

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 Kommunikationsmedienprodukte

vorgelegt von Markus Tacker  
am 2. Juli 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, 2. Juli 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, 2. Juli 2012

Markus Tacker

## Satz

Gesetzt mit  $\text{\LaTeX}$  / XeTeX

Schrift: Museo Sans<sup>1</sup> und Vollkorn<sup>2</sup>

## Diagramme

Erstellt mit Graphviz<sup>3</sup> und Google Docs<sup>4</sup>

---

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

<sup>2</sup>[http://friedrichalthausen.de/?page\\_id=411](http://friedrichalthausen.de/?page_id=411)

<sup>3</sup><http://www.graphviz.org/>

<sup>4</sup><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, 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.

# Inhaltsverzeichnis

<b>I</b>	<b>Einleitung</b>	<b>I</b>
<b>2</b>	<b>Definition</b>	<b>3</b>
<b>3</b>	<b>Problem-Analyse</b>	<b>5</b>
3.1	Die besondere Rolle von Text in Informations- und Kommunikationsmedien	6
3.2	Die Werkzeuge der Wahl zur Verwaltung von Text: Word und Excel . . . . .	8
3.3	Probleme bei der Verwendung von Textverarbeitungs- und Tabellenkalkulationsprogrammen im Verlauf eines Projekts . . . . .	12
3.4	Beispiele aus der Praxis . . . . .	18
3.4.1	Internetseite EnBW Transportnetze AG . . . . .	18
3.4.2	Banner-Kampagne Nintendo . . . . .	18
3.4.3	EA Phenomic: BattleForge . . . . .	19
3.4.4	MAN Truck & Bus AG: Neufahrzeug-Konfigurator . . . . .	19
3.5	Schlussfolgerung . . . . .	20
<b>4</b>	<b>Personas</b>	<b>21</b>
4.1	Eva, Konzepterin . . . . .	24
4.2	Lotte, Designerin . . . . .	25
4.3	Torsten, Texter . . . . .	26
4.4	Markus, Kunde . . . . .	27
4.5	Jorinde, Übersetzerin . . . . .	28
4.6	Jan, Produzent . . . . .	29
4.7	Arthur, Projektleiter . . . . .	30
<b>5</b>	<b>Konzeption einer an den spezifischen Workflow angepassten Anwendung</b>	<b>31</b>
5.1	Abzubildender Workflow . . . . .	32
5.1.1	Inhalt . . . . .	33
5.1.2	Inhaltsvorgaben . . . . .	34
5.1.3	Attribute . . . . .	34
5.1.4	Status . . . . .	35
5.2	Anforderungen an die Anwendung . . . . .	38

5.3	Komponenten der Anwendung . . . . .	39
5.3.1	Browserbasiertes GUI . . . . .	39
5.3.2	Schnittstellen . . . . .	40
5.3.3	Anwendungsserver . . . . .	41
5.4	Zugang . . . . .	43
5.5	Überblick über den Aufbau der gesamten Anwendung . . . . .	45
5.6	Zusammenfassung, Nachteile & Risiken des Konzepts . . . . .	46
<b>6</b>	<b>Entwurf des zentralen Systems</b>	<b>49</b>
6.1	Komponenten . . . . .	50
6.2	Domänenmodell . . . . .	52
6.3	Entwurf eines browserbasierten GUI . . . . .	55
6.3.1	Aufbau des GUIs . . . . .	57
6.3.2	Definieren des Produkts . . . . .	59
6.3.3	Texte erstellen . . . . .	61
6.3.4	Texte übersetzen . . . . .	62
6.3.5	Prüfen . . . . .	63
6.3.6	Kontext-Informationen . . . . .	64
6.4	Anbindung des GUI an den Anwendungsserver . . . . .	66
6.5	Implementierung des GUI . . . . .	68
6.5.1	Komponenten . . . . .	68
6.5.2	Klassendiagramm . . . . .	70
6.6	Entwurf des Anwendungsservers . . . . .	72
6.6.1	API, Cronjobs, CLI, ... . . . .	72
6.6.2	Dependency-Injection-Container . . . . .	72
6.6.3	Controller . . . . .	72
6.6.4	Models . . . . .	72
6.6.5	Persistenz, ORM . . . . .	73
6.6.6	Import/Export & Benachrichtigung . . . . .	73
6.6.7	Jobs . . . . .	74
6.6.8	Workflow . . . . .	74
6.6.9	Service-Adapter . . . . .	74

6.7	Implementierung des Anwendungsservers . . . . .	75
6.7.1	Kern-System . . . . .	75
6.7.2	API . . . . .	75
6.7.3	Persistenz . . . . .	77
6.7.4	Klassendiagramm . . . . .	78
6.8	Implementierter Workflow am Beispiel des Studiengangsflyers . . . . .	80
6.9	Zusammenfassung . . . . .	82
<b>7</b>	<b>Fazit</b>	<b>83</b>
<b>A</b>	<b>Anhang</b>	<b>I</b>
A.1	Screenshots . . . . .	I
A.2	API-Endpunkte . . . . .	V
A.2.1	Benutzer . . . . .	V
A.2.2	Projekt . . . . .	V
A.2.3	Elemente . . . . .	V
A.2.4	Container . . . . .	V
A.2.5	Texte . . . . .	VI
A.2.6	Text-Typen . . . . .	VI
A.2.7	Export . . . . .	VI
A.3	Content-Booklet . . . . .	VII

## Abbildungsverzeichnis

1	Bei der Erstellung von Texten beteiligte Personen . . . . .	5
2	Word-Dokument mit Texten für eine Internetseite . . . . .	9
3	Excel-Dokument mit Texten für eine Internetseite . . . . .	10
4	Übersicht über die Personas und den idealisierten Workflow . . . . .	23
5	Einfluss auf den Status eines Textbausteins . . . . .	36
6	Aufbau der Anwendung in stark vereinfachter Darstellung . . . . .	39
7	Aufbau der gesamten Anwendung im Überblick . . . . .	45
8	Komponenten der Anwendung . . . . .	50
9	Domänenmodell . . . . .	52
10	GUI-Klassendiagramm des Prototyps . . . . .	70
11	I/O-Komponente . . . . .	73
12	Persistierte Objekte im Prototyp . . . . .	77

13	Anwendungsserver-Klassendiagramm des Prototyps . . . . .	78
14	Auszug aus dem Content-Booklet des Studiengangsflyers . . . . .	VII

## Tabellenverzeichnis

1	Kriterien von Textbausteinen und verantwortliche Personen . . . . .	7
2	Interviewte Personen . . . . .	22
3	Stärke des Einflusses, den Mitarbeiter in einem Projekt haben . . . . .	33
4	Stärke des Einflusses auf den Status eines Textbausteins . . . . .	37
5	Umfang und Häufigkeit der Benutzung des browserbasierten GUIs bezogen auf die durchgeführte Operation . . . . .	56

## Verzeichnis der Wireframes

1	Aufbau des browserbasierten GUIs . . . . .	57
2	Bearbeiten der Produktstruktur . . . . .	59
3	Erstellen der Texte . . . . .	61
4	Übersetzen der Texte . . . . .	62
5	Überprüfen der Texte . . . . .	63
6	Kontext-Informationen . . . . .	64

## Verzeichnis der Screenshots

1	Browserbasiertes GUI des Prototyps . . . . .	69
2	Registrierung für neue Benutzer . . . . .	I
3	Login . . . . .	I
4	Projekt anlegen . . . . .	II
5	Projekt anzeigen, mit Darstellung des Projekt-Fortschritts . . . . .	II
6	Produktstruktur definieren . . . . .	III
7	Texte erstellen . . . . .	III
8	Qualitätssicherung . . . . .	IV
9	Darstellung im Vollbild-Modus mit ausgeblendeten Seitenleisten . . . . .	IV





# 1 Einleitung

Die Herstellung von Informations- und Kommunikationsmedienprodukte ist heute ein zentraler Wirtschaftsbereich in Deutschland. Zahlreiche Unternehmen und Abteilungen haben es sich zur Aufgabe gemacht, Produkte zu erschaffen, die Werbebotschaften transportieren oder Information zugänglich machen. Die technischen Möglichkeiten sind vielfältig wie nie. Videos in High-Definition auf Tablet-PCs über mobile Datenverbindungen zu streamen gehört inzwischen zum Alltag. Trotz allen Fortschritts können auch die modernsten Kommunikationsformen auf einen klassischen Bestandteil nicht verzichten: *Text*. Denn gerade die Medienvielfalt mit der Möglichkeit durch Multimedia-Inhalte, Interaktion und Personalisierung 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 sprichwörtliche »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 eines Mediums, der im Verlauf dessen Erstellung von den meisten Personen beeinflusst wird. Da die Erstellung von Medienprodukten 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 Produkts. Die stetig zunehmende Komplexität der Medienprodukte und der Wunsch nach Kostenreduktion trägt ihren Teil dazu bei, dass der Kreis derer, die an der Erstellung eines Medienprodukts beteiligt sind, weiter wächst: Spezialisten sind für bestimmte Aspekte des Produkts zuständig und verstärken bei Bedarf als freie Mitarbeiter das Team. Der Austausch zwischen allen Parteien über diese Anforderungen und die durch die Texte zu vermittelnden Inhalte hat sich trotz aller technischen Fortschritte in den letzten 15 Jahren nicht verändert: er 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.

Zu diesem Zweck wird nach der Definition wichtiger Begriffe in Kapitel 2 · S.3 zunächst in Kapitel 3 · S.5 analysiert, welche Probleme in branchenüblichen Projektverläufen auftreten. Aufbauend auf dieser Analyse und unter Zuhilfenahme von Personas, die in Kapitel 4 · S.21 vorgestellt werden, wird in Kapitel 5 · S.31 eine Lösung konzipiert, die versucht, die genannten Probleme zu beseitigen und den Anforderungen der Personas zu genügen. Für die wichtigsten Bestandteile der Lösung, den Anwendungsserver und das browserbasierte GUI, wird in Kapitel 6 · S.49 die konkrete Architektur entworfen und detaillierte Gestaltungsrichtlinien mithilfe von Wireframes festgelegt. Dieser Entwurf wird schließlich in

Form eines Prototyps umgesetzt, mit dessen Hilfe überprüft wird, inwieweit sich das vorgeschlagene Konzept praktisch einsetzen lässt und die entworfene Systemarchitektur geeignet ist, die benötigten Funktionen abzubilden.

Die Thesis schließt mit dem Fazit in Kapitel 7 · S.83.

## 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)** Allgemein lässt sich sagen, dass ein Workflow aus den zum Erreichen eines Zieles nötigen Arbeitsschritten besteht. [9, S.8] definiert einen Workflow als 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.

**Text (Textbaustein)** Damit sind die kleinsten sinnvoll identifizierbaren Bestandteile gemeint, aus denen sich der Text eines Produkts 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 (Medienprodukt, 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. [8, S.199–201]). Dies können z.B. Marketingmedien, wie Broschüren oder Fernsehwerbespots, aber auch softwarebasierte Produkte, wie eine Smartphone-Anwendung oder eine Internetseite, sein.

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

**Projekt** Die Erstellung von Medien erfolgt in Agenturen in Projektarbeit. Projekte sind einmalig, zeitlich begrenzt und vereinen zielgerichtet die zur Erstellung des Produkts 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.

**Produzent** Eine Person, die Medienprodukte erstellt.

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

**Wireframes** Wireframes dienen zur schnellen Skizzierung eines Produkts mit Hilfe von einfachen geometrischen Formen und Texten. Sie werden angefertigt, um einen Überblick über den gesamten Umfang eines Produkts zu bekommen und 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 grundlegend von anderen Ansichten unterscheidet.

