST-Welt sind alle Operationen atomar und können ohne Vorbedingung gestellt werden. In der Praxis ist dies nicht immer umsetzbar, REST fordert serverseitig Zustandslosigkeit, die aber bei Systemen in denen Daten gespeichert und verändert werden nicht realisierbar ist. Aufgrund seines flexibleren Aufbaus, der Möglichkeit ausgewählte Anfragen leicht mit HTTP-Caches zu beschleunigen und der freien Wahl der Nachrichtenformats ist REST aus Sicht des Autors die bessere Wahl zur Implementierung der Schnittstellenkommunikation.

5.5 Zugang

Diese Konzeption macht es möglich, jedem Mitarbeiter den passenden Zugang zu ermöglichen, im Einzelnen sind das:

Browserbasierter Zugang vom Desktop Den Webbrowser werden alle Mitarbeiter verwenden, da in diesem GUI alle Funktionen der Anwendung implementiert sind.

Browserbasierter Zugang von SmartPhones Auf gängigen SmartPhones sind Browser vorhanden, die in der Lage sind, die selben Inhalten anzuzeigen, wie ihr Desktop-Equivalent. Aufgrund der deutlich kleineren Bildschirmgröße und dem Fehlen einer Maus ist es aber sinnvoll, dem Rechnung zu tragen und eine angepasste Version der Anwendung für diese Geräte bereit zu stellen.

Zugang direkt über die Schnittstellen Vor allem im Bereich der Software-Entwicklung wird der direkte Zugriff der Entwickler auf die Schnittstellen der Anwendung eine wichtige Rolle spielen. So können diese die Integration des Systems in ihren Entwicklungsprozess optimal an die jeweiligen Bedürfnisse anpassen.

Exporte Die Möglichkeit, die Texte des Projektes in verschiedene Formate zu exportieren ist eine wichtige Funktion. Sie ermöglicht zum einen die Übernahme in System und Programme, deren Anbindung nicht möglich oder gewünscht ist und schafft zum anderen die Möglichkeit, ähnlich wie Schnittstellen, die verarbeiteten Daten auf eine Art und Weise zu verwenden, die nicht vorhergesehen wurde oder nicht im Sinne der Anwendung liegt.

Zugang über Plug-Ins Besonders für Mitarbeiter in der Produktion kann es wichtig sein, auf ihre angestammten Werkzeuge nicht verzichten zu müssen. Plug-Ins, also Erweiterungen für diese Werkzeuge, integrieren dann Teile der Funktionen der Anwendung in diese Werkzeuge. Beispielsweise könnte es ein Plug-In für *Adobe InDesign* ermöglichen, auf die Texte aus dem System zuzugreifen und diese direkt in Text-Rahmen im *InDesign*-Dokument zu übernehmen, so dass Copy&Paste der Texte aus der Web-Anwendung oder einem exportierten Dokument entfallen kann.

Benachrichtigungen Benachrichtigungen sind eine Form des Zugangs, die es Benutzern des Systems ermöglicht, über bestimmte Ereignisse informiert zu werden. Benachrichtigungen können in Form von E-Mails, SMS, über soziale Netzwerke wie Twitter, über Chat-Dienste wie Skype, IRC statt finden. Hierbei werden Information zu einem Ereignis übertragen z.B. der Status-Änderungen eines Textes. Denkbar sind auch auch Push-Exporte der Texte aufgrund eines bestimmten Ereignisses, z.B. der FTP-Export der Texte als CSV-Datei, sobald das Projekt abgeschlossen ist oder Änderungen freigegeben wurden.

Software von Drittanbietern Dadurch, dass alle Funktionen der Anwendung über eine Api exponiert werden, sind auch fremde Softwarehersteller in der Lage, Teile der Funktionen oder die gesamte Anwendung in einer eigenen GUI zu implementieren. So können Be-

dürfnisse von Anwendergruppen mit besonderen Anforderungen abgedeckt werden, die bei der Konzipierung der Anwendung nicht berücksichtigt wurden.

5.6 Überblick über den Aufbau des gesamten Systems

Abbildung 7 liefert einen Überblick über den Aufbau des gesamten Systems:

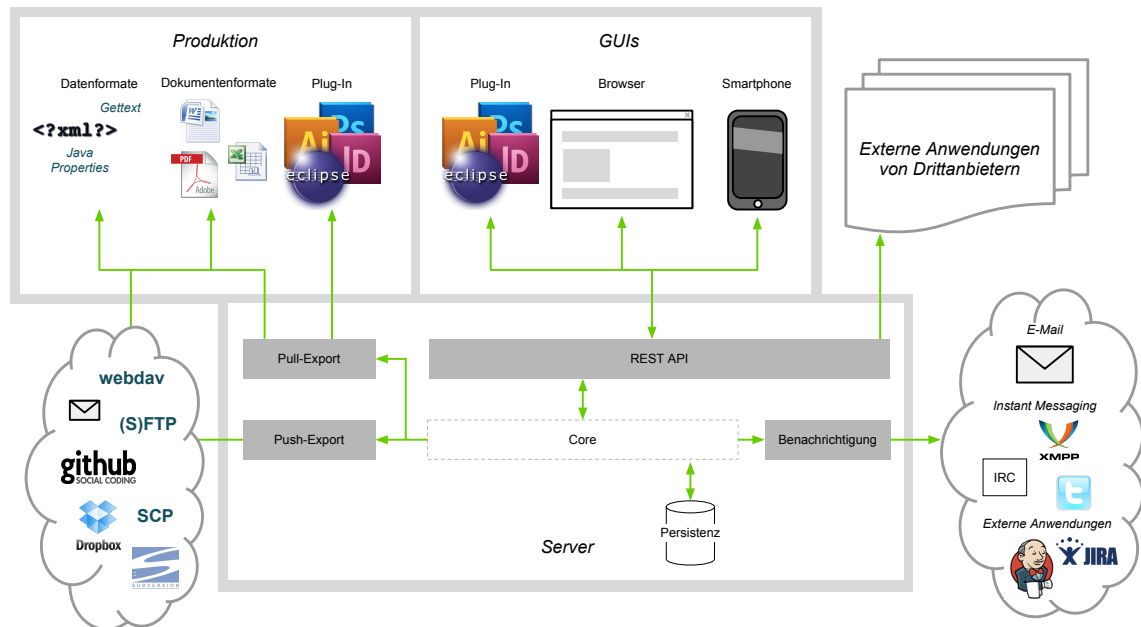


Abbildung 7: Aufbau des gesamten Systems im Überblick

Die Zentrale Komponente der Anwendung bildet der Server. Für die Benutzer erfolgt der Zugriff mit Hilfe einer GUI, die mit der REST-API des Servers kommuniziert. Eine browserbasierte GUI auf Basis von HTML5 und JavaScript bildet den Hauptzugang zum System, der auch auf SmartPhones verwendet werden kann. Zusätzlich gibt es spezielle Plugins für Adobe-Produkte und weitere wichtige Produktionsumgebungen. Auch native GUIs für Smartphones verwenden die gleiche API. Die Schnittstellen können auch von Drittanbietern dazu verwendet werden, eigenen Clients für das System zu entwickeln. In die Endprodukte gelangen die Texten über den Export, exportiert wird dabei in viele Formate, neben Datenformaten wie z.B. XML werden auch Dokumentenformate wie z.B. Word exportiert. Der Export kann durch den Anwender erzeugt werden (*Pull-Export*), aber auch automatisch, z.B. nach festgelegten Zeitplänen oder Ereignissen erfolgen. Dieser *Push-Export* erfolgt auf je nach Projekt festlegbaren Orte, wie z.B. FTP-Server oder Versionsverwaltungssysteme. Die Benachrichtigungen über Aufgaben und Änderungen an Texten kann via E-Mail, aber auch mittels Instant-Messaging-Systeme oder durch den Aufruf fremde API-Endpunkte erfolgen.

5.7 Zusammenfassung, Nachteile & Risiken des Konzepts

In diesem Kapitel wurde eine Anwendung konzipiert und der darin abgebildete Workflow beschrieben. Die Konzipierung der Anwendung als Web-Anwendung, bei der alle durchführbaren Operationen über Schnittstellen abgedeckt sind, ermöglicht es, für jeden Mitarbeiter die passenden Zugangswege anzubieten. Als Hauptzugang wird der Webbrowser verwendet, so ist sichergestellt, dass alle Mitarbeiter alle Funktionen des Systems ohne zusätzliche Aufwände wie die Installation neuer Software verwenden können. Für spezielle Anwendungsfälle ist es mit Hilfe der API möglich, Plug-Ins zu entwickeln, die sich in die bevorzugten Werkzeuge der Anwender integrieren.

Nachteile & Risiken Ein Nachteil dieses Konzepts liegt in der Zentralisierung der Datenspeicherung. Da alle Daten auf einem zentralen Server verwaltet werden, ist dieser auch der *Single Point of Failure*, d.h. sollte der Server ausfallen, kann kein Mitarbeiter weiterarbeiten. Für einen kommerziellen Betrieb eines solchen Systems ist es also unabdingbar, dass die Server-Infrastruktur ausfallsicher konzipiert ist.

Das Übertragen der Daten auf einen zentralen Server kann auch zu Bedenken bei den beteiligten Unternehmen führen. Es gibt gerade bei größeren Unternehmen Vorbehalte dagegen, Informationen auf Systemen von Drittanbietern zu speichern. Hier gilt es, genau wie im Hinblick auf die Verfügbarkeit des Systems, einen vertrauenswürdigen Betreiber für die Server-Infrastruktur zu finden. Alternativ ist es jedoch problemlos möglich, das System auch *In-House*, also auf Servern im Unternehmen als *Appliance*, zu betreiben, wobei dann aber zusätzliche Wartungsaufwände entstehen, und damit einige Vorteile des SaaS-Modells ausgehebelt werden.

Da alle Mitarbeiter über das Internet mit der Anwendung verbunden sind, spielt auch die Bandbreite und Verfügbarkeit einer Internetverbindung eine Rolle. Im Unternehmensbereich spielt dies aber inzwischen nur noch eine untergeordnete Rolle. Trotzdem sollten geeignete Maßnahmen ergriffen werden, die die Arbeit auch mit einer langsamen oder sogar ganz ohne eine Internetverbindung ermöglicht (Offline-Access).

Das größte Risiko dieses Konzepts ist, dass Mitarbeiter gezwungen werden, sich von ihren bekannten Werkzeugen zu lösen. Gerade bei Mitarbeitern, die vor allem mit Textverarbeitungsprogrammen arbeiten und ansonsten kaum mit anderen Werkzeugen Kontakt haben, wird der Umstieg von der unstrukturierten Arbeitsweise in *Word* auf die, bis auf den einzelnen Text heruntergebrochene Arbeitsweise in der vorgeschlagenen Anwendung, schwer fallen. Man kann aber davon ausgehen, dass für alle Beteiligten die Vorteile der Lösung erkenntlich werden und sich eine Abneigung gegen eine Änderung angestammter Arbeitsabläufe leicht abbauen lässt.

Im nächsten Kapitel wird eine Anwendung entworfen, die dieses Konzept umsetzt.