### 3 Problem-Analyse

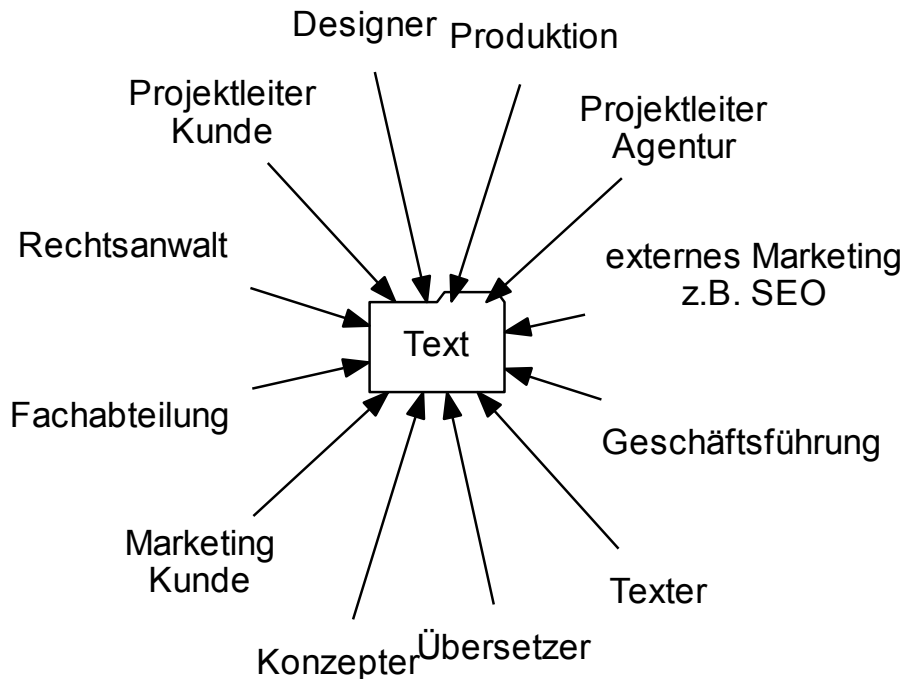


Abbildung 1: Bei der Erstellung von Texten beteiligte Personen

In diesem Kapitel werden die Probleme beschrieben, die bei der Erstellung von Informations- und Kommunikationsmedienprodukten in Zusammenhang mit den beinhalteten Texten auftreten. Die Abschnitte 3.1 und 3.2 analysieren die besondere Rolle von Text und die verwendeten Werkzeuge für dessen Verwaltung, anschließend zeigt Abschnitt 3.3 · S.12 typische Probleme auf, die im Verlauf von Projekten zu dessen Erstellung entstehen. Abschnitt 3.4 · S.18 belegt dies mit Beispielen aus der Praxis.

Die Analyse des Problems basiert auf im April 2012 geführten Interviews mit Personen, die in ihrem Arbeitsalltag regelmäßig mit Texten zu tun haben. Eine Liste der interviewten Personen findet sich in Tabelle 2 · S.22.

### 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, existieren Medien, die sich über Text als Hauptbestandteil definieren. 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<sup>5</sup>, über Werbemittel wie Newsletter bis zur Unternehmenswebsite – die Möglichkeit Inhalte hierarchisch zu strukturieren und sogar über eine Suche zugänglich zu machen hebt eine medienbedingte Limitierung des Umfangs, wie bei Druckprodukten, praktisch auf.

Betrachtet man Projekte, 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 und da Text sehr viele verschiedenen Kriterien erfüllen muss, sind an der Erstellung der Texte für das Medium mehr Personen beteiligt, als es z.B. für die Gestaltung, die Auswahl von Bildmaterial oder für die Programmierung der Fall ist. Tabelle 1 · S.7 listet exemplarisch eine typische Gruppe von Personen auf, die im Verlauf des Projekts Einfluss auf den Text eines Produkts hat. Dieser Einfluss wird dabei in der Regel nicht in optimaler Reihenfolge und im Sinne des geplanten Projektverlaufs 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 z.B. von Anwälten sollten im besten Fall vor einer Übersetzung vorliegen, diese richten sich aber vorrangig nach ihren eigenen Terminplänen. Auch die Kriterien durch die Text beeinflusst wird, sind sehr vielfältig: Im Entwurf und in der Umsetzung der Produkte legen Designer, Informations-Architekten und Produzenten die Struktur von Text wie Art der Ansprache, maximale Wortlänge, Anzahl der Wörter einer Überschrift fest. Diese werden durch das verwendete Medium vorge-

---

<sup>5</sup>Internetseite mit sehr kompaktem Inhaltsumfang für einen speziellen Zweck (z.B. Produkt, Gewinnspiel) oder einen besonderen Aktionszeitraum

<b>Kriterium</b>	<b>Art</b>	<b>Verantwortlich</b>
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

geben. 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.

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 Kundenseite bekannt sind. Dieses Expertenwissen kann in der Regel nicht innerhalb der meist kurzen Projektlaufzeit an die umsetzende Agentur vermittelt werden. Es ist also unvermeidlich, dass Text während des gesamten Projektverlaufs geändert werden kann. Neben den Einflüssen durch Experten gibt es auch projektbedingt meist kurzfristige Einflüsse auf Text. Sind in Texten Informationen enthalten, die einen zeitlichen Aspekt abbilden, ergeben sich durch Verzögerungen im Projekt automatisch Änderungsanforderungen. Ein Beispiel sind Gewinnspiele: Verschiebt sich durch Probleme während des Projekts 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 Projekts 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, das vom ursprünglichen Text im Quelldokument bis zur Darstellung im fertigen Produkt viele aufwändige Arbeitsschritte nötig sein können.

### 3.2 Die Werkzeuge der Wahl zur Verwaltung von Text: Word und Excel

Zur Abbildung der komplexen Abläufe bei der Erstellung von Informations- und Kommunikationsmedienprodukten 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“ [10] 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“ [16] – 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.12 eingegangen. Zuerst muss jedoch untersucht werden, warum statt spezieller Werkzeuge, die für den komplizierten Workflow 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** 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 auch Zusatzinformationen, wie die Klasse des Texts (Überschrift, Fließtext, Schaltfläche), dort zuzuordnen. Abbildung 2 · S.9 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 im Quellcode der Website verwendet werden, die dann an deren Stelle durch entsprechende Methoden einer Template-Engine eingefügt werden. Über die Formatierung der Überschriften im Dokument wird die Hierarchie der Texte definiert.

Der Einsatz von Tabellen statt Dokumenten ist eine weitere Möglichkeit die verwendeten Texte zu erfassen. Abbildung 3 · S.10 zeigt beispielhaft ein Excel-Dokument, in dem pro Zeile ein Text definiert wird. In den Spalten finden sich neben dem eigentlichen Text Zusatzinformationen wie z.B. die Text-Klasse. Tabellarische Dokumente werden oft bei umfangreichen Projekten oder dann verwendet, wenn mehrere Sprachversionen verwaltet werden müssen.

**Rechtschreibkorrektur** In Textverarbeitungsprogrammen sind ausgefeilte Funktionen zur Rechtschreibkorrektur enthalten, die bereits während der Eingabe auf Fehler aufmerk-



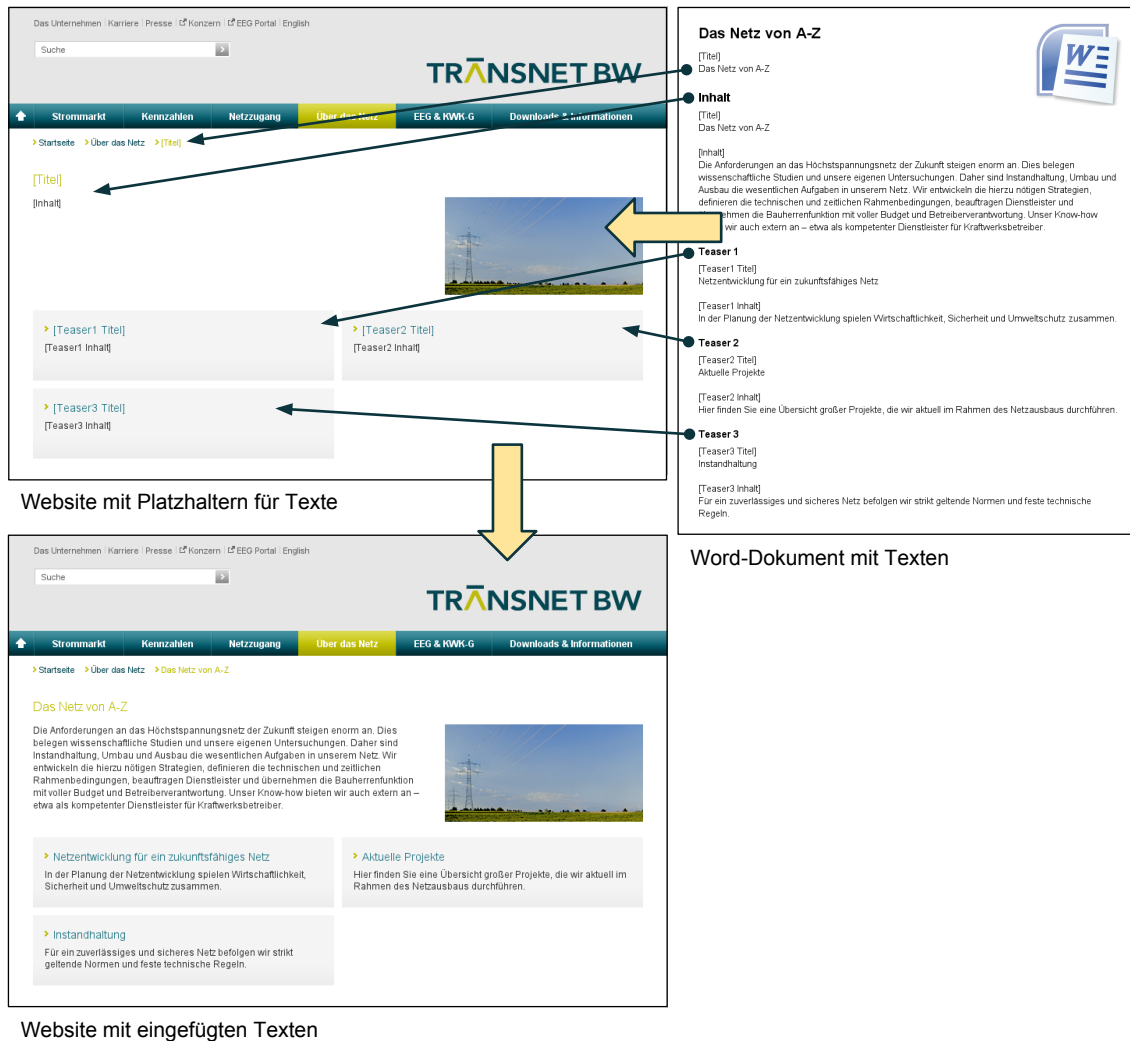


Abbildung 2: Word-Dokument mit Texten für eine Internetseite

sam machen und für viele Sprachen verfügbar sind. So ist sichergestellt, dass bereits die erste Version eines Texts 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 oder für Hinweise zu dessen 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 mehre-

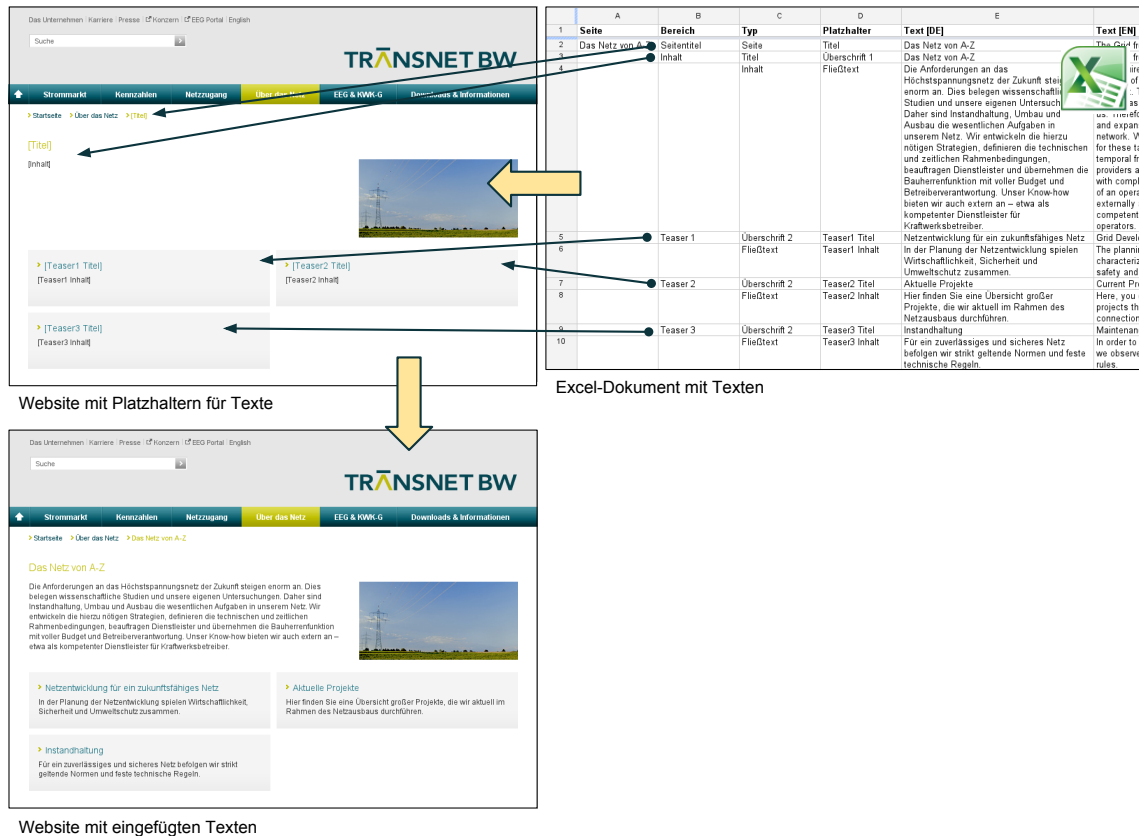


Abbildung 3: Excel-Dokument mit Texten für eine Internetseite

re Versionen eines Dokuments 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 automatisiert anzulegen. Dies hilft bei größeren Projekten einen Überblick über den Aufbau des Produkts 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 wird vor allem PDF bei Word-Dokumenten zur Abstimmung, unter anderem auch deswegen, weil es in Adobe Acrobat umfangreiche Korrekturfunktion-

nen gibt. Bei Excel-Dokumenten wird vor allem CSV verwendet, da Texte in diesem Format leicht in andere Systeme importiert werden können.

**Formatierungsfunktionen** Umfangreiche Formatierungsfunktion erlauben es Texte besonders auszuzeichnen. Oft werden durch farbige Markierung Passagen hervorgehoben, die entfallen oder inhaltlich überarbeitet werden müssen. Auch können Formatierungen so angelegt sein, wie sie im Produkt erscheinen sollen – in der Regel werden dann Teile des Texts oder einzelne Wörter 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 praktisch bei allen Projektbeteiligten eine Installation von Microsoft Office vorausgesetzt werden kann, werden Word und Excel zu »leichtgewichtigen« Werkzeugen, die dem 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. 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. 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 auf den ersten Blick für den Einsatz zur Verwaltung von Texten für Medienprodukte geeignet. Dies erklärt, warum sie zu Beginn eines Projekts als geeignet angesehen und in Agenturen immer wieder als Werkzeug für die Erfassung, Definition und Übersetzung der Texte eines Projekts 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 Projekts

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 von Informations- und Kommunikationsmedienprodukten. In den für diese Bachelor-Thesis geführten Interviews haben jedoch alle Personen von vielfältigen Problemen im Zusammenhang mit diesen Werkzeugen berichtet. Dies belegt zum einen, dass im gängigen Workflow viele Möglichkeiten zur Verbesserung existieren und liefert zum anderen 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 ein 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 indem die Dokumenten-Datei entweder per E-Mail jeweils zum nächsten Bearbeiter verschickt wird oder die Datei auf einen, durch alle Beteiligten erreichbaren, Speicherort verschoben wird. In Agenturen handelt es sich hierbei meistens um eine Netzwerklaufwerk – hierauf haben aber nur Mitarbeiter Zugriff, die Zugang zum lokalen Netzwerk der Agentur haben. Soll die Datei auch externen Mitarbeitern oder dem Kunden zur Verfügung gestellt werden, muss diese per E-Mail verschickt oder in extern erreichbare Speicherorte kopiert werden, wie z.B. FTP-Server, Wikis und Extranet-Portale. Üblich ist auch der Einsatz spezieller Programme zum Dateiaustausch, 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. dem Kunden zur Abnahme geschickt wurden. Dann kommt es dazu, dass mehrere Versionen des Dokuments parallel existieren: eine Version beim Kunden, die dort mit Änderungen und Ergänzungen versehen wird und eine Version in der Agentur, die sich aufgrund von Änderungen im Verlauf des Projekts ändert. Um anschließend alle Beteiligten auf den aktuellen Stand zu bringen, müssen die verschiedenen Versionen des Dokuments 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 den 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, die leicht zu verwechseln sind. 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 Texte nicht kennen, was ebenfalls irrtümliche Änderungen an den Texten durch fehlerhaftes Copy&Paste begünstigt. Auch ist die Ein-

deutigkeit 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 sie z.B. nicht ausreichend informiert sind, oder kommt es zu gleichzeitiger Änderungen durch zwei Personen kann es so auch zu zwei verschiedenen Dateien mit dem gleichen Dateinamen kommen. Durch die Verteilung der Dokumente auf verschiedene Speicherorte kommt es zu Situationen, in denen nicht klar ist, wer aktuell die *neueste* Dateiversion hat.

Tatsächlich existiert mit Microsoft SharePoint, einer Software für Intra-, Extra- und Internetportale, eine Lösung, die dieses Probleme behebt [13]. Mithilfe von Shared Documents lassen sich Dokumente 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 durch den Bearbeiter wieder »eingchecked«. 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 gleichzeitig 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. Zudem setzen die Funktionen zum gemeinsamen Bearbeiten von Word- oder Excel-Dokumenten voraus, dass alle Mitarbeiter über die neueste Programm-Version verfügen [12]. Dies kann aber nicht bei allen Projektbeteiligten, besonders nicht auf Kundenseite, vorausgesetzt werden.

**Monolithische Dokumente** Das Zusammenführen aller Textbausteine eines Produkts in einem Dokument hat den Nachteil, dass dieses nur als Ganzes weitergeben werden kann. In bestimmten Konstellationen muss aber sichergestellt werden, dass bestimmte Inhalte nicht von allen Projektbeteiligten einsehbar sind. Zum einen kann es sich dabei um Informationen handeln, die sensibel sind oder der Geheimhaltung unterliegen, 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, verschieden Versionen des Dokuments anzulegen, die an den jeweiligen Personenkreis angepasst sind. Dies erzeugt die Probleme, 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 ihrem Unternehmen an die zuständige Person weiterleiten, auf deren Antwort warten und dann die Antwort zurück spielen. Hierdurch bilden sich umfangreiche und langlebige E-Mail-Kommunikationsketten, an denen viele, meistens zu viele, Personen beteiligt sind, die in ungeordneter Reihenfolge Feedback liefern.

**Strukturierung von Dokumenten** Einer der Gründe, warum Word und Excel zur Standardausstattung auf jedem Büro-Computer gehören, ist, 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 Produkts in einem Text- oder Tabellen-Dokument anzulegen. Hierbei wird meistens eine hierarchische Struktur mithilfe von Abschnitten angelegt (vgl. Abb. 2 · S.9). In Excel wird im Hinblick auf die kompaktere Darstellung meistens auf besondere Formatierungen verzichtet und mit sich wiederholenden Zellen gearbeitet, die eine hierarchische Struktur simulieren – die zweidimensionale Tabellendarstellung ist für komplexere Hierarchien nicht ausgelegt (vgl. Abb. 3 · S.10). Diese aufwändige Strukturierung des Dokuments muss auch geschehen, damit sich alle Anwender im Dokument zurecht finden können und eine eindeutige Zuordnung zwischen Textbausteinen im Dokument und den dafür vorgesehenen 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, um zu vermeiden, dass es im späteren Verlauf des Projekts durch nicht berücksichtigte Fälle 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 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 anlegt, 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 Texts nicht mehr zugeordnet werden kann und im Produkt fehlt.

Werden Tabellen in Word-Dokumenten verwendet um z.B. tabellarische Inhalte in einem Produkt zu beschreiben, führt das 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 erschwert.

Üblich ist auch das Einfügen von Bildern. Dies ist nötig, um die Zuordnung der Texte zu erleichtern oder um Untertitel für Fotos im Produkt zu definieren. Hierbei treten dann zusätzliche Formatierungsprobleme auf, da das Platzieren von Bildern in Office nur in beschränktem Maße beeinflusst werden kann.

Problematisch ist auch der Austausch der Formatierungen zwischen verschiedenen Programm-Versionen von Word und 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 Tabellenkalkulationsprogrammen 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 jedoch gerade beim Übertragen dieser Formatierungen Unsauberkeiten.

Soll die Formatierungen in der Produktion dann schließlich in das Produkt übernommen werden, muss dies manuell geschehen, da sich diese Auszeichnungen nicht automatisch in die jeweiligen Produktionswerkzeuge übertragen lassen, weil dafür z.B. keine Konverter existieren oder die Formatierungsregeln nicht 1:1 übernommen werden können. Das manuelle Anlegen der Formatierungen im Produkt kann, ähnlich wie bei Copy&Paste, zu Übertragungsfehlern führen.

**Notizen und Anmerkungen** Neben den eigentlichen Textbausteinen werden in Word- und Excel-Dokumenten auch Hinweise und Kommentare zu den Texten mithilfe von Notizen oder direkt in das Dokument platzierten, besonders formatierten, Texten hinterlegt. Werden Notizen verwendet, besteht 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 dies dazu führen, dass diese Hinweise übersehen werden oder bei 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 mithilfe von E-Mails in denen die Anmerkungen bzw. Änderungswünsche aufgezählt werden. Es muss dabei dann sichergestellt werden, dass die entsprechenden Personen über diese Informationen verfügen und auch informiert sind, welche gültig sind und welche veraltet.

Bei der Übersetzung von Texten kommt es mitunter vor, dass Hinweistexte, die 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 alle wichtigen Zusatzinformationen zu Texten, wie z.B. Angaben über maximale Satzlängen, da sie die fertigen, abgenommenen Texte in der Ausgangssprache in einem separaten 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 Dokument-Versionen identifiziert werden müssen, wenn es nicht sinnvoll ist, alle Texte im Produkt 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. Es existieren jedoch noch weitere Probleme, die sich ebenfalls negativ auf die Arbeitsgeschwindigkeit auswirken. Zu den wichtigsten Usability-Problemen 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. Hierzu müssen das Originaldokument und die Übersetzung nebeneinander in zwei Fenstern geöffnet werden. Zum Vergleichen der Texte muss dann abwechselnd in diesen beiden Fenstern gescrollt werden. Kleinere Usability-Probleme sind etwa, dass das Anzeigen der Zeichenanzahl in einem Satz nur umständlich via Kontextmenü zu erreichen ist, diese Funktion aber häufig verwendet wird.

Typografische Details wie optionale Wort-Trennungen oder optionale Zeilenumbrüche sind in Word zwar möglich, können aber, wie die Formatierungen, nicht automatisch in das Endprodukt übernommen werden. Da Umbrüche sprachspezifischen Regeln folgen, müssen diese Informationen bereits während der Übersetzung hinterlegt werden, später in der Produktion werden sonst im Zweifelsfall Umbrüche willkürlich gesetzt.

Ein Problem technischer Art ist, dass sich bei großen Dokumenten mit oberen zweistelligen Seitenzahlen die Reaktionszeit der Anwendung merkbar verringert. Aufgrund unterschiedlicher Programm-Versionen, aber auch unterschiedlicher Betriebssysteme, kommt es beim Austausch der Dateien darüber hinaus zu verschiedenen Kompatibilitätsproblemen, da in Agenturen meistens Mac OS verwendet wird, auf Kundenseite jedoch Windows verbreitet ist.



**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 Vereinbarungen 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 Projekts immer wieder Schleifen entstehen, die Mehrarbeit erzeugen.

---

Dass Word und Excel trotz der genannten Probleme immer wieder zum Einsatz kommen, zeigt der nächste Abschnitt mit einer Reihe von Praxisbeispielen.

## 3.4 Beispiele aus der Praxis

Im Rahmen der für diese Bachelor-Thesis geführten Interviews mit Branchenexperten wurden die nachfolgenden Praxisbeispiele zusammengestellt. Diese zeigen für die in diesem Kapitel genannten Probleme Beispiele aus realen Projekten.