6 Entwurf des zentralen Systems

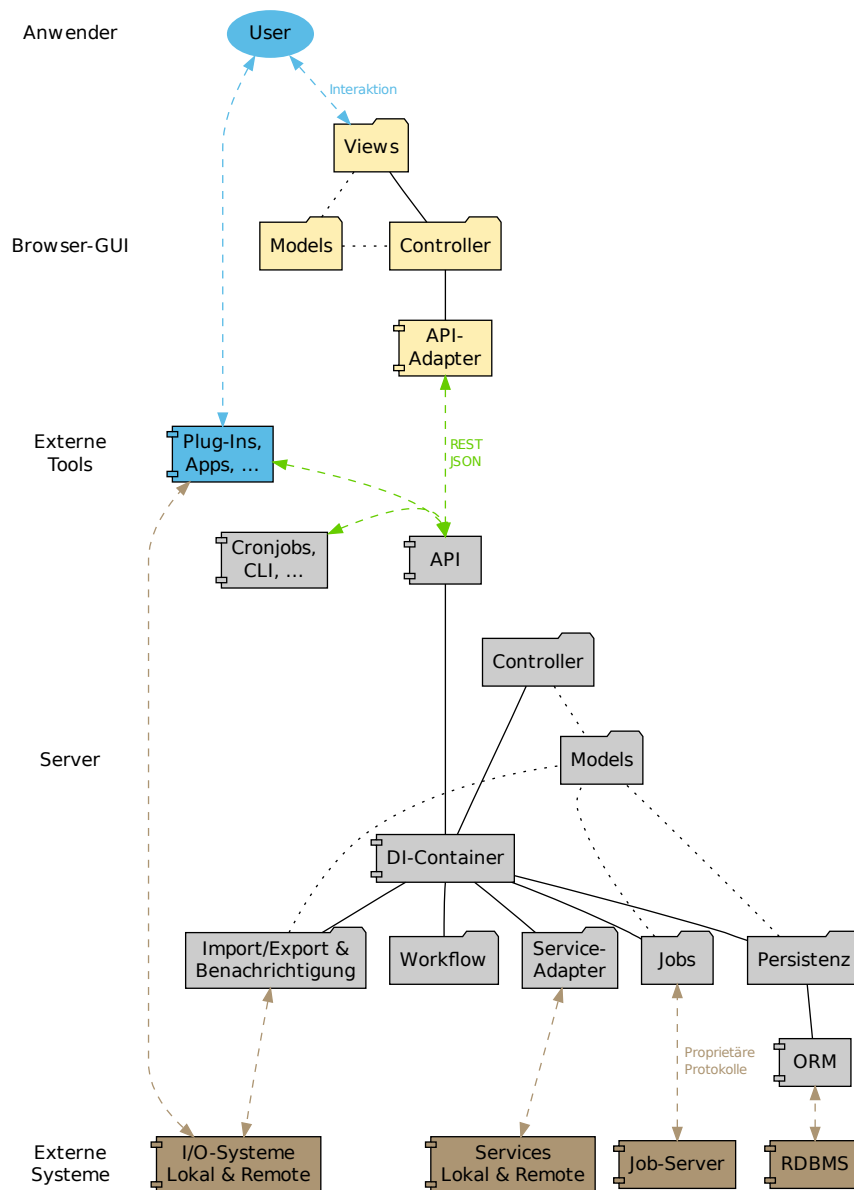


Abbildung 8: Komponententenden des zentralen Systems

Im vorigen Kapitel 5 / S.28 wurde eine Anwendung konzipiert, die das gesamte Spektrum der Anforderungen abdeckt, die in den vorangegangenen Kapiteln beschrieben wurden. In diesem Kapitel werden die Komponenten beschrieben, die zur Erfüllung der wichtigsten Anforderungen benötigt werden.

Abbildung 8 liefert einen Überblick über die in den folgenden Abschnitten beschriebenen Komponenten des Systems. *Blau* eingezeichnet sind der Endanwender und externe Benutzer-

Werkzeuge wie z.B. Plug-Ins und Software von Drittanbietern. *Gelb* eingezeichnet sind die Komponenten des browserbasierten GUI. *Grau* eingezeichnet sind die Komponenten des Servers. *Braun* eingezeichnet sind (aus Sicht des Servers) externe Dienste, die durch diesen angebunden werden. *Gestrichelte Linien* zeigen Kommunikationsverbindung zwischen Komponenten über Schnittstellen. *Durchgezogene Linien* zeigen Verbindungen innerhalb eines Systems.

Der zentrale Bestandteil der Anwendung ist der Server, mit dem verschiedene GUIs über Schnittstellen kommunizieren. Das wichtigste GUI ist das browserbasierte Interface, da hiermit alle Mitarbeiter arbeiten, dessen Gestaltung in Abschnitt 6.1 / S.41 und Anbindung an den Server in Abschnitt 6.2 / S.49 beschrieben wird. Abschnitt 6.4 / S.53 beschreibt die Komponenten des Servers. Abschließend wird in Abschnitt 6.6 / S.60 das Domänenmodell vorgestellt, das allen Teilen der Anwendung zu Grunde liegt.

Zur Validierung des Entwurfes wurde ein Querschnitt durch die vorgestellte Architektur also Prototyp umgesetzt, mit dessen Hilfe sich überprüfen lässt, inwieweit sich das vorgeschlagene Konzept praktisch einsetzen lässt und die entworfene Systemarchitektur geeignet ist, die benötigten Funktionen abzubilden. In diesem Kapitel wird der Entwurf direkt den jeweiligen im Prototypen umgesetzten Komponenten gegenübergestellt.

6.1 Entwurf eines browserbasierten GUI

Das browserbasierte GUI der Anwendung ist der wichtigste Zugang zum System, da es von allen Mitarbeitern zu irgendeinem Zeitpunkt verwendet wird. Die Zeit, die der Einzelne damit verbringt, kann sich je nach Rolle stark unterscheiden. Neben den sehr unterschiedlichen Anforderungen an das GUI sind die Benutzer auch technisch sehr unterschiedlich stark versiert, wie in den Personas in Kapitel 4 / S.18 gezeigt wurde. Diesen Umständen muss bei der Gestaltung eines GUI Rechnung getragen werden. In Anlehnung an [11] wurden für den Entwurf der browserbasierten GUI folgende Leitlinien gewählt:

1. Das Wichtigste zuerst: Die aktuelle Aufgabe soll immer im Fokus der Darstellung liegen.
2. Schnell zum Ziel: Alle Aufgaben müssen leicht und unkompliziert durchführbar sein.
3. Nicht ablenken: Veränderungen in der Darstellung, z.B. durch Benachrichtigungen, lenken ab und müssen deswegen so gestaltet sein, dass diese sich nach den Präferenzen des Nutzers richten.
4. Hilfe nur einen Klick entfernt: Das Hilfesystem muss kontextsensitiv verfügbar sein und ist eine Kernfunktion der Anwendung

Da die Anwendung von allen Benutzern gerne verwendet werden soll und vor allem die Usability und die Zeitersparnis ein wichtiger Punkt sind, wie man auch skeptische Mitarbeiter überzeugen kann, mit ihren alten Gewohnheiten zu brechen, ist die Beachtung dieser Grundsätze essentiell. Funktionalitäten, die nur von wenigen gebraucht werden, sollten, wenn überhaupt, optional einblendbar sein. Die Konzeption des Systems ermöglicht

Operation (vgl. 5.1)	Eva Konz.	Lotte Des.	Torsten Texter	Jorinde Übersetz.	Jan Prod.	Arthur Projektl.	Markus Kunde
Definieren	●	◐	○	○	○	○	○
Schreiben	◐	○	●	●	○	○	○
Korrektur	○	○	◐	◐	○	◐	◐
Qualitätskontrolle	◐	○	○	○	○	●	◐
Freigabe	○	○	○	○	○	◐	◐
Veröffentlichung	○	○	○	○	◐	○	○

Tabelle 5: Umfang und Häufigkeit der Benutzung des browserbasierten GUIs bezogen auf die durchgeführten Operation

es leicht, zusätzliche Anwendungen für besondere Benutzergruppen zu schaffen, z.B. Plug-Ins für die speziellen Werkzeuge einzelner Mitarbeiter. Für diese »Power-User« ist dieser Weg meistens der bessere Weg.

Anhand der Personas lässt sich ermitteln, wer und in welchem Umfang das browserbasiert GUI verwenden wird. Tabelle 5 / S.42 zeigt in der Übersicht, welche Operationen besonders bei der Entwicklung des GUIs beachtet werden müssen. Besonders viel Zeit werden von Konzeptern und von den Mitarbeiter, die die Inhalte erstellen, kontrollieren und freigeben im GUI verbracht, da diese Tätigkeiten bezogen auf die einzelnen Texte des Produktes sehr arbeitsintensiv sind.

Die folgenden Wireframes zeigend dementsprechend die wichtigsten Ansichten des GUIs.

6.1.1 Aufbau des GUIs

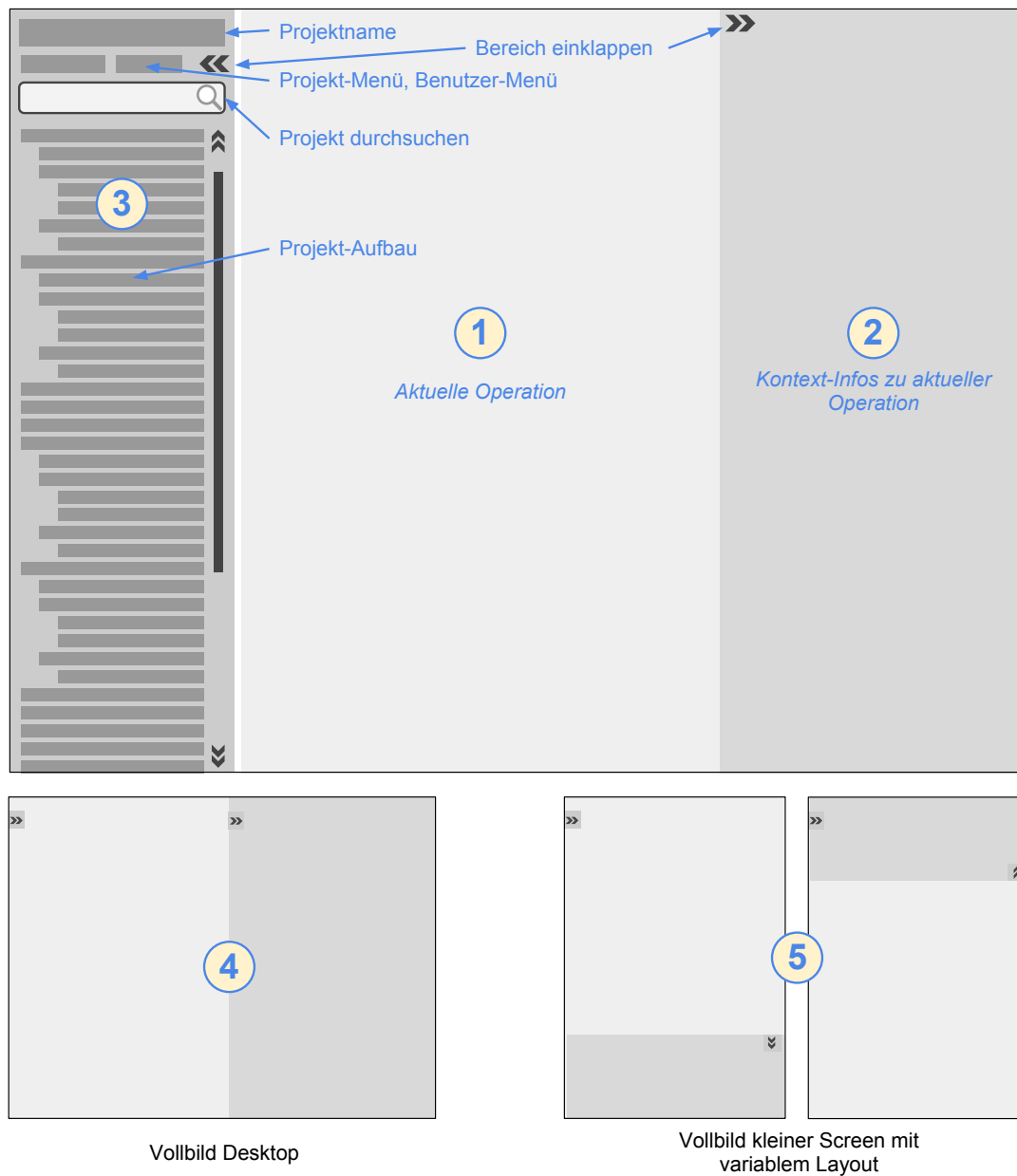


Abbildung 9: Aufbau des browserbasierten GUIs

Abbildung 9 zeigt den Grundaufbau des GUIs. Zentraler Bereich ist die Darstellung der *aktuellen Operation* ① mit den zugehörigen *Kontext-Informationen* ②. Auf der linken Seite findet sich eine Spalte über die im Projekt navigiert werden kann ③. Um den Fokus auf die aktuellen Aufgaben zu verbessern sind die Kontext-Information und die Projektspalte ausblendbar ④. Das gesamte Layout passt sich flexibel an verschiedene Bildschirmgrößen und -formate an, zusätzlich kann die Position der Kontext-Information an die eigenen Vorlieben angepasst werden ⑤.