### 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<sup>6</sup> erarbeitet. Hierzu wurden die bestehenden Inhalte, die sich auf etwa 100 Seiten der Internetpräsenz 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.9), die dann dem Kunden zur Abstimmung via E-Mail zur Verfügung gestellt wurden. Da die Freigabe des Booklets nicht auf einmal sondern nur Abschnittsweise erfolgte, musste 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 waren der zusätzliche Arbeitsaufwand, um die Word-Dokumente zu erzeugen und der Zeitaufwand 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 Werbebannern Texte mit nur wenigen Worten pro Zeile gibt. Diese Zusatzinformationen werden aber oft nicht vom Übersetzer hinterlegt.

---

<sup>6</sup><http://enbw-transportnetze.de/>

### 3.4.3 EA Phenomic: BattleForge

Mehrere tausend Texte und deren Übersetzung für das auf Spielkarten basierende Echtzeit-strategiespiel BattleForge von EA Phenomic wurde bei diesem Projekt mithilfe einer Excel-Tabelle verwaltet. Hierbei wurden in der ersten Spalte Identifier für alle Texte vergeben. In den weiteren Spalten wurden die Übersetzungen eingetragen. Übersetzungsbüros haben die Texte mithilfe einer Kopie der Datei übersetzt. Praktikanten haben die übersetzten Texte dann wieder in die Master-Datei eingepflegt. Hierbei kam es regelmäßig zu Fehlern, zum einen wurden Zeilen und damit Identifier gelöscht und zum anderen wurden mehrzeilige Texte manchmal versehentlich in mehrere Zeilen 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 ggf. Einfluss auf die Texte im Spiel und deren 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 CMS verwaltet. Müssen neue Texte eingepflegt oder aktualisiert werden erhalten die Landesniederlassungen ein Delta als Excel-Dokument, das in der Regel alle vier Monate jeweils 1.000 bis 1.500 geänderte 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, einfügen. 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 Kommunikationsmedienprodukten spielt. Es wurde gezeigt, dass Microsoft Word und Excel ausgewählt werden, um die Texte für diese Produkte zu Verwalten und den vielen Projektbeteiligten zugänglich zu machen. Der Grund warum diese Programme für die Erfassung, Beschreibung, Erstellung, Übersetzung und Übertragung von Texten in das Produkt verwendet werden ist der, dass keine dedizierten Lösungen existieren, die explizit die genannten Abläufe abbilden. Statt dessen wird Software verwendet, die bei allen Beteiligten vorhanden ist und mit welcher diese bereits vertraut sind, wodurch sie mit den Texten in einem gewohnten Umfeld arbeiten können. Die Verwendung von Dateien ermöglicht den Austausch unter den Beteiligten im Netzwerk, per E-Mail oder über Programme. 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 kommt oder es wegen der schlechten Eignung der gewählten Software zu einer langsamen Arbeitsgeschwindigkeit führt. Die Probleme dieses Workflows werden jedoch erst im Verlauf des Projekts sichtbar und betreffen vor allem Agenturen, die als Dienstleister in einem Abhängigkeitsverhältnis stehen. Auf deren Seite werden die Missstände durch Mehrarbeit und aufwändige, sich wiederholende Arbeitsschritte ausgeglichen, die aufgrund ihrer Natur fehleranfällig sind.

---

Dieses Kapitel zeigt, dass vorhandene Werkzeuge entweder ungeeignet sind oder mangels Akzeptanz oder Komplexität nicht eingesetzt werden. Ziel dieser Bachelor-Thesis ist es, eine Anwendung zu konzipieren, die die genannten Probleme beseitigt. Für die Akzeptanz einer solchen Anwendung ist es besonders wichtig, dass sich diese stark an den Bedürfnissen ihrer Benutzer orientiert und für die unterschiedlichen Nutzungsarten und Erfahrung der Benutzer im Umgang mit Softwaresystemen den passenden Zugang bietet. Als Grundlage für die Konzeption der Anwendung in Kapitel 5 · S.31 werden im nächsten Kapitel Personas vorgestellt, die die typischen Nutzer dieser Anwendung repräsentieren.

## 4 Personas

Im vorigen Kapitel wurden die Probleme geschildert, die in Projekten auftreten, in denen Informations- und Kommunikationsmedienprodukte erstellt werden. Diese Analyse liefert wichtige Information für die Konzeption einer Anwendung in Kapitel 5 · S.31, die die genannten Probleme behebt. In diesem Kapitel werden Personas vorgestellt, die die typischen Benutzergruppen dieser Anwendung repräsentieren und deren Aufgaben und Erwartungen zusammenfasst.

*„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.“ [1, S.36 ff.]*

Personas sind ein wichtiger Baustein für die Konzeption einer Anwendung. 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. [2, S.38 ff.]

Die Personas basieren auf im April 2012 geführten Interviews mit den in Tabelle 2 · S.22 aufgezählten Personen. Die Namen und Fotos der Personas basieren zwar auf den interviewten Personen, dienen aber lediglich dazu, das Merken der Personas zu erleichtern.

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.23 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 Projekts, hat aber keinen Einfluss auf den Text. Er darf jedoch als wichtiges Bindeglied zwischen allen Beteiligten nicht fehlen.

---

**Projektleiter**

---

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

---

**Konzept, Design**

---

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

---

**Produktion**

---

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

---

**Texter**

---

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

---

**Übersetzer**

---

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

---

**Kunde**

---

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

Tabelle 2: Interviewte Personen

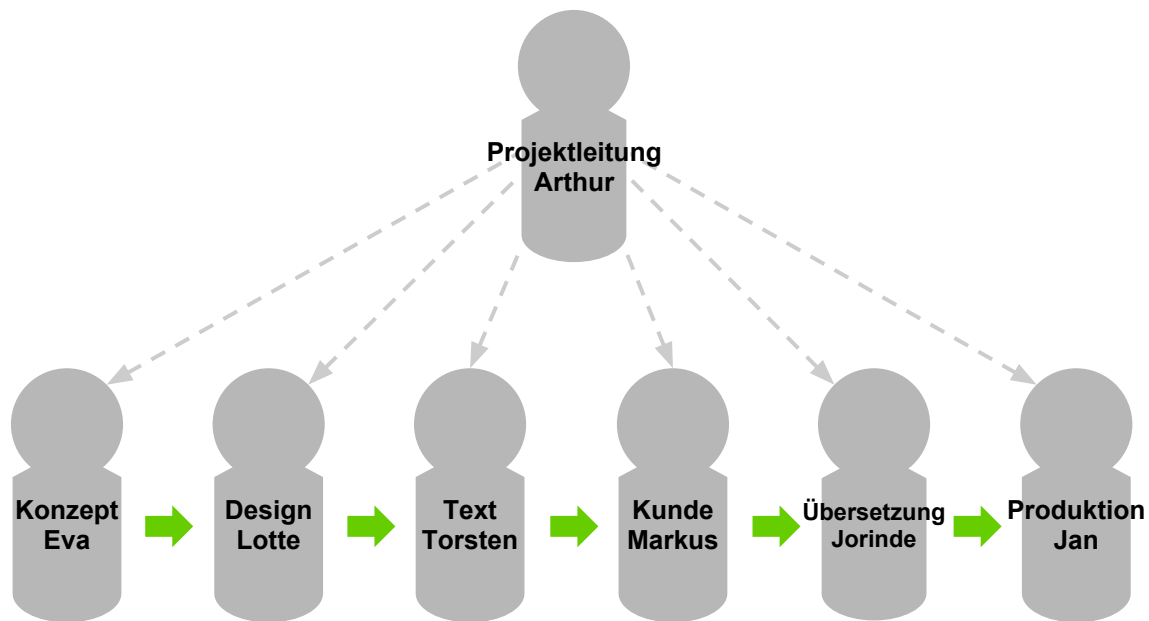


Abbildung 4: Übersicht über die Personas und den idealisierten Workflow

Bei der Formulierung der Personas wurde bewusst darauf verzichtet, persönliche Daten wie Alter und Bildung zu verwenden, da diese keinen Einfluss auf die Konzeption der Anwendung haben. Die Personas enthalten dementsprechend <sup>7</sup>

- 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 Projekts *über Texte* austauschen wird
- die verwendeten Werkzeuge zur Erfüllung der Aufgabe und die Erfahrung im Umgang mit Anwendungen im Allgemeinen
- die wichtigsten Szenarien, die die Persona mithilfe der Anwendung durchführen wird
- allgemeine Anforderungen an die Anwendung
- und Angaben darüber, wie der Zugriff auf die Anwendung erfolgt.

<sup>7</sup>Die Reihenfolge der einzelnen Punkte ist aus Layoutgründen nicht bei allen Personas gleich.

## 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 Produkts leicht erreichen zu lassen. Hierzu erstellt sie einen Überblick über das Produkt mithilfe 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* sowie *Lotte* über das Produkt ab. Sie gibt Feedback zu den Texten von *Torsten* und deren Integration in das Produkt durch *Jan*.

### **Werkzeuge und Erfahrung**

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

### **Szenarien**

Eva definiert die einzelnen logischen Bestandteile des Produkts (z.B. Seiten, Abschnitte) und legt darin fest, 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 Produkts enthält. So kann sie leicht den Überblick behalten.

### **Anforderungen**

Eva verwendet die Anwendung häufig, deswegen müssen Funktionen zur Definition des Produkts, der Texte und der Rahmenbedingungen einfach zu bedienen sein. Sie will nie die Übersicht verlieren und leicht Elemente verändern können, da in der Konzeptionsphase häufig Anpassungen am Produkt vorgenommen werden.

### **Zugang**

Eva greift auf die Anwendung 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* 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* bei der Gestaltung des Produkts ab. Mit *Torsten* spricht sie über ihre Vorgaben zu Texten und überprüft deren Einhaltung in der Umsetzung durch *Jan*.

### **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 die Anwendung 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* 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* in den Layouts verwenden.

Während der Umsetzung des Produkts ergeben sich Änderungen am Styleguide. Sie möchte, dass *Jan* alle Stellen im Produkt anpasst, die von diesen 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. Die Anwendung 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 der Anwendung.

### 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 des Konzepts und des Designs berücksichtigen sowie Wünsche und möglicherweise verbindliche Richtlinien des Kunden beachten. Hierzu erstellt Torsten für alle Textbausteine des Produkts die Texte in der endgültigen Fassung in seiner Muttersprache.

#### **Organisation, Abstimmung**

Torsten ist selbständig. Er stimmt sich mit *Eva* und *Lotte* bezüglich deren Vorgaben ab. Er berücksichtigt inhaltliche Vorgaben, die er von *Markus* erhält. Er steht *Jorinde* 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 Texte zum Produkt. Dazu befüllt er die vom Design vorgegebenen Textbausteine. Dabei kann er die Vorgaben sehen, die für den jeweiligen Text gelten (z.B. die maximale Textlänge).

Torsten benötigt Kontext-Informationen zum aktuellen Text und kann in der Anwendung die vom Kunden zur Verfügung gestellten Materialien aufrufen. Auch die Hinweise zur Zielgruppe und Funktion des Texts aus dem Konzept kann er sich ansehen, ohne die aktuelle Ansicht der Anwendung verlassen zu müssen.

Der Kunde wünscht Änderungen an den Texten. Diese sind in der Anwendung bei den jeweiligen Textbausteinen hinterlegt, so dass 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**

Der Editor zum Erstellen von Texten muss in der Anwendung an Torstens 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 die Anwendung 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 Produkts. 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 Produkts durch.

### **Organisation, Abstimmung**

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

### **Werkzeuge und Erfahrung**

Markus arbeitet vor allem mit Word, Excel und PowerPoint. Ansonsten hat er keine weiteren Erfahrungen im Umgang mit Anwendungen.

### **Szenarien**

Markus nimmt die fertigen Texte ab. Hierzu kann er einzelne Textbausteine freigeben oder Änderungen anfordern, die er mithilfe von Kommentaren beschreibt.

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

Materialien können in die Anwendung 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 in der Anwendung übernommen.

### **Anforderungen**

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

### **Zugang**

Markus arbeitet von seinem Büro-PC aus und greift auf die Anwendung 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 Projekts 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* bei inhaltlichen Fragen zu Texten ab. Für die Übersetzung spezieller Begriffe stimmt sie sich mit *Markus* 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 Anwendungen zu gewöhnen.

### **Zugang**

Jorinde greift auf die Anwendung 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*, 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 Projekts.

Das Projekt enthält an mehreren Stellen die gleichen Formulierungen. Jorinde bekommt 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 in der Anwendung 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 Produkts verantwortlich. Er hat aber auch während der Entwurfsphase Einfluss auf die Rahmenbedingungen für Texte, vor allem wenn es um technische Parameter geht (z.B. die maximale Zeilenlänge).

### **Organisation, Abstimmung**

Jan arbeitet auf Seiten der Agentur und stimmt sich mit *Eva* über den Aufbau und mit *Lotte* über die Gestaltung des Produkts ab. Von *Markus* 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 Texte aus der Anwendung mit den Texten im Dokument. Nachdem sich bereits verwendete Texte geändert haben, öffnet Jan das Dokument erneut und kann mithilfe 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 der Anwendung und erzeugt automatisch die Sprachdateien.

Jan entdeckt ein Problem mit der Textlänge einer Überschrift. Er meldet dieses Problem in der Anwendung. *Lottes* Änderungen betreffen alle Überschriften. Jan kann sich in der Anwendung alle betroffenen Stellen im Produkt anzeigen lassen.

### **Werkzeuge und Erfahrung**

Jan arbeitet je nach Produkt mit DTP-Produkten oder IDEs<sup>8</sup>, 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 die Anwendung zu. Er verfügt über eine schnelle Internetverbindung.

---

<sup>8</sup>für diese Persona ist es unerheblich, ob ein Medium oder Software entsteht

## 4.7 Arthur, Projektleiter



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

Arthur koordiniert als Projektleiter mit allen Beteiligten den Ablauf des Projekts, 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 die Anwendung vor allem immer verfügbar sein, Unterbrechungen im Projektverlauf durch einen Anwendungsausfall sind nicht akzeptabel. Auch ein Datenverlust muss ausgeschlossen sein. Es muss möglich sein alle Daten des Projekts zu exportieren.

### **Zugang**

Arthur greift von seinem Laptop auf die Anwendung zu und verfügt über eine schnelle Internetverbindung.

### **Szenarien**

Arthur legt ein neues Projekt an und fügt Mitarbeiter mithilfe 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 in der Anwendung 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 die Anwendung 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. Die Anwendung 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 Kommunikationsmedienprodukten 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 beschrieben, bevor beginnend ab Abschnitt 5.3 · S.39 die Anwendung konzipiert wird, die diese Anforderungen erfüllt.

In Kapitel 6 · S.49 wird auf Basis dieses Konzepts eine Anwendung entworfen und in Form eines Prototyps umgesetzt.

## 5.1 Abzubildender Workflow

Hauptaufgabe der Anwendung ist es, den Workflow von der Definition bis zur Übernahme der fertigen Texte in das Produkt abzubilden und dabei nicht nur Funktionen zum *Definieren*, *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 gezeigt, wie umfangreich die Anzahl der Personen ist, die Einfluss auf die Texte eines Produkts haben. Die Rollenverteilung ist dabei von Projekt zu Projekt verschieden. Mit den in Kapitel 4 vorgestellten Personas wurde 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 Textbausteins** 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 zugedachte 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 Projekts festgelegt.
2. Das **Schreiben eines Texts** 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 Projekts kopiert und eingefügt.
3. In der **Korrektur** wird der Text inhaltlich, orthografisch und grammatikalisch überprüft und entsprechend angepasst. Der Korrektor muss dabei für eine orthografische oder grammatikalische Überprüfung des Texts 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 Projekts, entspricht.
5. Durch die **Freigabe** wird der Text abgenommen und kann nun in das Produkt übernommen werden. Die Freigabe unterscheidet sich von der Qualitätskontrolle durch ihren au-to-ri-tativen Charakter. Qualitätskontrollen 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 finale Produkt eingebracht.



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

Tabelle 3: Stärke des Einflusses, den Mitarbeiter in einem Projekt haben

○ Kein Einfluss ● Viel Einfluss

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

1. den **Inhalt** des Texts,
2. den **Vorgaben** für diese Inhalte,
3. die **Attribute** wie z.B. »maximale Textlänge« oder »Position im Medium«,
4. und den **Status** wie z.B. »neu« und »freigegeben«.

Das Gewicht des Einflusses der Mitarbeiter ist je nach Rolle unterschiedlich, Tabelle 3 · S.33 zeigt dies in einer Übersicht.

Nachfolgend sind diese Text-Eigenschaften näher beschrieben.

### 5.1.1 Inhalt

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.

Personen die Einfluss auf den *Inhalt* haben sind vor allem diejenigen, die die Texte für das Produkt liefern. Dies sind vor allem Texter und Übersetzer. Üblich sind aber auch Anpassungen der Inhalte durch Experten. Ein Beispiel hierfür ist die Suchmaschinenoptimierung von Texten. Hierbei werden Texte auf das Vorhandensein von bestimmten Formulierungen und Stichwörtern hin optimiert.

### 5.1.2 Inhaltsvorgaben

Vorgaben für Inhalte werden in Form von Richtlinien formuliert und nicht ins Produkt übernommen. Diese Richtlinien dienen Textern als Orientierungshilfe, wie die Texte zu formulieren sind. Richtlinien werden von verschiedenen Mitarbeitern formuliert:

- Vom Konzept werden grundlegende Vorgaben geschaffen, wie z.B. Annahmen über die Zielgruppe und den Zweck des Produkts.
- 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.
- Nicht selten werden externe Experten zu Projekten hinzugezogen, die für bestimmte Aspekte Richtlinien definieren. Besonders bei Internetseiten werden SEO-Konzepte erstellt, die festlegen, dass die Texte bestimmte Schlüsselwörter und Formulierungen enthalten sollen, um in den Algorithmen der Suchmaschinenbetreiber bessere Positionierungen in Suchergebnissen zu erreichen.

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. Diese impliziten Vorgaben müssen ebenfalls explizit hinterlegt werden, um möglichst wenig Raum für Interpretationen durch die Autoren der Texte zu bieten.

### 5.1.3 Attribute

Attribute legen die Rahmenbedingungen von Text fest. Diese werden vor allem in der Gestaltung des Produkts 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 Texts festgelegt. Die Attribute lassen sich in vier Bereiche unterteilen:

**Identifizier** Dies sind eindeutige Bezeichner für Textbausteine. Sie dienen dazu, Referenzen zwischen den Textbausteinen in der Anwendung 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 mit

der gleichen Funktion 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 noch weitere Regeln für die Verwendung der Klassen festgelegt, z.B. dass eine Headline immer vor einem Fließtext stehen muss. Teil der Klassifizierung können auch Angaben zur Zeichenlänge sein. Da dies aber nicht immer zwingend der Fall ist und es auch Textlängenbeschränkungen ohne Bezug zur Klasse eines Texts geben kann (z.B. bei Formularen) werden diese separat betrachtet.

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

**Position** Für Textbausteine wird festgelegt, wo dieser im Produkt erscheint. Positionsangaben bestehen meistens aus zwei Komponenten. Zum einen wird eine hierarchische Position definiert, z.B. „im zweiten Kapitel“ oder „unter der Überschrift auf der Seite »Über uns«“. Zum anderen kann es exakte Positions- 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«.

#### 5.1.4 Status

Der 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 sollen, wenn sie als Gesamtes vorliegen – das dies in der Praxis nicht immer eingehalten werden kann, spielt an dieser Stelle keine Rolle. Auch externe Dienstleister bekommen aus Kostengründen meistens alle Texte im Paket, damit eine zügige Abarbeitung des Auftrages gewährleistet wird. Praktisch ist der Status eines Texts 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 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. Abbil-

Abbildung 5 · S.36 gibt einen Überblick über die Entscheidungsprozesse im Verlauf eines Projekts und wer auf diese Einfluss nimmt. Tabelle 4 · S.37 listet den Einfluss tabellarisch auf und enthält zusätzlich Informationen, wie stark der jeweils ausgeübte Einfluss in der Regel ist.

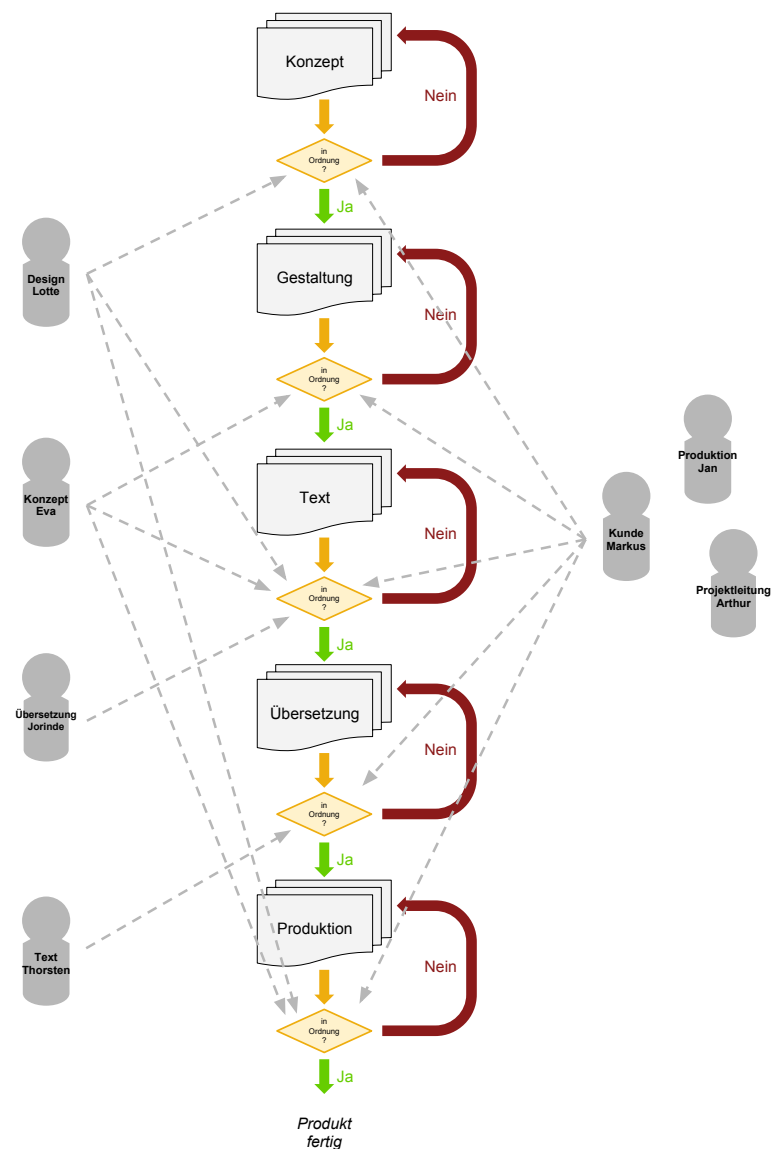


Abbildung 5: Einfluss auf den Status eines Textbausteins

**Ereignisse** Da sich die Texte des Produkts aus vielen einzelnen Bausteinen zusammensetzen, werden auch die Eigenschaften jeweils einzeln gesetzt. Aus Sicht des Projektverlaufs ergeben sich aus der Gesamtheit alle Zustände bestimmte Ereignisse, die sich auf die Ände-

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

Tabelle 4: Stärke des Einflusses auf den Status eines Textbausteins

○ Kein Einfluss ● Viel Einfluss

rung der Status der Texte zurückführen lassen. Immer dann, wenn alle Texte eines Produkts 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.