Die Projekt-Spalte ③ bietet direkten Zugang zu allen Teilen des aktiven Projekts. Die Projektstruktur wird mit einem Navigationsbaum dargestellt, über den direkt zu den jeweiligen Abschnitten gesprungen werden kann. Über das Suchfeld lassen sich die Einträge im Baum filtern. In der Spalte befindet sich oben zur Orientierung der Namen des aktiven Projektes. Über das ausklappbare Projektmenü gelangt man Teilen der Anwendung, die nicht direkt über die Texte erreichbar sind, z.B. Projektauswahl, Mitarbeiterverwaltung, Export. Über das ausklappbare Benutzermenü kann man sich ausloggen, sein Profil bearbeiten und persönliche Einstellungen anpassen.

6.1.2 Definieren des Produktes

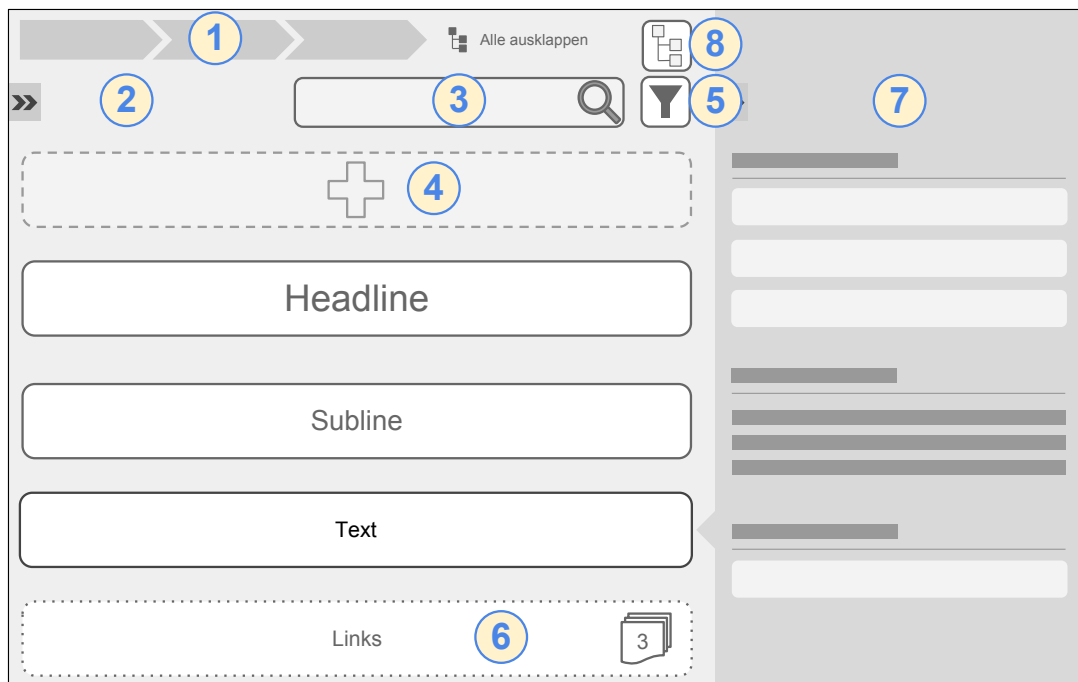


Abbildung 10: GUI zum Bearbeiten der Produktstruktur

Abbildung 10 zeigt die Darstellung des GUIs zur Bearbeitung der Produktstruktur. Über die Breadcrumb-Navigation ①, die die aktuelle Position in der Projekthierarchie zeigt, ist eine Orientierung auch ohne das eingeklappte Projektmenü möglich. Über eine Schaltfläche lassen sich alle Container ober der aktuellen Hierarchieebene aufklappen, so ist können umfangreiche Änderungen an vielen Elementen gleichzeitig vorgenommen werden, ohne dass man sich durch die verschiedenen Ebenen »hangeln« muss. Im Inhaltsbereich ② werden die auf der aktuellen Ebene befindlichen Inhalts-Elemente, Textbausteine und Container, angezeigt. Die Darstellung erfolgt in eine kompakten Weise, die einen schnellen Überblick über die Inhalte bietet ohne viel Scrollen zu müssen. Die einzelnen Inhalts-Elemente können mit Drag&Drop in ihre Reihenfolge angepasst werden. Über das Suchfeld ③ können die angezeigten Elemente gefiltert werden. Über eine Schaltfläche ④ können neue Inhalts-

Elemente erstellt werden. Zusatzinformationen wie der Typ, der Status, Kommentare oder ähnliche können auch verwendet werden, um die Inhalte auf der Seite zu filtern, z.B. um nur die Element anzuzeigen, für die noch die Inhalte fehlen ⑤. Durch Klick auf einen Container ⑥ gelangt man eine Ebene tiefer, in diesen Container hinein. Das Icon gibt die Anzahl der untergeordneten Element an. In der Spalte für die Kontext-Information ⑦ werden Detailinformation zum aktuell ausgewählten Element dargestellt, die an die eingestellten Ansichts-Art angepasst sind. Hier findet sich auch die Bearbeitungshistorie und die Kommentare zu den Inhalts-Elementen.

Über eine Schaltfläche ⑧ kann zwischen den vier Ansichten (Definieren 6.1.2, Texten 6.1.3, Übersetzen 6.1.4 und Kontrolle 6.1.5), die die Inhaltselemente manipulieren, gewechselt werden.

6.1.3 Texte erstellen

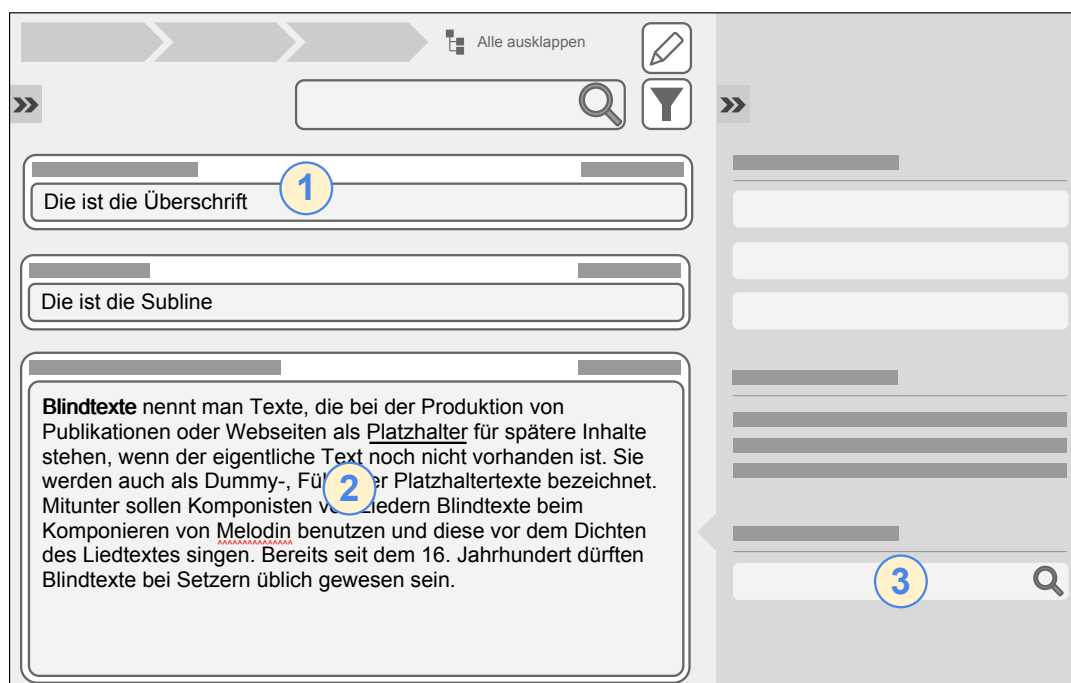


Abbildung 11: GUI zum Erstellen der Texte

Abbildung 11 zeigt die Darstellung des GUIs zur Bearbeitung der Texte des Produktes. Die Inhaltselemente in der aktuelle Hierarchie werden dabei mit Eingabefeldern für die Inhalte dargestellt ①. Links über dem Eingabefeld wird die Bezeichnung des Textbausteins angezeigt und rechts darüber Hinweise zu Längebeschränkungen (falls vorhanden) mit Zähler. Während der Eingabe können bereits Auszeichnungen vorgenommen werden, die Schaltflächen dazu, werden eingeblendet, sobald sich der Cursor im Feld befindet. Innerhalb der Kontext-Informationen ③ besteht die Möglichkeit direkt eine Suche zu starten, die Suchergebnisse werden in einem Dialog-Fenster geöffnet.

6.1.4 Texte übersetzen



Abbildung 12: GUI zum Übersetzen der Texte

Abbildung 12 zeigt die Darstellung des GUIs zur Übersetzung der Texte des Produktes. Hierzu wird der Inhalt des Textbausteines in der Original-Sprache angezeigt und darunter ein Eingabefeld, in dem die Übersetzung eingetragen wird ①. Über einen Filter-Dialog ② kann konfiguriert werden, welche Sprachen angezeigt werden sollen.

6.1.5 Prüfen



Abbildung 13: GUI zum Überprüfen der Texte

Abbildung 13 zeigt die Darstellung des GUIs zur Kontrolle und Freigabe der Texte des Produktes. In dieser Ansicht werden die verschiedenen Stati der Textbausteine bearbeitet. Hierzu sind die Textbausteine mit ihren Übersetzungen dargestellt ①. Über die mit Icons markierten Schaltflächen ② lassen sich die einzelnen Stati direkt setzen. Von links nach rechts sind das: Korrigiert, Geprüft, Freigegeben und Veröffentlicht. Es wird jeweils zwischen »keine Angabe« (grau), »abgelehnt« (rot) und »in Ordnung« (grün) unterschieden. Hier wird auch die Anzahl der hinterlegten Kommentare angezeigt. Zum massenhaften bearbeiten von Stati können mehrere oder alle Elemente über die Checkboxes ③ ausgewählt werden. Über die Status-Icons in der Kopfzeile kann dann der Status für alle ausgewählten Elemente gleichzeitig gesetzt werden.

6.1.6 Kontext-Informationen



Abbildung 14: Kontext-Informationen

Abbildung 14 zeigt die Darstellung des GUIs zur Anzeige und Bearbeitung der Kontext-Informationen. Der Inhalt der Ansicht ist an die jeweilige Operation angepasst. Im oberen Bereich kann zwischen verschiedenen Kontextinformationen zu dem jeweiligen Inhalt umgeschaltet werden ① (v.l.n.r):

1. Neues: Listet die neuesten Informationen aus den vier Bereichen und den aktuellen Status.
2. Änderungshistorie: Zeigt die vergangenen Änderungen an, mit der Möglichkeit, diese zu kommentieren und zurückzunehmen.
3. Material: Zeigt hinterlegte Materialien an. Dabei handelt es sich um Dateien und Freitext-Notizen.
4. Kommentare: Zeigt die Kommentare an.
5. Struktur: Zeigt Informationen zur Struktur innerhalb des Produktes, wie z.B. die Attribute, an.