**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 Projekts 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 Anforderung – 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 Endgeräten** Es gibt unterschiedliche Anforderungen an den Zugang zur Anwendung, 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 Medienprodukten werden spezialisiert Werkzeuge verwendet, die durch die hier vorgestellte Lösung niemals ersetzt werden können. Es muss also eine Integration in diese Werkzeuge mithilfe von Plug-Ins oder Ähnlichem existieren, mit denen es möglich wird, die Texte aus der Anwendung komfortabel 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 Komponenten der Anwendung

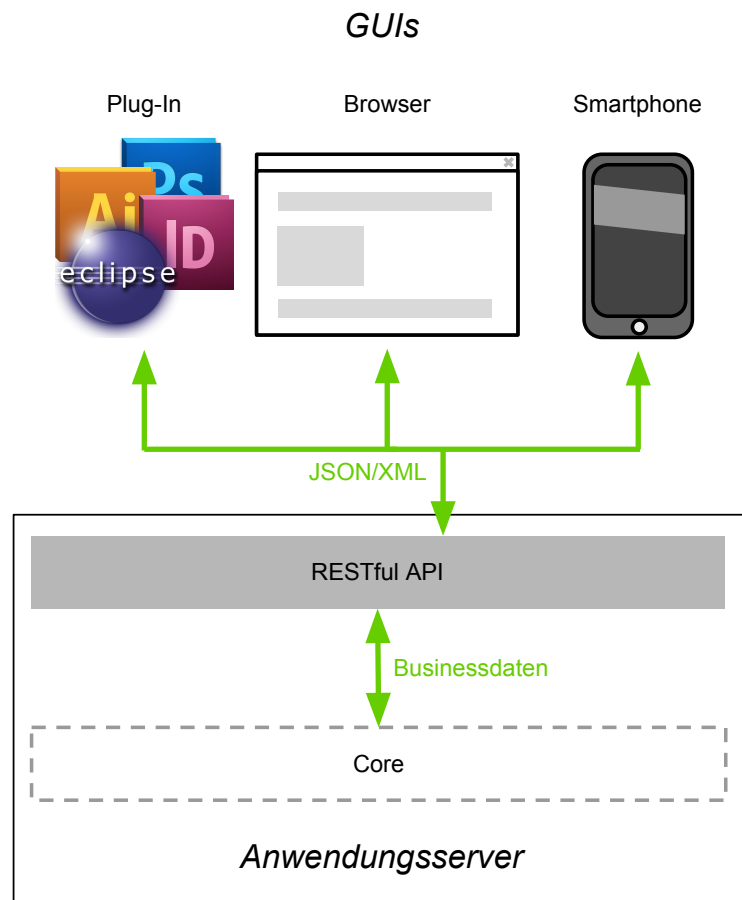


Abbildung 6: Aufbau der Anwendung in stark vereinfachter Darstellung

Die im vorangegangenen Abschnitt beschriebenen Anforderungen an die Anwendung, besonders im Hinblick auf den universellen, jederzeitigen Zugriff von überall, sind die entscheidenden Anforderungen, die für die Konzeption der Anwendung 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 mithilfe von HTML, CSS und JavaScript dar. Die Businesslogik und die Datenhaltung werden in einem Anwendungsserver realisiert, mit dem das GUI über eine Schnittstelle kommuniziert. Abbildung 6 · S.39 zeigt den Aufbau der Anwendung in stark vereinfachter Darstellung.

### 5.3.1 Browserbasiertes GUI

War es in den letzten Jahren noch üblich, dass Fragmente des GUIs mit serverseitigen Template-Sprachen erzeugt wurden (vgl. [3, S.48]), hat die zunehmende Verbreitung von mobi-

len Clients ein Umdenken zur Folge. Zum einen stellen Desktop-Clients, mobile Browser-Clients und native Apps zwar die gleichen Daten einer Anwendung dar, verwenden dafür aber nicht zwangsläufig die gleiche GUI-Technologie. Zum anderen werden Clients immer leistungsstärker, 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. [7] und [11]). Clients kommunizieren über Schnittstellen mit dem Anwendungsserver 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 Mobilfunk-Netz große Vorteile.

Für Web-Anwendungen bedeutet das, dass diese das zur Darstellung benötigte HTML direkt im Client erzeugen, mithilfe von JavaScript und darin implementierten Templating-Systemen. 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 mithilfe von JavaScript zwischen der Anwendung und dem Server ausgetauscht. Mobile Endgeräte, die über eigene GUI-Toolkits verfügen, oder Software von Drittanbietern können dann dieselben Schnittstellen verwenden, ohne dass serverseitige Anpassungen vorgenommen werden müssen.

Web-Anwendungen haben den Vorteil, dass sie ohne Installation auf dem Rechner des Benutzers lauffähig sind. Sie können also unmittelbar verwendet werden. Kompatibilitätsprobleme mit alten Browser-Versionen (z.B. dem Internet Explorer 6) können inzwischen mithilfe des ChromeFrame<sup>9</sup> komfortabel umgangen werden. Der Umfang an frei verfügbaren 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) existieren. Insgesamt sind Webbrowser der aktuellen Generation mächtige Werkzeuge zur Erstellung von CRUD-Anwendungen [15].

### 5.3.2 Schnittstellen

Die Verwendung einer einheitlichen Schnittstelle durch alle Clients ermöglicht ein konsistentes Verhalten der Anwendung über alle Zugangswege hinweg. Dies ist besonders im Fall dieser Anwendung von Bedeutung, da die Benutzer wünschen, dass sich die Texte direkt innerhalb ihrer bevorzugten Werkzeuge abrufen und einbinden lassen. Dies ist nur mithilfe von Plug-Ins möglich, die in der jeweiligen Software-Umgebung der Werkzeuge

---

<sup>9</sup><https://developers.google.com/chrome/chrome-frame/>



entwickelt werden müssen. Aus diesem Grund ist es unvermeidlich, dass für alle Funktionen der Anwendung eine öffentliche Schnittstelle existiert.

Aufgrund des flexiblen Aufbaus, der Möglichkeit ausgewählte Anfragen leicht mit HTTP-Caches zu beschleunigen und der freien Wahl des Nachrichtenformats ist eine an den REST-Prinzipien orientierte Implementierung der Schnittstelle gut geeignet für die Abbildung der Operationen der Anwendung. Die Struktur dieser Kommunikation 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.

### 5.3.3 Anwendungsserver

Die Orientierung an den REST-Prinzipien hat auch Einfluss auf die Konzeptionierung des Anwendungsservers (vgl. [4, S.127]). Grundsätzlich ermöglicht dies den Verzicht auf komplexe Servlet Container mit langer Lebenszeit, die Ressourcen und Zustände zwischen Anfragen beibehalten. Der Anwendungsserver kann, vereinfacht gesprochen, als Input-Output-Maschine implementiert werden, dessen Lebenszyklus nur den Verlauf einer Anfrage überdauert. Dieses Programmiermodell findet sich in populären Sprachen zur Entwicklung von Webanwendungen, wie z.B. PHP oder Python, wird aber inzwischen auch wieder vermehrt von neueren Entwicklungen im Bereich der asynchronen Serversoftware, z.B. NodeJS<sup>10</sup> oder vert.x<sup>11</sup>, aufgegriffen.

Die Abschnitte 6.4 · S.66 und 6.6 · S.72 im nächsten Kapitel beschreiben eine mögliche Implementierung dieses Konzepts ausführlicher.

Der Betrieb des Anwendungsservers erfolgt üblicherweise als *Software as a Service* (SaaS), hierbei wird eine Instanz der Anwendung auf einem Internetserver betrieben und alle Benutzer greifen auf diesen zu. Die Vorteile dieses Konzepts 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 und ohne deren Zutun aktualisieren zu können, eliminiert vielfältige Probleme, die sonst in Umgebungen entstehen, in denen unterschiedliche Programmversionen existieren. Die Zentralisierung von Anwendungssoftware birgt aber auch einige Risiken. Besonders bei Instanzen, die von mehreren Unternehmen verwendet werden, müssen im Hinblick auf Zugriffsschutz und Datensicherheit besondere Vorkehrungen getroffen werden, die so bei klassischen, nur unternehmensintern zur Verfügung stehenden, Anwendungen nicht notwendig sind. Die Konzeptionierung einer geeigneten Systemumgebung für den Betrieb der Anwendung muss aufgrund des Umfangs gesondert betrachtet werden – diese Maßnahmen habe zudem wenig

---

<sup>10</sup><http://nodejs.org/>

<sup>11</sup><http://vertx.io/>

Bedeutung für den Kern-Funktionsumfang der Anwendung. Maßnahmen zum Schutz vor unbefugtem Zugriff werden in diesem Konzept deswegen mithilfe eines einfachen Benutzermodells nur angedeutet, da diese, je nach Art des Betriebs der Anwendung, unterschiedlich realisiert werden können.

## 5.4 Zugang

Die im vorigen Abschnitt beschriebene Konzeption der Anwendung als browserbasierte Web-Anwendung mit vollständiger Schnittstellen-Abdeckung macht es möglich, jedem Mitarbeiter einen für ihn passenden Zugang zu ermöglichen. Im Einzelnen sind das:

**Browserbasierter Zugang vom Desktop** Den Zugang zur Anwendung mit einem 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, dieselben Inhalte anzuzeigen, wie ihr Desktop-Äquivalent. 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 bereitzustellen.

**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 der Anwendung in ihren Entwicklungsprozess optimal an die jeweiligen Bedürfnisse anpassen.

**Exporte** Die Möglichkeit, die Texte des Projekts in verschiedene Formate zu exportieren ist eine wichtige Funktion. Sie ermöglicht zum einen die Übernahme in Systeme und Anwendungen, deren Anbindung nicht möglich oder gewünscht ist und schafft zum anderen die Möglichkeit, ähnlich wie beim Zugang über 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 der Anwendung 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 der Anwendung 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 und ähnliche Systeme stattfinden. Hierbei werden Informationen

zu einem Ereignis übertragen z.B. die Status-Änderungen eines Texts. Denkbar sind 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 Bedürfnisse von Anwendergruppen mit besonderen Anforderungen abgedeckt werden, die bei der Konzipierung der Anwendung nicht berücksichtigt wurden.

## 5.5 Überblick über den Aufbau der gesamten Anwendung

Abbildung 7 liefert einen Überblick über den Aufbau der gesamten Anwendung:

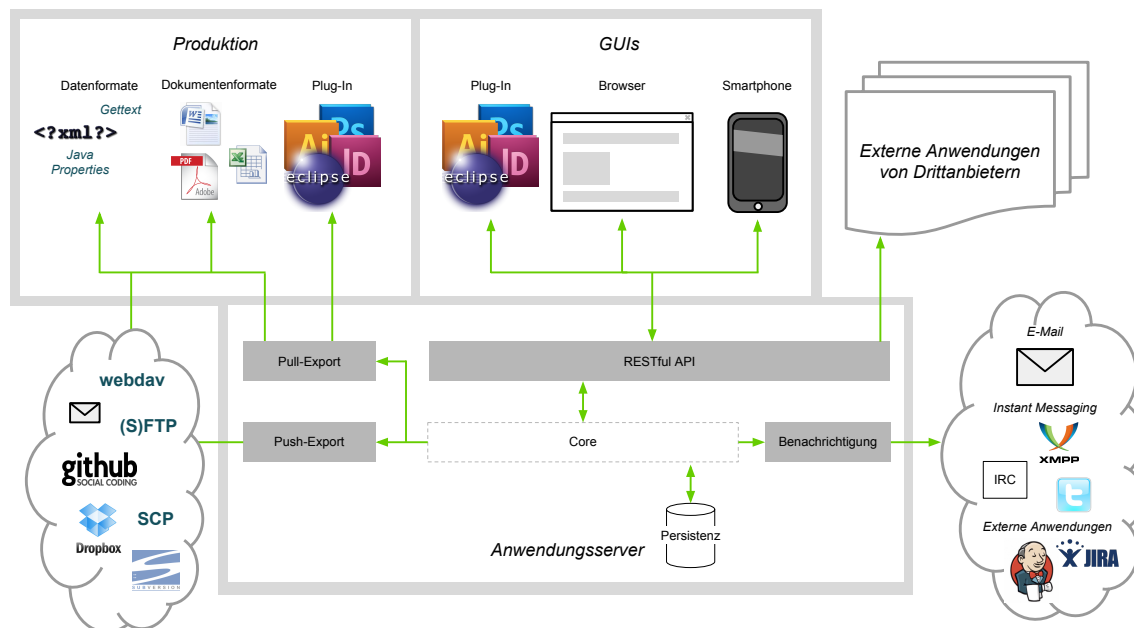


Abbildung 7: Aufbau der gesamten Anwendung im Überblick

Die zentrale Komponente der Anwendung bildet der Anwendungsserver. Für die Benutzer erfolgt der Zugriff mithilfe eines GUI, die mit der RESTful-API des Servers kommuniziert. Eine browserbasierte GUI auf Basis von HTML5 und JavaScript bildet den Hauptzugang zur Anwendung, der auch auf Smartphones verwendet werden kann. Zusätzlich gibt es spezielle Plug-Ins 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, eigene Clients für die Anwendung zu entwickeln. In die Endprodukte gelangen die Texte über den Export, exportiert wird dabei in viele Formate, neben Datenformaten wie z.B. XML werden auch Dokumentenformate wie z.B. Word unterstützt. 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 Orten, wie z.B. FTP-Server oder Versionsverwaltungssysteme. Die Benachrichtigungen über Aufgaben und Änderungen an Texten kann via E-Mail aber auch mittels Instant-Messaging-Systemen oder durch den Aufruf fremde API-Endpunkte erfolgen.