Über das Suchfeld ② lässt sich die aktuellen Ansicht filtern. In den Kontextinformation findet sich direkter Zugang zu Informationen und Operationen bezogen auf den aktuellen Inhalt. Die Inhalte lassen sich über Formulare ③ direkt bearbeiten, Notizen und Dateien werden angezeigt ④ und der Diskussion mithilfe der Kommentarfunktion ist möglich ⑤.

6.2 Anbindung der GUI an den Server

Die in Abbildung 8 / S.40 *gelb* eingezeichneten Komponenten zeigen den Aufbau eines browserbasierten GUIs. Mittlerweile ist es üblich, klassische Paradigmen aus der Softwareentwicklung auch in browserbasierten Anwendung zu verwenden. Dementsprechend wird die GUI in Form einer MVC-Anwendung implementiert. Die Kommunikation mit dem Server wird in einer eigenen Komponente, dem *API-Adapter*, gekapselt, die über die REST-Schnittstelle JSON-Daten mit dem Server austauscht. Die Repräsentation der Domänendaten erfolgt dabei mithilfe von entsprechenden *Models*, die durch die *Controller* mit den *Views* verbunden werden. Die *Models* sind nicht zwangsläufig mit den *Models* auf der Serverseite identisch sondern können Aggregationen sein, oder nur Teile der Daten abbilden, sie orientieren sich an der öffentlichen API des Servers, die nicht zwingenderweise die internen *Models* 1:1 nach außen weiterreicht. *Views* sind einzelne Bestandteile einer Ansicht. So lassen sich individuelle Bereiche der Darstellung leicht aktualisieren, ohne die ganze Seite neu laden zu müssen – in [10, S.1–5 und S.65–72] ist dieses Prinzip ausführlich erläutert.

Ein Problem beim Datenaustausch mit JSON ist, dass diese üblicherweise nur reine Daten enthält, da schlichterweise Objekte serialisiert werden. Beim Serialisieren werden dann nur die Eigenschaften einer Objekt-Instanz beibehalten, Informationen wie die Klasse oder Relationen zu anderen Objekten werden verworfen. Werden APIs nach diesem Schema verwendet, müssen die Entwickler wissen, hinter welchem Endpunkt sie welche Arten von Daten erwarten. Diese Information wird als Quellcode hart kodiert.

Um zukünftigen Änderungen an der Schnittstelle ohne Änderungen auf Clientseite begegnen zu können sollten die Antworten der Schnittstelle mit Meta-Informationen versehen sein, die es dem Client ermöglichen, ohne hart kodierte Zuordnung zwischen Endpunkt und Domänenobjekt auszukommen. *JSON-LD*⁷ hat sich dabei als Lösung bewährt. Hierbei werden JSON-Objekt mit Meta-Informationen versehen, die beschreiben, um *was* für ein Objekt es sich bei der Antwort handelt (*@context*) und um *welches* (*@id*):

```
{
  "@context": "http://json-ld.org/contexts/person.jsonld",
  "@id": "http://dbpedia.org/resource/John_Lennon",
  "name": "John Lennon",
  "born": "1940-10-09",
  "spouse": "http://dbpedia.org/resource/Cynthia_Lennon"
}
```

In dieser Art kann ein JSON-Objekt auch mit Informationen über zugehörige *andere* Objekte versehen werden. Im Beispiel verweist *spouse* auf die URL unter der das JSON-Objekt der Ehefrau abgerufen werden kann. Auf diese Weise lassen sich die vollständige Datenstruktur einer Anwendung als Graph darstellen. Hierzu werden alle Objekte mit Relationen anno-

⁷<http://json-ld.org/>

tiert und der Client kann entscheiden, welchen Relationen er folgt und welchen nicht. Die die Relations-Informationen auch die URLs zu den Endpunkten enthalten, müssen in der Implementierung keine Endpunkte mehr hart codiert werden, es ist lediglich nötig, einen Einstiegs-Punkt zu kennen von dem aus man mit sich durch den Graphen »hangeln« kann, indem man Objekt-Relationen folgt, deren Kontexte die Anwendung erkennt.⁸

6.3 Implementierung des GUI

Im Prototypen wurden das GUI als reine JavaScript-Anwendung umgesetzt. Als Framework für den Aufbau der Anwendung wurde *Backbone.js*⁹ eingesetzt. *Backbone.js* stellt Basisklassen für Models, Views und Collections (Listen von Models) zur Verfügung, die mit Hilfe eines zentralen Routers zu einer Anwendung verbunden werden können, die mit einer RESTful-API kommuniziert. Der Router übernimmt dabei die Aufgabe, abhängig von der URL im Browser zu steuern, welche View gerade eingezeigt wird.

Um die einzelnen Bestandteile der Anwendung möglichst modular zu halten und Ladezeiten so kurz wie möglich zu halten, kommt *require.js*¹⁰ zum Einsatz, mit dessen Hilfe es möglich ist, nur die für den aktuellen Code-Pfad nötigen Abhängigkeiten zu laden. Neben dem Nachladen von JavaScript-Code ist es damit auch möglich HTML-Dateien nachzuladen, dies wird in der Implementierung dazu verwendet, den HTML-Code zur Darstellung der Anwendung ebenfalls modular und nur bei Bedarf nachzuladen. Dies ermöglicht zudem eine saubere Trennung zwischen der Darstellungslogik, die in JavaScript implementiert ist und dem zugehörigen HTML-Quellcode. *Underscore.js*¹¹, auf dem *Backbone.js* aufbaut, liefert hierzu einfache Templating-Funktionen.

Zur Realisierung der GUI-Funktionalität wurde *Twitter Bootstrap*¹² eingesetzt, ein HTML5-Framework, das für viele GUI-Elemente bereits vorgefertigte Darstellungen bietet. Einige Elemente werden mit Hilfe von JavaScript mit erweiterter Funktionalität versehen, in *Twitter Bootstrap* kommt dabei *JQuery*¹³ zum Einsatz. *Twitter Bootstrap* ermöglicht von sich aus das GUI *responsive* zu implementieren, d.h. dass keine festen Maßangaben verwendet werden. So ist eine saubere Darstellung auf vielen unterschiedlichen Endgeräte sichergestellt. Mit Hilfe von CSS *media queries* können einfach Anpassung für bestimmte Bildschirmgrößen implementiert werden.

Alle genannten Komponenten sind umfangreich getestet und funktionieren in allen aktuellen Browsern und sind auch abwärtskompatibel zu älteren Browsern.

⁸<http://coderbyheart.de/blog/relationen-in-linked-data>

⁹<http://backbonejs.org/>

¹⁰<http://requirejs.org/>

¹¹<http://underscore.org/>

¹²<http://twitter.github.com/bootstrap/>

¹³<http://jquery.com/>

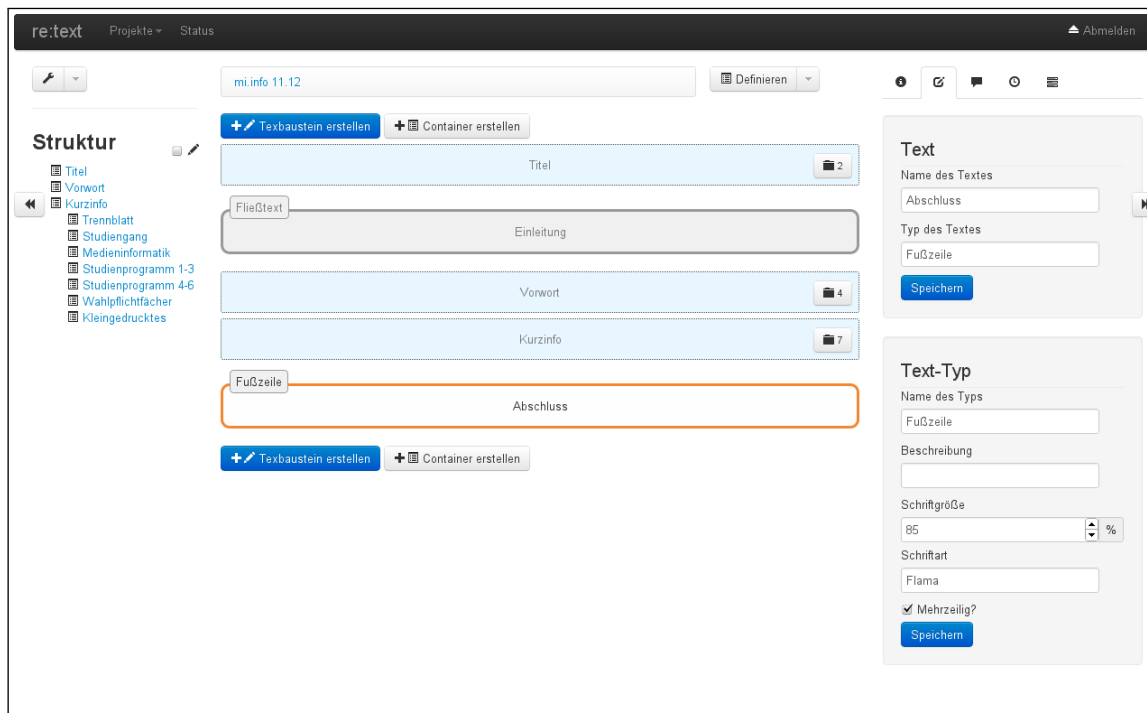


Abbildung 15: GUI des Prototypen

Abbildung 15 zeigt den Aufbau des GUI des Prototypen beispielhaft in der Ansicht zur Definition des Produktes (vgl. Abschnitt 6.1.2 / 44). Die Anordnung der Elemente aus dem Entwurf wurde weitgehend beibehalten. Unterschiede resultieren aus im Prototyp nicht implementierte Funktionen, die nicht mit Platzhalter-Elementen dargestellt werden, sondern entfallen. Weitere Screenshots finden sich in Anhang A / S.65.

6.3.1 Klassendiagramm

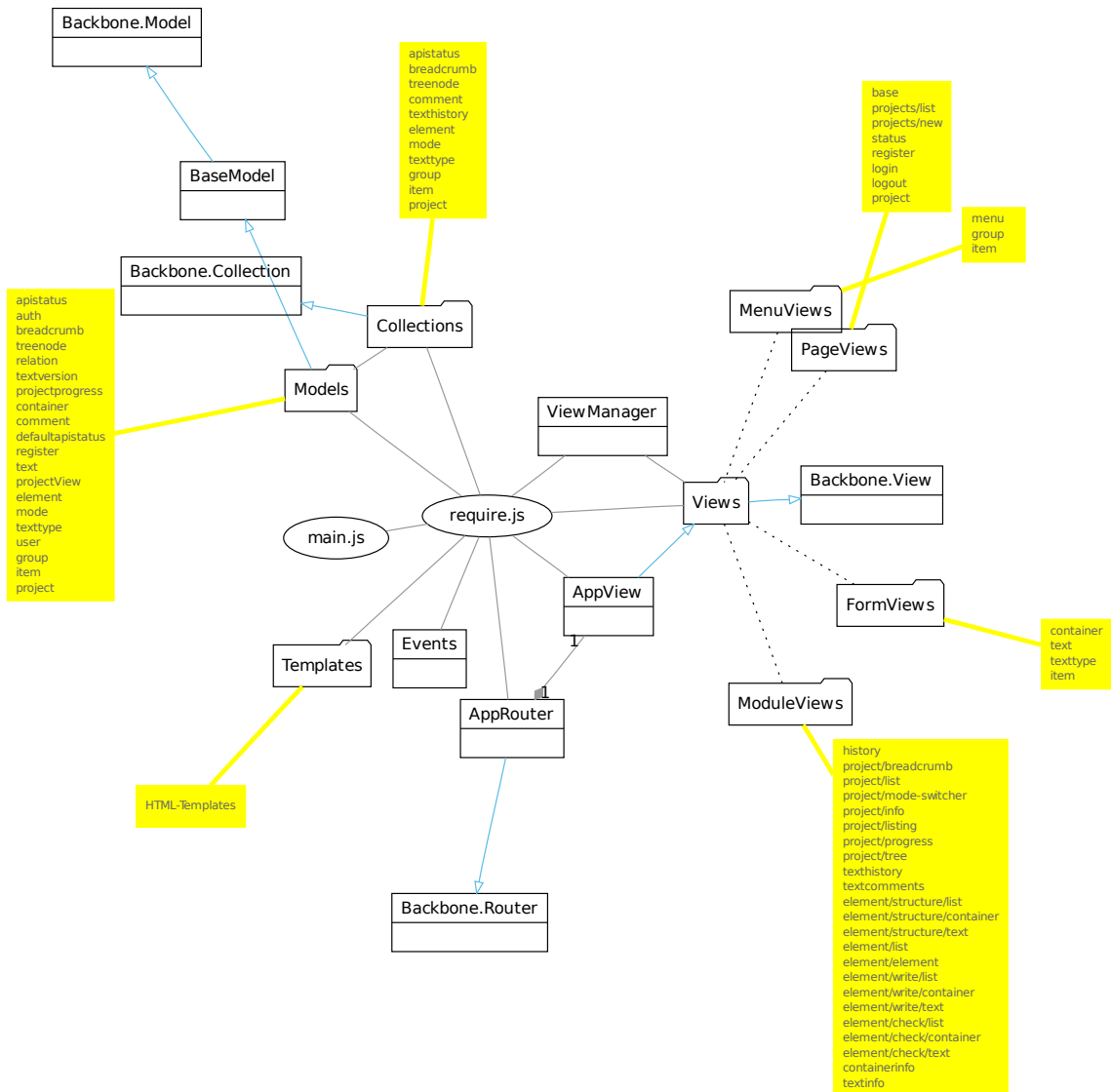


Abbildung 16: GUI-Klassendiagramm des Prototypen

Abbildung 16 zeigt das Klassendiagramm der browserbasierten Anwendung. Zentraler Einstiegspunkt ist die JavaScript-Datei `main.js`, die die Anwendung initialisiert. Dazu lädt sie mit Hilfe von `require.js` die notwendigen Komponenten. Dabei handelt es sich zum einen um die `AppView`, die grundlegende Zustände der Anwendung verwaltet und den `AppRouter`, der je nach aktueller Route die jeweilige `PageView` initialisiert und anzeigt. Die `PageViews` erzeugen danach wiederum eigenständig die von ihnen benötigten Ansichten (`ModuleViews`, `MenuViews` und `FormViews`) und laden die dafür benötigten Daten mithilfe der `Models` und `Collections` – in *Backbone.js* sind *Views* eigentlich Controller, eine Trennung zwischen Darstellungslogik und Businesslogik ist nicht vorgesehen. Den HTML-Quellcode zur Dar-

stellung laden die Views ebenfalls selbständig aus dem Templates nach. Abhängigkeiten wie JavaScript-Objekte und Templates werden dabei zentral von *require.js* bereitgestellt, dass damit die Aufgabe eines DI-Containers übernimmt und praktisch auch so verwendet werden kann.

Die Kommunikation mit der API wird durch die *Backbone.js*-Klassen `Backbone.Model` und `Backbone.Collection` abstrahiert, die RESTful-APIs ansprechen können, die dem CRUD-Paradigma entsprechen. Siehe dazu die Beschreibung der API-Implementierung in Abschnitt [6.5.1 / S.56](#).

6.4 Entwurf des Anwendungsservers

In diesem Abschnitt werden die in Abbildung 8 / [S.40](#) grau eingefärbten Komponenten des Servers beschrieben.

6.4.1 API, Cronjobs, CLI, ...

In der API-Komponenten, die als REST-API implementiert ist, sind über eine Routing-Tabelle alle Controller registriert. Da das System mit einer vollständigen Abdeckung aller Operation durch API-Endpunkte implementiert ist, wird diese auch lokalen ausgeführten Scripten wie z.B. Cronjobs oder CLIs verwendet. Dies ermöglicht eine saubere Trennung, auch für administrative Aufgaben, und verhindert, dass Sonderfälle oder Workarounds für bestimmte Aufgaben kultiviert werden.

6.4.2 Dependency-Injection-Container

Der einzelnen Komponenten des Servers sind nach dem Prinzip der losen Kopplung verbunden (vgl. [\[9, S.62\]](#)). Hierzu werden die einzelnen Komponenten als Services in einem Dependency-Injection-Container (DI-Container) registriert und von diesem instanziiert. Alle Komponenten können auf den DI-Container zugreifen und dort andere Komponenten anfordern. Der DI-Container übernimmt, wie der Name schon sagt, auch die Aufgabe Abhängigkeiten der einzelnen Komponenten bereitzustellen. Ein so aufgebautes System, bei dem konkrete Instanzen nicht mehr an der Stelle erzeugt werden, an der sie verwendet werden, sondern von außen »hereingereicht« werden, ist leicht zu modifizieren und vor allem leicht zu testen, da sich Abhängigkeiten für Tests leicht durch Mock-Objekte austauschen lassen (vgl. [\[7, Kap.2\]](#)).

6.4.3 Controller

Zentrale Komponente bilden die Controller. Hier ist die Business-Logik der Anwendung implementiert. Die einzelnen Operationen sind in verschiedenen Controllern implemen-

tiert, so ist das System leicht erweiter- und wartbar. Die Controller werden von der API-Komponente entsprechende den Routen instanziiert, für die sie registriert sind. Controller haben über den DI-Container Zugriff auf alle Teile des Systems und erzeugen die Antwort auf eine Anfrage, die mit der API-Komponenten an den Client zurück geschickt wird.

6.4.4 Models

Zu den Kern-Komponenten gehören auch die Models. Diese können ohne den DI-Container direkt von den jeweiligen Komponenten instanziiert werden, da sie reine Datenobjekt sind und werden Logik enthalten, noch Abhängigkeiten besitzen. Siehe hierzu auch das Domänenmodell auf S.60.

6.4.5 Persistenz, ORM

Mit dieser Komponente werden die innerhalb der Anwendung erzeugten Daten persistiert. Mithilfe eines Object-Relational-Mappers (ORM) werden die Domänenendaten, die durch die Models ausgedrückt werden, in relationale Daten umgewandelt, sofern zur Datenspeicherung ein relationales Datenbanksystem (RDBMS) verwendet wird. Wird hier eine nicht-relationale Datenbank (Dokumentendatenbank, Key-Value-Store) verwendet, kann der ORM entfallen bzw. wird durch ein entsprechende Komponente ersetzt.

6.4.6 Import/Export & Benachrichtigung

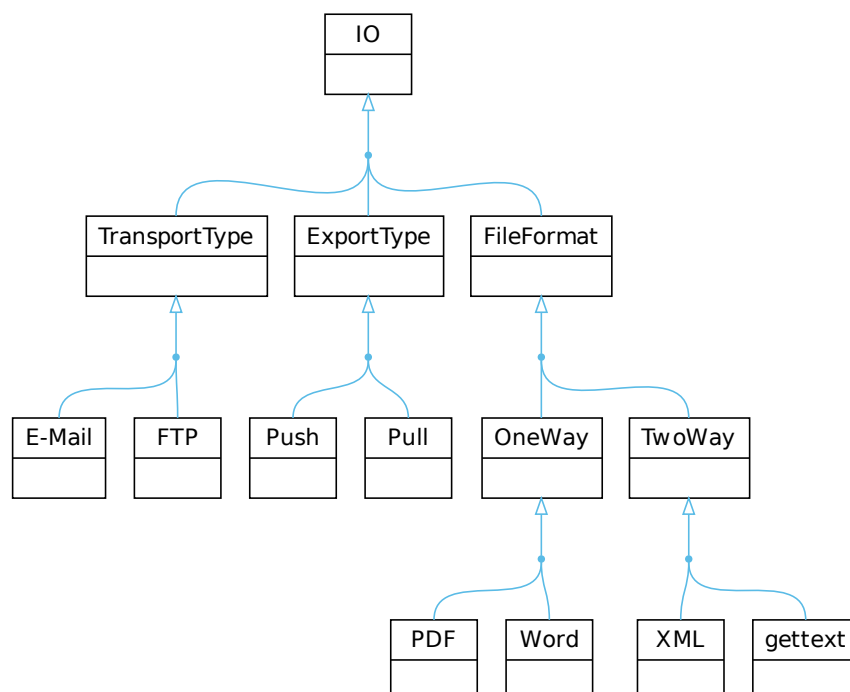


Abbildung 17: IO-Komponente

Diese Komponente ist für den Import von Daten aus verschiedenen Formaten in das interne Repräsentationsmodell und den umgekehrten Export zuständig. Die Komponente ist unterteilt in drei Bereiche, die nach dem Transport-Weg (z.B. E-Mail oder FTP), der Art des Ex- oder Importes (synchron, asynchron bzw. Push oder Pull) und nach dem Dateiformat unterscheiden (vgl. Abb. 17). Hierbei gibt es synchrone Exporte, z.B. den Download eines Exports über die Schnittstelle und asynchrone Exporte, z.B. den Upload auf einen FTP-Server. Benachrichtigungen sind ebenfalls in dieser Komponente implementiert, eine Benachrichtigung ist z.B. ein asynchroner Export via E-Mail, wobei keine Daten exportiert werden, sondern z.B. nur die Informationen zu einem Ereignis.

6.4.7 Jobs

Zeitaufwändige und wiederkehrende Operation wie z.B. der Datenimport oder das Versenden von Benachrichtigungen werden mithilfe von Jobs verarbeitet. Die Informationen zu einem Job werden in die Message-Queue eines Job-Server eingestellt und von dort registrierten Job-Runnern verarbeitet.

6.4.8 Workflow

In der Workflow-Komponente sind die Funktionen zur Definition und Ablaufsteuerung von Workflows implementiert.

6.4.9 Service-Adapter

Über Service-Adapter werden externe Dienste oder andere Softwarekomponenten angebunden, die bestimmte Funktionen für die Anwendung zur Verfügung stellen, z.B. Wörterbücher, Datei-Konverter usw.

6.5 Implementierung des Anwendungsservers

Für den Prototypen wurden das PHP-Framework *Symfony2*¹⁴ eingesetzt, das von sich aus bereits viele Anforderungen aus dem vorangegangenen Abschnitt erfüllt. Da im Kern jeder Web-Anwendung HTTP-Anfragen beantwortet werden, überträgt *Symfony2* dieses Paradigma auf die Art und Weise, wie im Framework Anfragen verarbeitet werden: mit Hilfe einer Routen-Konfiguration werden Controller für auf bestimmte Pfade in der Anfrage-URL registriert und vom Framework automatisch instanziiert und aufgerufen. Die Controller schreiben dann ihre Antwort in ein Antwort-Objekt, das vom Framework als Antwort

¹⁴<http://symfony.com/>

gesendet wird. Oberstes Prinzip bei der Entwicklung des Frameworks war die Modularisierung und die Wiederverwendbarkeit, so dass alle Komponenten nur mit Hilfe eines Service-Containers kommunizieren. Dadurch sind *Symfony2* besonders leicht zu warten und auch zu testen, entsprechende Unit-Testing-Komponenten werden bereits mitgeliefert.

6.5.1 API

Die API des Anwendungsservers wurde als RESTful-API implementiert und deckt alle Funktionen des Anwendungsservers ab. Sie folgt dem CRUD-Paradigme und unterstützt dementsprechen die jeweiligen Methoden: POST zum Erzeugen von (CREATE), GET zum Lesen von Daten (READ), PUT zum Aktualisieren bestehender Daten (UPDATE) und DELETE zum Löschen von Daten. Datenobjekte werden als JSON-Objekte übertragen, bei GET-Anfragen sind URL-Parameter zulässig. Eine Aufstellung alle API-Endpunkte findet sich in Anhang [B / 69](#).