## 5.6 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 der Anwendung ohne zusätzliche Aufwände, wie die Installation neuer Software, verwenden können. Für spezielle Anwendungsfällen ist es mithilfe 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 einer solchen Anwendung 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 der Anwendung, einen vertrauenswürdigen Betreiber für die Server-Infrastruktur zu finden. Alternativ ist es jedoch problemlos möglich, die Anwendung 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 kann dies aber vorausgesetzt werden. 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 Konzeptes 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 das in diesem Kapitel vorgestellte Konzept umsetzt. Die Kernfunktionen der Anwendungen werden dazu prototypisch implementiert, im Einzelnen sind das die Definition eines Produkts, das Befüllen der Texte mit Inhalten, der Austausch über die Texte und die Abnahme sowie der Export des Projekts in Formate, mit denen das Produkt umgesetzt werden kann. Dies dient dazu, die Konzeption des Workflows und der Anwendung zu validieren.





## 6 Entwurf des zentralen Systems

Im vorigen Kapitel wurde eine Anwendung konzipiert, mit der sich die in Kapitel 3 · S.5 genannten Probleme beheben lassen. In diesem Kapitel wird ein Prototyp entworfen, der dieses Konzept umsetzt. Mithilfe des Prototyps soll gezeigt werden, dass sich der konzipierte Workflow mit der vorgeschlagenen Anwendungsart – einer browserbasierte Web-Anwendung mit vollständiger Schnittstellen-Abdeckung – abbilden lässt und einen Mehrwert für Projekte zur Erstellung von Informations- und Kommunikationsmedienprodukten bietet.

Der Funktionsumfang des Prototyps ist eingeschränkt, deckt aber die wichtigsten Anforderungen ab. Im Prototyp soll es möglich sein

- die Struktur eines Produkts zu definieren
- die Texte des Produkts zu befüllen
- sich mit anderen Mitarbeitern über die Texte auszutauschen
- die Texte zu prüfen und freizugeben
- das Projekt zu exportieren, um die Texte weiterverarbeiten zu können.

Die Beschreibungen der Bestandteile der Anwendung in diesem Kapitel gehen zum Teil über diesen Funktionsumfang hinaus, da die Implementierung des Prototyps auch dazu dient, zu zeigen, dass die vorgeschlagene Architektur geeignet ist, als Vorlage für eine belastbare Implementierung der Anwendung für den realen Einsatz zu dienen. Der Entwurf liefert hierfür schon wichtige Vorüberlegungen.

## 6.1 Komponenten

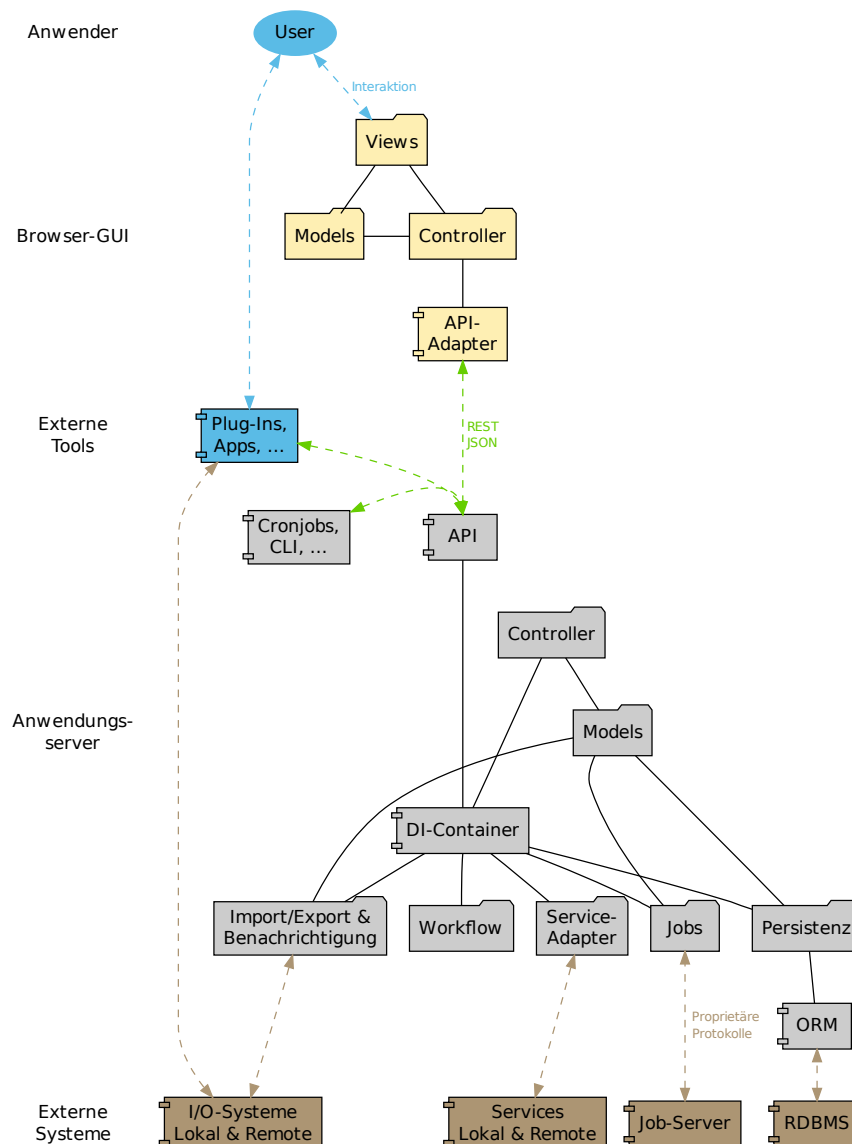


Abbildung 8: Komponenten der Anwendung

Abbildung 8 liefert einen Überblick über die in den folgenden Abschnitten beschriebenen Komponenten der Anwendung. *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 Anwendungsservers. *Braun* eingezeichnet sind (aus Sicht des Anwendungsservers) externe Dienste, die durch diesen angebunden werden. *Gestrichelte Linien* zeigen Kommunikationsverbindungen zwischen Komponenten über Schnittstellen. *Durchgezogene Linien* zeigen Verbindungen innerhalb eines Systems.

Der zentrale Bestandteil der Anwendung ist der Anwendungsserver, mit dem verschiedene GUIs über Schnittstellen kommunizieren. Das wichtigste GUI ist das browserbasierte Interface, da hiermit alle am Projekt beteiligten Personen arbeiten. Abschnitt 6.6 · S.72 beschreibt die Komponenten des Anwendungsserver und Abschnitt 6.3 · S.55 die Gestaltung des browserbasierten GUIs. In Abschnitt 6.4 · S.66 wird die Anbindung dieses GUIs an den Anwendungsserver beschrieben.

Zur Validierung des Entwurfes wurde ein Querschnitt durch die vorgestellte Architektur als 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 Abschnitt 6.5 · S.68 und 6.7 · S.75 wird die prototypische Implementierung der GUI bzw. des Anwendungsservers, die auf dem Entwurf basiert, beschrieben.

Im nächsten Abschnitt wird jedoch zuerst das Domänenmodell, das die gemeinsame Grundlage für GUI und Anwendungsserver bildet, beschrieben.

## 6.2 Domänenmodell

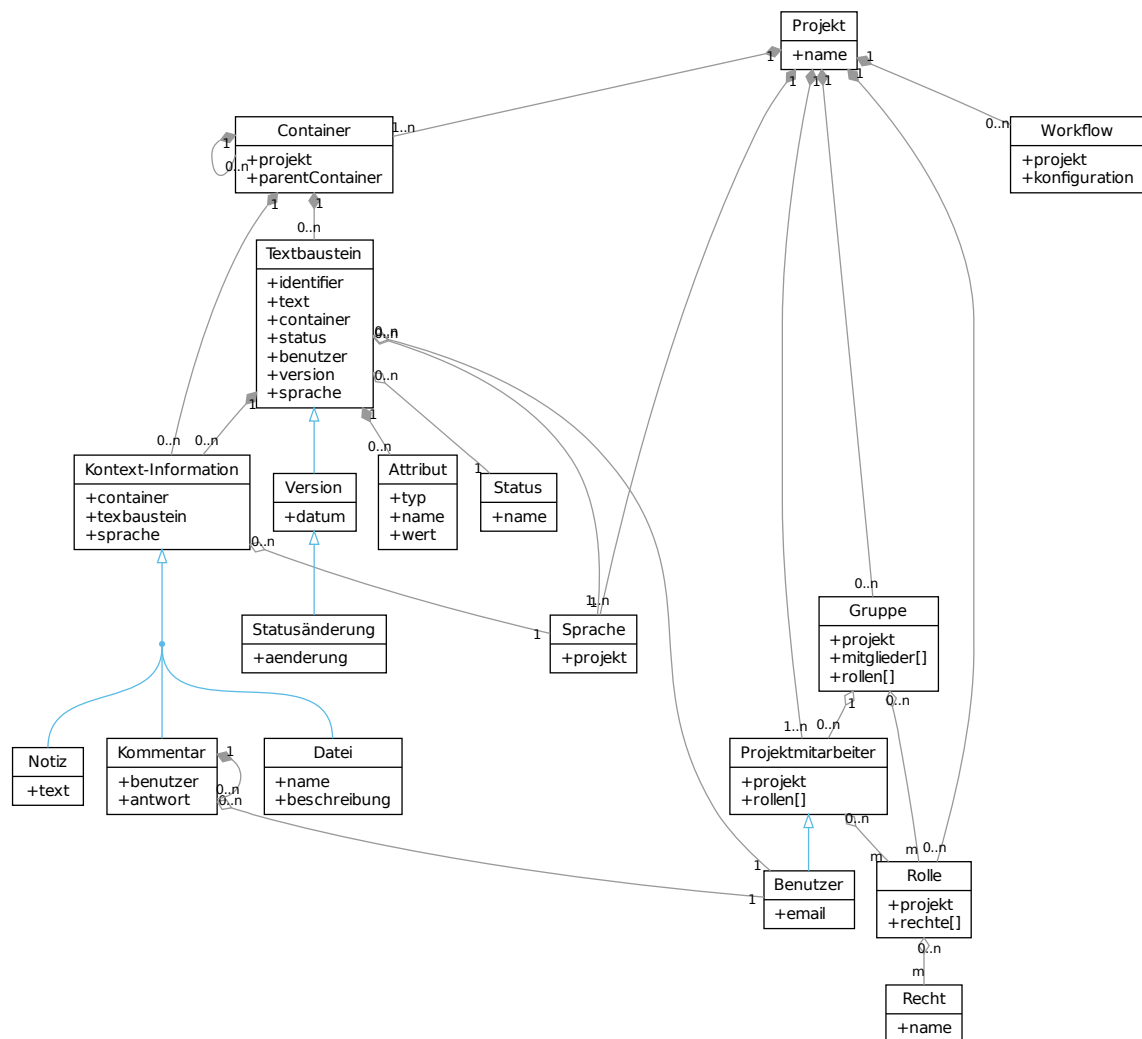


Abbildung 9: Domänenmodell

Aus den vorangegangenen Überlegungen zur Anwendung und zum Workflow lässt sich ein Domänenmodell extrahieren, das die einzelnen logischen Objekte innerhalb der Anwendung beschreibt. Abbildung 9 zeigt das Modell in der Übersicht. Der Entwurf in diesem Kapitel und der Prototyp orientieren sich an diesem Domänenmodell.

Die einzelnen Objekte werden nachfolgend beschrieben:

### Attribut

Beschreibt die Attribute eines Textbausteins.

### Benutzer

Repräsentiert einen Benutzer des Systems.

### Container

Container dienen zur hierarchischen Organisation der Texte innerhalb des Projekts. Container können weitere Container und Texte enthalten. Ein Container ohne übergeordneten Container befindet sich auf der obersten Ebene. Es kann mehrere Container 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 bilden den Rahmen für alle Texte eines einzelnen Produkts. Dieses Model geht davon aus, dass auch alle anderen Einstellungen (Gruppenzugehörigkeit, Rechte, Workflows, ...) immer projektspezifisch sind. Dies ist in einem Szenario sinnvoll, in dem unterschiedliche Unternehmen Zugriff auf die selbe Instanz der Anwendung haben. So wird eine saubere Trennung der Daten erreicht.