Die ausgelieferten JSON-Objekte enthalten Zusatzinformationen zum Kontext und zu Relationen, wie in Abschnitt 6.2 / S.49 beschrieben.

```
{
  "@context": "http://jsonld.retext.it/Container",
  "@subject": "/api/container/4fdf26e7820b905118000001",
  "@relations": [
    {
      "@context": "http://coderbyheart.de/jsonld/Relation",
      "relatedcontext": "http://jsonld.retext.it/Project",
      "href": "/api/project/4fdf26e7820b905118000000",
      "list": false
    },
    {
      "@context": "http://coderbyheart.de/jsonld/Relation",
      "relatedcontext": "http://jsonld.retext.it/Element",
      "role": "http://jsonld.retext.it/ontology/child",
      "href": "/api/element?parent=4fdf26e7820b905118000001",
      "list": true
    },
    ...
  ],
  "id": "4fdf26e7820b905118000001",
  "name": "Abschnitt 1",
  ...
}
```

Dieses Listing zeigt als Beispiel das JSON-Objekt eines Containers. In der @relations-Liste ist der Verweis auf das Eltern-Element enthalten, aber auch auf die Liste mit untergeordneten Elementen dieses Containers. Im Client werden die Relationen je nach Bedarf durchsucht und die passende ausgewählt, sofern weitere Daten benötigt werden. Die jeweilige URL ist in href enthalten, so dass sich im Quellcode des Clients nur wenige hart-kodierte Endpunkte finden.

6.5.2 Persistenz

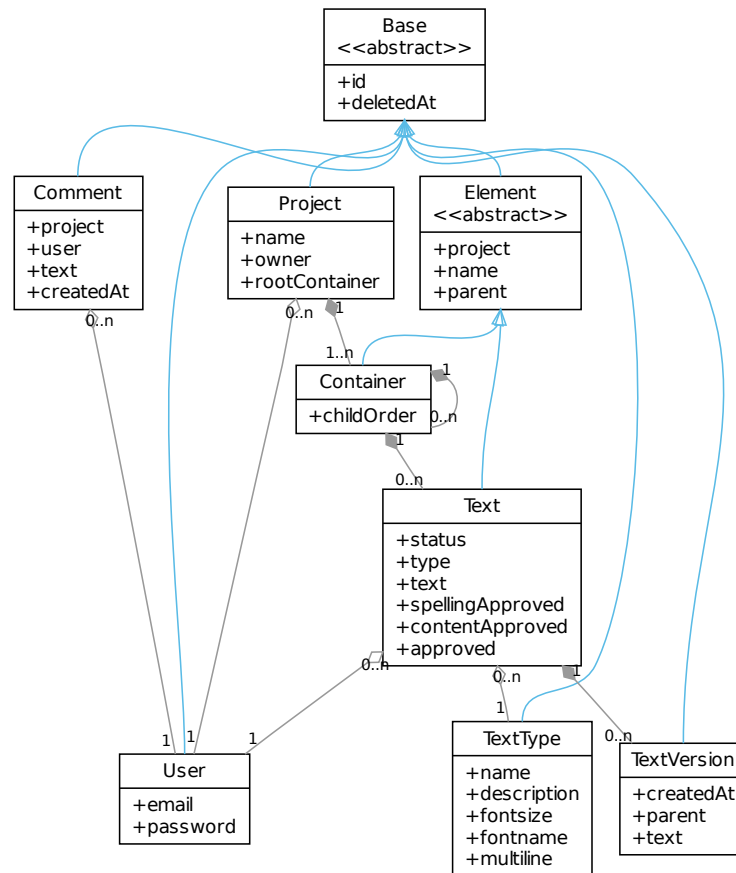


Abbildung 18: Persistierte Objekte im Prototypen

Zum Persistieren der Daten kommt *MongoDB*¹⁵, eine No-SQL-Datenbank zum Einsatz, diese ermöglicht das unkomplizierte speichern auch komplexer Dokumentenstrukturen. Die Wahl einer nicht-relationen Datenbank bietet den Vorteil, dass aufwändiges Zusammensetzen und Zerlegen von Dokumenten entfallen kann. In der Datenbank werden die in Abbildung 18 gezeigten Domänenobjekte gespeichert. Das Laden- und Speichern ist durch den Einsatz eines Object-Document-Mappers (ODM) für Doctrine 2 und MongoDB¹⁶ weitestgehend automatisiert. Dieses Modell ist deutlich einfacher als das in Abschnitt 6.6 / S.60 vorgestellte Domänenmodell, da im Prototyp nur wenige, entscheidende Funktionen implementiert wurden.

¹⁵<http://www.mongodb.org/>

¹⁶<https://github.com/doctrine/mongodb-odm>

6.5.3 Klassendiagramm

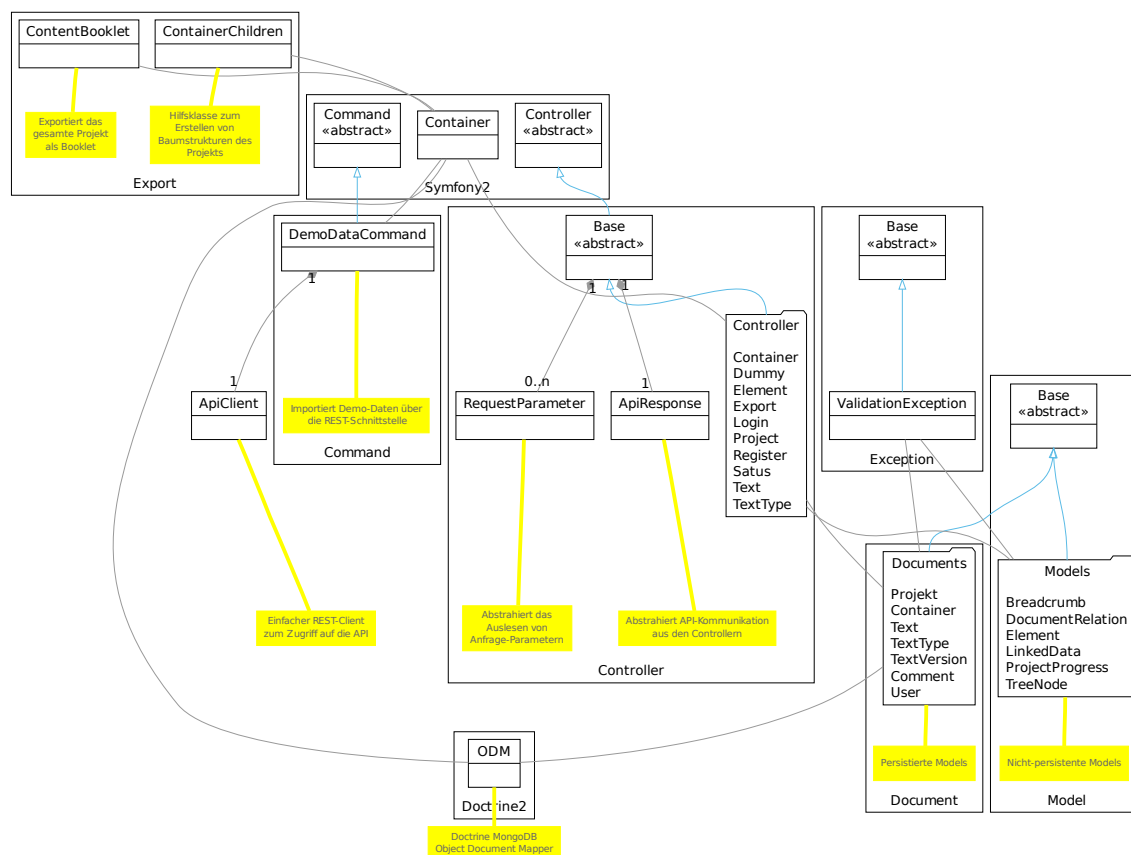


Abbildung 19: Anwendungsserver-Klassendiagramm des Prototypen

Abbildung 19 zeigt das Klassendiagramm der Implementierung des Prototypen. Zentrale Komponente ist der Dependency-Injection-Container (DI-Container) des Symfony2-Frameworks, dieser instanziiert je nach Pfad der Anfrage einen Controller und ruft dort die entsprechende Methode auf. Dieses Listing zeigt die am Beispiel des LoginControllers, in dem die Methode loginAction aufgerufen wird, wenn der Pfad login lautet und die HTTP-Methode der Anfrage POST ist.

```
class LoginController extends Base {
    /**
     * @Route("/login", requirements={"_method":"POST"})
     */
    public function loginAction() { ... }
}
```

Innerhalb der Controller steht der DI-Container zur Verfügung über den dann z.B. der ODM instanziiert werden kann, um Daten aus der Datenbank zu laden. Der ODM verwendet zur

Aus den vorangegangenen Überlegungen zur Anwendung und zum Workflow lässt sich ein Domänenmodell extrahieren, die einzelnen logischen Objekte innerhalb der Anwendung beschreibt, mit deren Hilfe alle Operationen abgebildet werden. Abbildung 20 zeigt das Modell in der Übersicht.

Attribut

Beschreibt die Attribute eines Textbausteins.

Benutzer

Repräsentiert einen Benutzer des Systems

Container

Containern dienen zur hierarchischen Organisation der Texte innerhalb des Projektes. Containern können weitere Container und Texte enthalten. Eine Container ohne übergeordneten Container befindet sich auf der obersten Ebene. Es kann mehrere Containern auf der obersten Ebene geben.

Gruppe

Mitarbeiter können in Gruppen zusammengefasst werden. Dies erleichtert die Konfiguration des Workflows und der Rechte.

Kommentare

Kommentare enthalten Hinweise und Fragen zu einzelnen Texten und Textbausteinen.

Kontext-Information

Projektspezifische Kontext-Informationen lassen sich hinterlegen und Textbausteinen und Containern zuordnen.

Projekt

Projekte bildet den Rahmen für alle Texte eines einzelnen Produktes.

Projektmitarbeiter

Gestattet einem Benutzer die Mitarbeit an einem Projekt und legt dabei fest, welche Rechte dem Benutzer für das Projekt zustehen.

Recht

Beschreibt ein Recht, eine Operation auf einem Objekt auszuführen.

Rolle

Beschreibt die verschiedenen Rollen innerhalb der Anwendung. Die Rechte der Rollen sind durch die Zuordnung von Benutzern zu Projekten durch den Projektmitarbeiter immer an das jeweilige Projekt gebunden.

Sprache

Die Texte jedes Projekts liegen in einer oder mehreren Sprachen vor.

Status

Beschreibt die verschiedenen Zustände eines Textbausteins (vgl. Abschnitt 5.1.3 / S.32).

I. Neu, Textbaustein erzeugt

2. Leer, Textbaustein definiert
3. Befüllt, Textbaustein mit Inhalt befüllt
4. Korrigiert, Orthografie geprüft
5. Geprüft, Inhalt geprüft (Qualitätsicherung)
6. Freigegeben, durch Kunden freigegeben
7. Veröffentlicht, in Produkt übernommen

Statusänderung

Beschreibt eine Änderung eines Status durch einen Benutzer, z.B. durch Freigabe oder Überprüfung.

Textbaustein

Beschreibt einen einzelnen Textbaustein.

Version

Beschreibt eine Version des Inhalts eines Textbausteins.

Workflow

Beschreibt einen projektspezifischen Workflow.

6.7 Implementierter Workflow

Innerhalb der Anwendung wird das Projekt angelegt und die dafür benötigten Textbausteine definiert. Hierbei können detaillierte Angaben zu deren Eigenschaften gemacht werden, z.B. über den Verwendungszweck oder die maximal Länge. Die einzelnen Textbausteine werden bei diesem Vorgang entsprechend dem Aufbau des Endproduktes in eine Reihenfolge gebracht und hierarchisch angeordnet. So wird eine leichte Orientierung und Zuordnung der Text zum Endprodukt möglich.

Nachdem die benötigten Textbausteine definiert wurden, werden diese durch Texter befüllt. Für Texter stellt die Anwendung Hilfsfunktionen zur Verfügung. Dazu zählen Informationen wie Zeichenlänge und Wortanzahl und Rechtschreibkorrektur mit Wörterbuch.

Sobald die Texte hinterlegt wurden durchlaufen sie die Qualitätskontrolle durch andere Mitarbeiter des Projektes und anschließend den Freigabeprozess beim Kunden. Wurden die Texte freigegeben, können die zusammengestellten Texte in das Endprodukt übernommen werden.

Alle Vorgänge werden innerhalb der Anwendung protokolliert und sind so für jeden Beteiligten leicht nachvollziehbar. Aufgaben können automatisch aufgrund von Änderungen erzeugt werden, oder von Mitarbeiter angelegt werden. So wird sichergestellt, dass alle Projektmitarbeiter jederzeit über ihre Aufgaben bezüglich der Texte informiert sind, bei Änderungen die verantwortlichen Mitarbeiter informiert werden. Dadurch wird es möglich auch bei Korrekturen in letzter Minute diese Änderungen gezielt und transparent zu übernehmen.

6.8 Umsetzung am Beispiel des Studiengangsflyers

Die vorgeschlagene Lösung wird anhand eines realen Projekts auf ihre Praxistauglichkeit hin überprüft. Es handelt sich dabei um die einmal im Jahr erscheinende Informationsbroschüre des Studienganges Medieninformatik an der Hochschule RheinMain. Die Broschüre zu Beginn des Wintersemesters 2011/2012 einen Umfang von 28 Seiten zuzüglich Titel und Rückseite. In ihr findet sich das Grußwort des Studiengangsleiters, eine Kurzinfo über den Studiengang, das Studienprogramm mit Informationen zum Verlauf des Studiums, ein Terminkalender, Informationen zu Einrichtungen des Fachbereiches, eine Liste mit Personen im Fachbereich, sowie eine Umgebungskarte und ein Gebäudeplan. Die Broschüre wird von den Mitarbeitern des Fachbereiches selber erstellt.

6.9 Zusammenfassung

In diesem Kapitel wurde das zentrale System der Anwendung, bestehend aus Server und browserbasierten GUI entworfen und beschrieben. Der Aufbau des Servers mithilfe einer lose gekoppelten Architektur ermöglicht einfache Erweiterbarkeit und Wartung. Das browserbasierte GUI ist als JavaScript-MVC-Anwendung entworfen, die über eine REST-API mit dem Server kommuniziert. Die wichtigsten Ansichten des GUIs wurden mit Hilfe von Wireframes beschrieben.

7 Fazit

A Screenshots

The screenshot shows the registration page of the re:text application. The header bar is dark grey with the logo 're:text' and a 'Status' link on the left, and 'Registrieren' and 'Anmelden' links on the right. The main heading is 'Registrieren'. Below it, a subtext reads: 'Zum Verwenden dieser Anwendung musst Du dich mit deiner E-Mail-Adresse registrieren.' The registration form is a light grey box containing a text input field with the placeholder 'name@domain.de' and a blue 'Registrieren' button.

Abbildung 21: Registrierung für neue Benutzer

The screenshot shows the login page of the re:text application. The header bar is dark grey with the logo 're:text' and a 'Status' link on the left, and 'Registrieren' and 'Anmelden' links on the right. The main heading is 'Anmelden'. Below it, a subtext reads: 'Logge dich hier mit deiner E-Mail-Adresse und deinem Passwort ein.' There is a link 'Noch keinen Account?' in blue. The login form is a light grey box containing two text input fields: one for the email address (placeholder 'name@domain.de') and one for the password (placeholder 'Passwort'). Below the email field is a blue link 'Passwort vergessen?'. A blue 'Anmelden' button is to the right of the password field.

Abbildung 22: Login

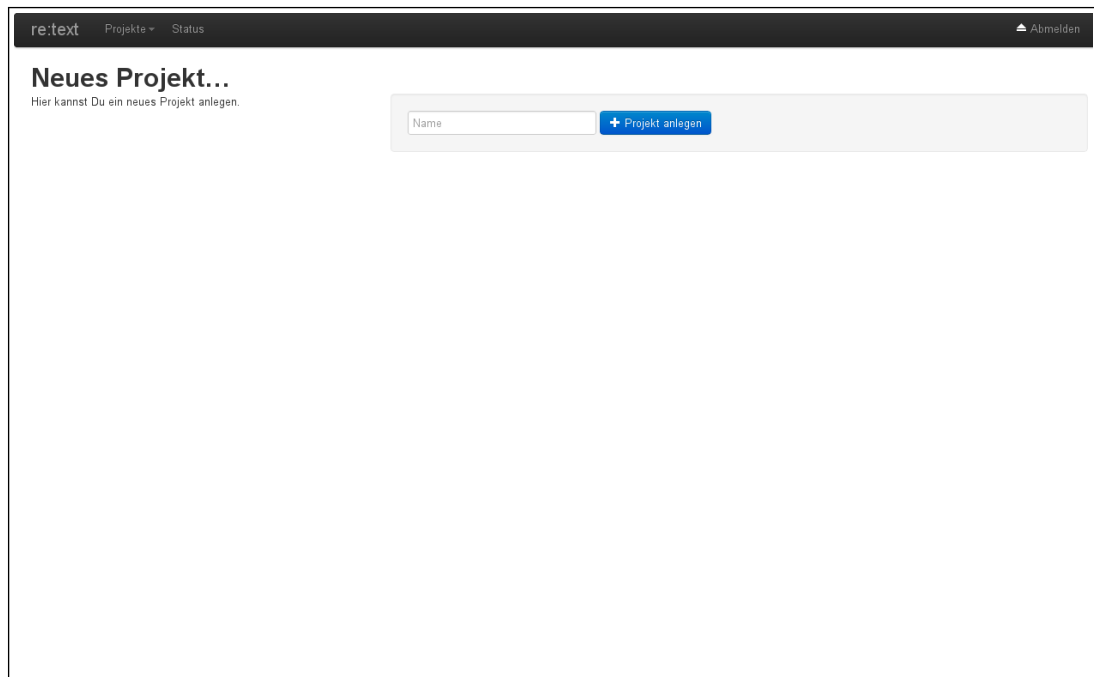


Abbildung 23: Projekt anlegen

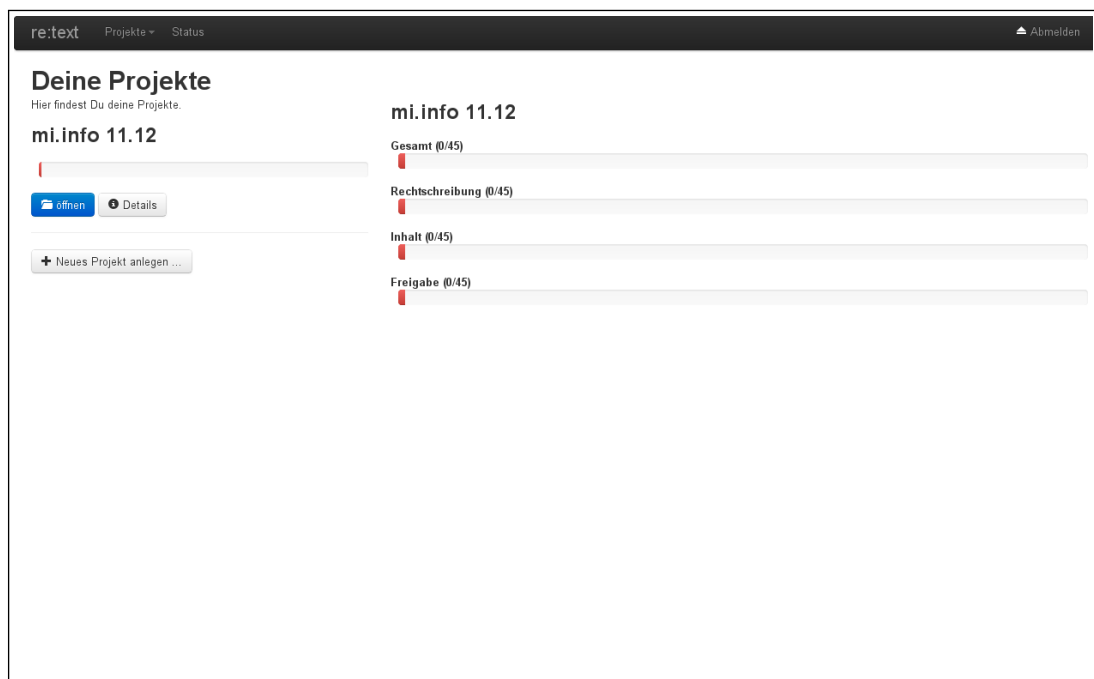


Abbildung 24: Projekt anzeigen, mit Darstellung des Projekt-Fortschritts

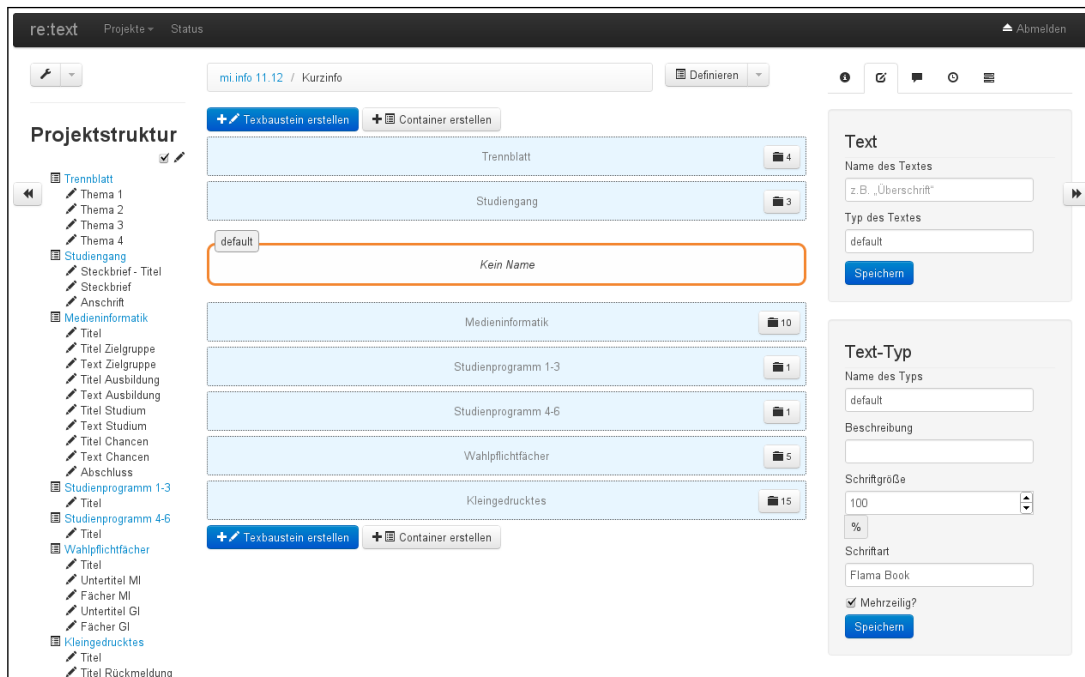


Abbildung 25: Produktstruktur definieren, vgl. 6.1.2 / S.44

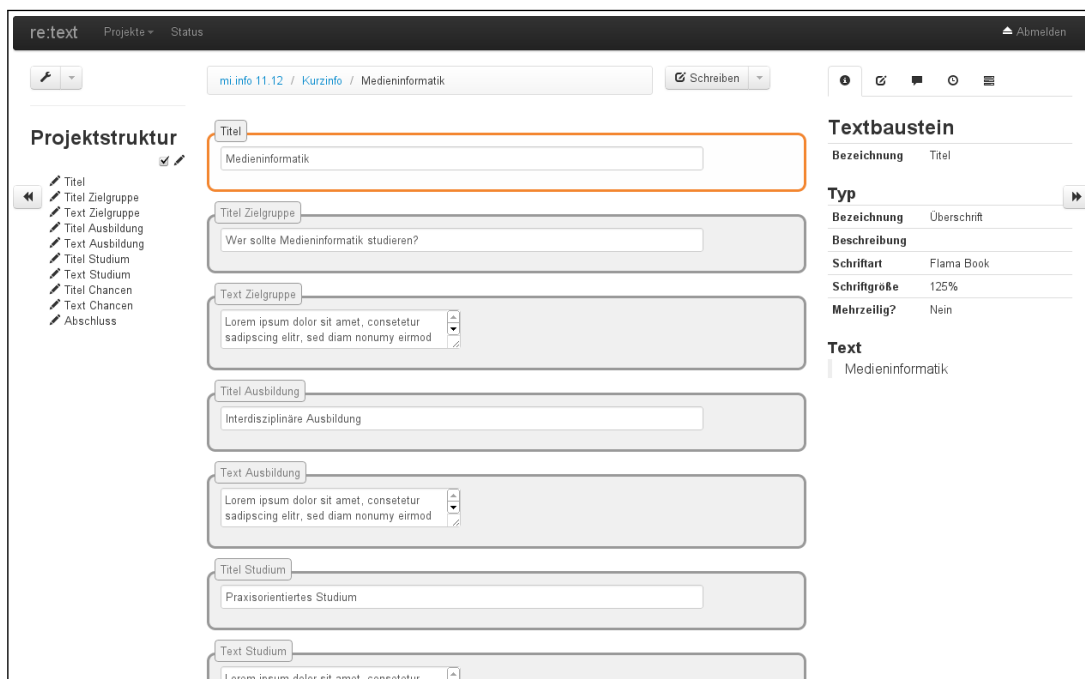


Abbildung 26: Texte erstellen, vgl. 6.1.3 / S.45

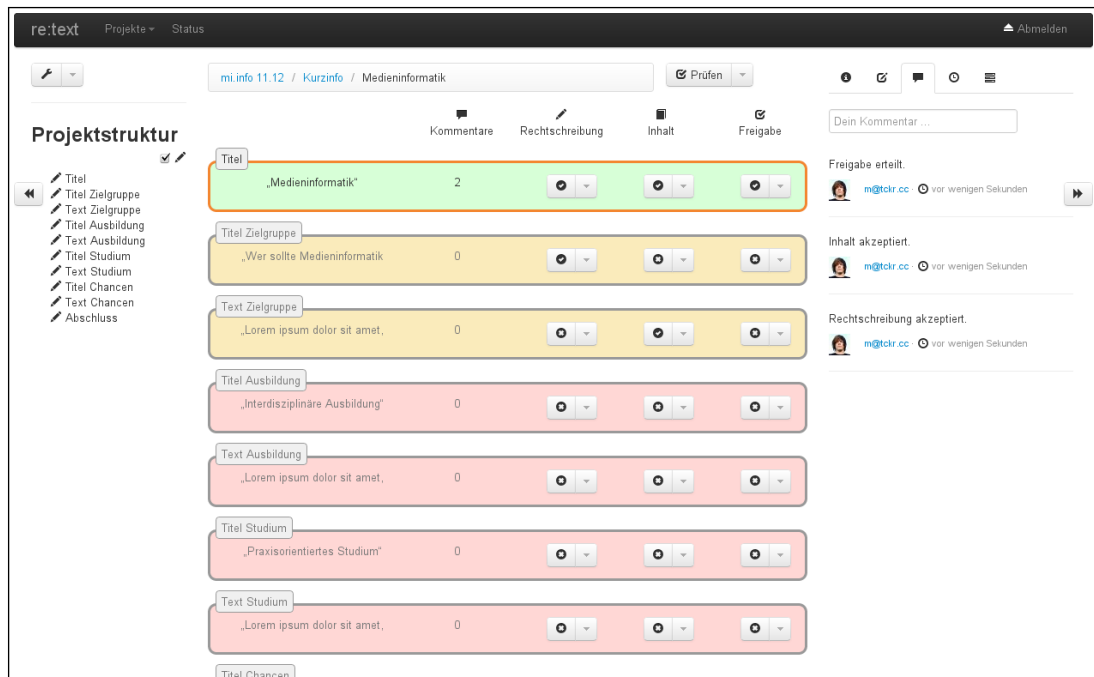


Abbildung 27: Qualitätssicherung, vgl. 6.1.5 / S.47

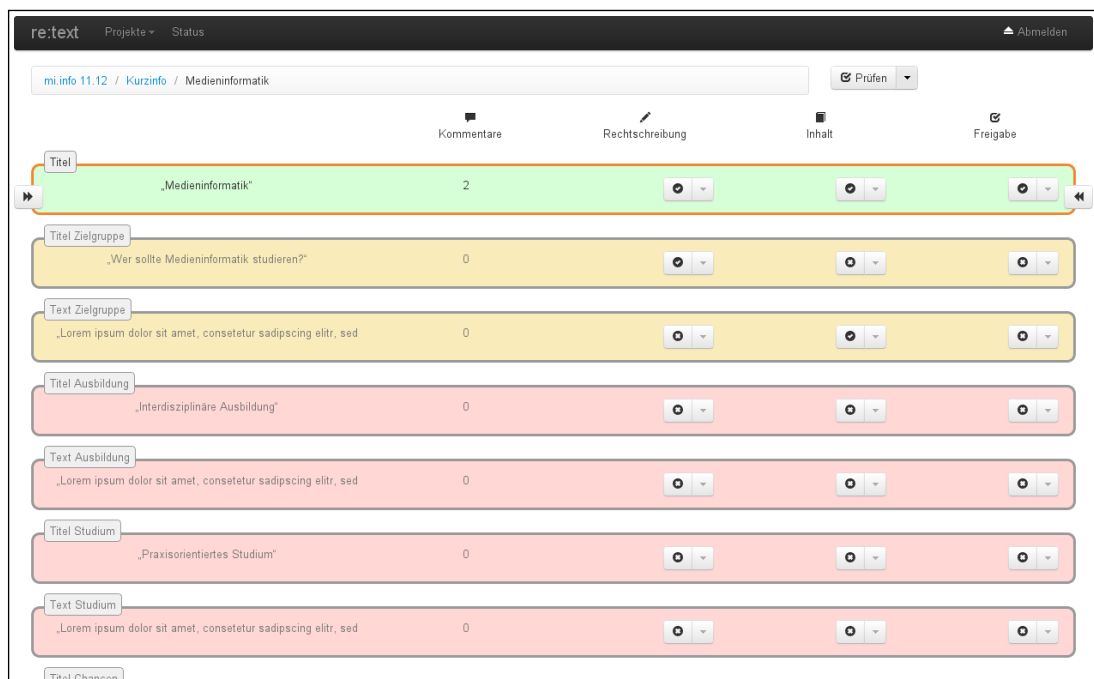


Abbildung 28: Darstellung im Vollbild-Modus mit ausgeblendeten Seitenleisten, vgl. 6.1.1 / S.43

B API-Endpunkte

B.1 Benutzer

Methode	Pfad	Beschreibung
POST	/user	Benutzer registrieren
POST	/login	Login
POST	/logout	Logout
GET	/auth	Sitzungsstatus abfragen
GET	/user/<id>	Benutzerprofil laden

B.2 Projekt

Methode	Pfad	Beschreibung
POST	/project	Neues Projekt anlegen
GET	/project	Liste mit Projekten des Nutzers laden
GET	/project/<id>	Projekt laden
GET	/project/<id>/progress	Fortschritt des Projektes laden

B.3 Elemente

Methode	Pfad	Beschreibung
GET	/element	Liste mit Elementen (Container, Text) unterhalb eines Elementes laden

B.4 Container

Methode	Pfad	Beschreibung
POST	/container	Container anlegen
GET	/container	Liste mit Containern unterhalb eines Elementes laden
GET	/container/<id>	Container laden
PUT	/container/<id>	Container aktualisieren
DELETE	/container/<id>	Container löschen
GET	/container/<id>/breadcrumb	Navigationspfad ab einem Container bis zum obersten Container laden
GET	/container/<id>/tree	Baumstruktur ab einem Container laden

B.5 Texte

Method	Pfad	Beschreibung
POST	/text	Text anlegen
GET	/text/<id>	Text laden
PUT	/text/<id>	Text aktualisieren
DELETE	/text/<id>	
POST	/text/<id>/comments	Kommentar zu einem Text anlegen
GET	/text/<id>/comments	Kommentare zu einem Text laden
GET	/text/<id>/history	Änderungshistorie zu einem Text laden

B.6 Text-Typen

Method	Pfad	Beschreibung
GET	/texttype	Liste mit Text-Typen eines Projektes laden
GET	/texttype/<id>	Text-Typ laden
PUT	/texttype/<id>	Text-Typ aktualisieren

B.7 Export

Method	Pfad	Beschreibung
GET	/export/contentbooklet.pdf	Content-Booklet eines Projektes als PDF exportieren
GET	/export/contentbooklet.html	Content-Booklet eines Projektes als HTML exportieren

Literatur

- [1] Adobe Systems Incorporated. Professionelle Software für Redaktionen | Adobe InCopy CS5.5. [Online; Stand 12. April 2012; <http://1.tckr.cc/HMQNo8>].
- [2] D.M. Brown. *Communicating design: developing web site documentation for design and planning*. Safari Books Online. Peachpit Press, 2007.
- [3] Sean Coates. PHP as a templating language. [Online; Stand 14. Mai 2012; <http://1.tckr.cc/KG5ZSR>].
- [4] M. Cohn. *User Stories Applied: For Agile Software Development*. The Addison-Wesley Signature Series. Addison-Wesley, 2004.
- [5] Microsoft Corporation. Zusammenarbeit an Dokumenten und gemeinsame Dokumenterstellung. [Online; Stand 17. April 2012; <http://1.tckr.cc/Ja5QX0>].

- [6] J. Dunkel, A. Eberhart, S. Fischer, C. Kleiner, and A. Koschel. *Systemarchitekturen für Verteilte Anwendungen: Client-Server, Multi-Tier, SOA, Event Driven Architectures, P2P, Grid, Web 2.0*. Hanser Fachbuchverlag, 2008.
- [7] S. Freeman and N. Pryce. *Growing Object-Oriented Software, Guided by Tests*. Addison-Wesley signature series. Addison Wesley, 2009.
- [8] Brett Hill. Check it out! easily enable document version control in sharepoint online. [Online; Stand 13. April 2012; <http://1.tckr.cc/IUMZQu>].
- [9] J.T. Howerton. Service-oriented architecture and web 2.0. *IT Professional*, 9(3):62–64, Mai-Juni 2007.
- [10] A. MacCaw. *JavaScript Web Applications*. O'Reilly Series. O'Reilly Media, 2011.
- [11] Jakob Nielsen. 10 heuristics for user interface design. [Online; Stand 10. Mai 2012; <http://1.tckr.cc/IWoaFd>].
- [12] Corporate Vice President of the Developer Division at Microsoft Corporation S. Somasegar. Key software development trends. [Online; Stand 3. Mai 2012; <http://1.tckr.cc/II1jNz>].
- [13] H. Schanze and S. Pütz. *Metzler Lexikon Medientheorie, Medienwissenschaft: Ansätze, Personen, Grundbegriffe*. Metzler, 2002.
- [14] WoodWing. Wie funktioniert das? | WoodWing.com. [Online; Stand 12. April 2012; <http://1.tckr.cc/HJMWTI>].
- [15] Workflow Management Coalition. *Terminology & Glossary*, 3.0 edition, Februar 1999.