### Projektmitarbeiter

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

### Recht

Beschreibt das 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 [5.1.4](#) · S.35).

1. Neu, Textbaustein erzeugt
2. Leer, Textbaustein definiert
3. Befüllt, Textbaustein mit Inhalt befüllt
4. Korrigiert, Orthografie geprüft
5. Geprüft, Inhalt geprüft (Qualitätssicherung)
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.

## 6.3 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.21 gezeigt wurde. Diesen Umständen muss bei der Gestaltung Rechnung getragen werden. In Anlehnung an [14] 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 es leicht, zusätzliche Anwendungen für besondere Benutzergruppen zu schaffen, z.B. Plug-Ins für die speziellen Werkzeuge einzelner Mitarbeiter. Für diese »Power-User« ist eine solche Herangehensweise meistens der bessere Weg.

Anhand der Personas lässt sich ermitteln, wer und in welchem Umfang das browserbasierte GUI verwenden wird. Tabelle 5 · S.56 zeigt in der Übersicht, welche Operationen besonders bei der Entwicklung des GUIs beachtet werden müssen. Besonders viel Zeit werden von Konzeptern und von den Mitarbeitern, die die Inhalte erstellen, kontrollieren und freigeben im GUI verbracht, da diese Tätigkeiten bezogen auf die einzelnen Texte des Produkts sehr arbeitsintensiv sind. In den folgenden Wireframes werden dementsprechend die wichtigsten Ansichten des GUIs beschrieben.

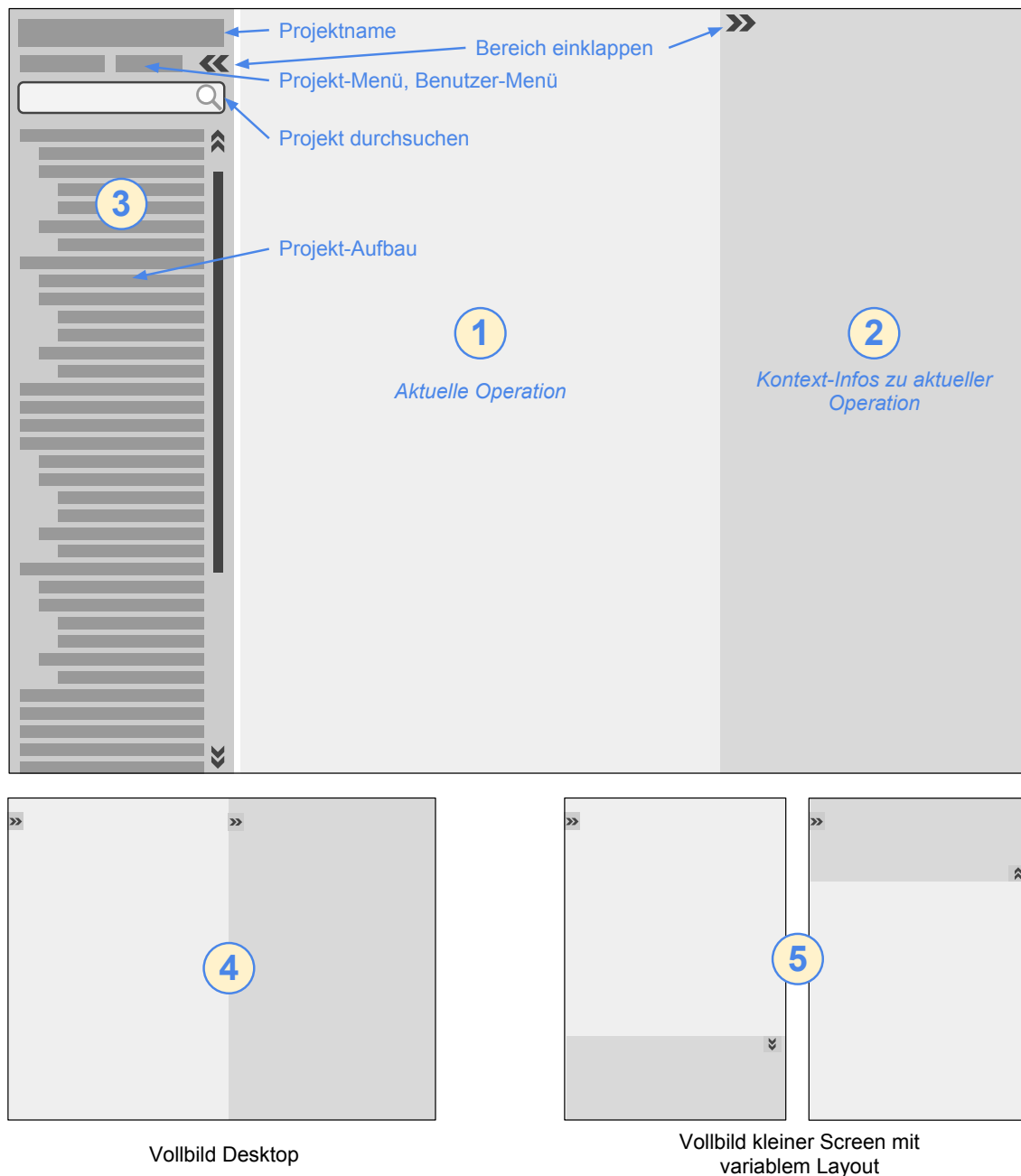
Operation (vgl. 5.1)	<b>Eva</b> Konz.	<b>Lotte</b> Des.	<b>Torsten</b> Texter	<b>Jorinde</b> Übersetz.	<b>Jan</b> Prod.	<b>Arthur</b> Projektl.	<b>Markus</b> Kunde
Definieren	●	◐	○	○	○	○	○
Schreiben	◐	○	●	●	○	○	○
Korrektur	○	○	◐	◐	○	◐	◐
Qualitätskontrolle	◐	○	○	○	○	●	◐
Freigabe	○	○	○	○	○	◐	◐
Veröffentlichung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

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

○ selten ● häufig



### 6.3.1 Aufbau des GUIs



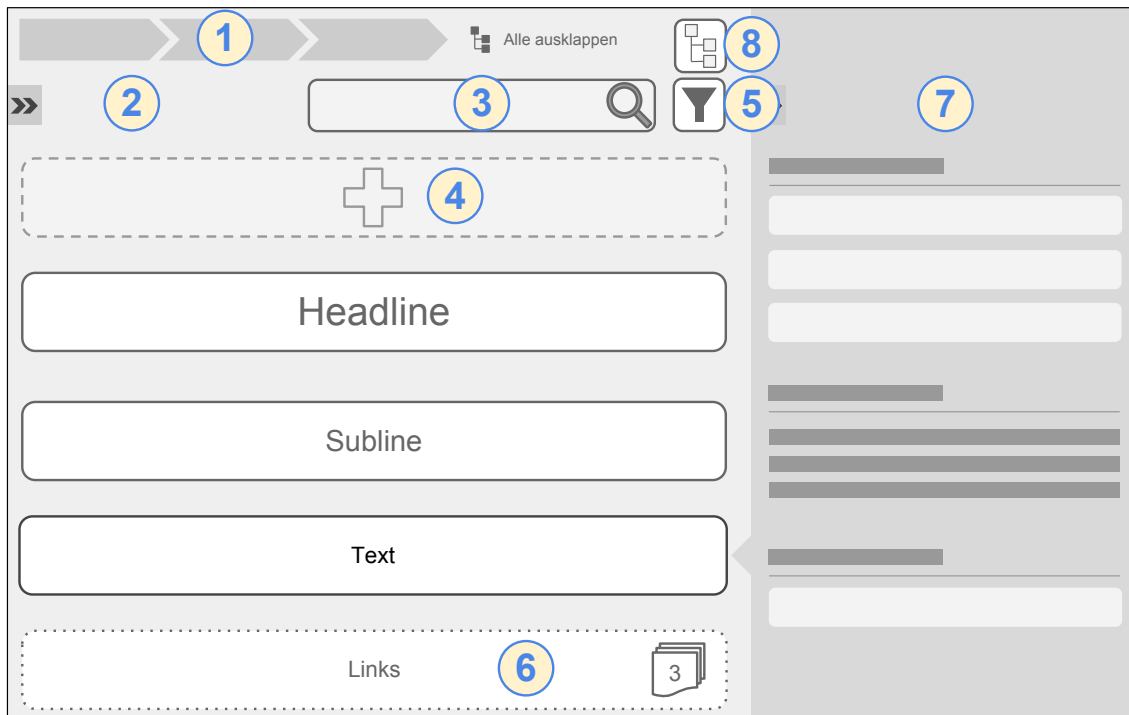
Wireframe I: Aufbau des browserbasierten GUIs

Wireframe I 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 Name des aktiven Projekts. Über das ausklappbare Projektmenü gelangt man zu Ansichten der Anwendung, die nicht direkt über die Texte erreichbar sind, z.B. die Projektauswahl, die Mitarbeiterverwaltung oder den Export. Über das ausklappbare Benutzermenü kann man sich ausloggen, sein Profil bearbeiten und persönliche Einstellungen anpassen.

### 6.3.2 Definieren des Produkts

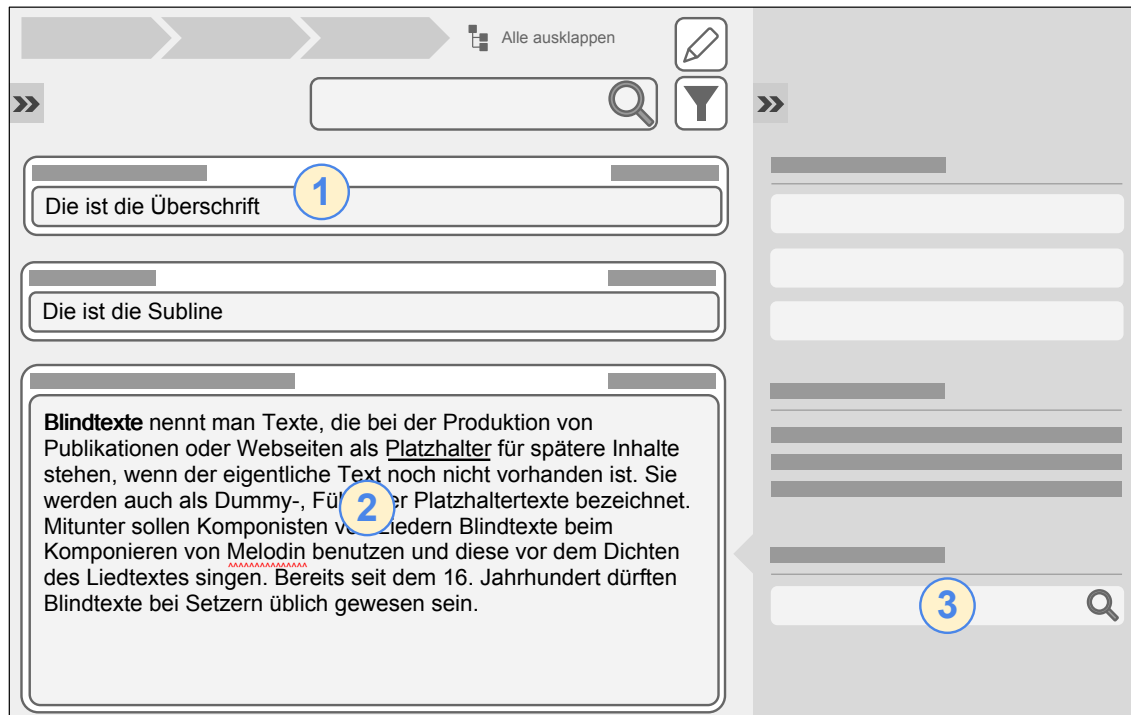


Wireframe 2: Bearbeiten der Produktstruktur

Wireframe 2 zeigt die Darstellung des GUIs zur Bearbeitung der Produktstruktur. Über die Breadcrumb-Navigation ①, die die aktuelle Position in der Projekthierarchie anzeigt, ist eine Orientierung auch ohne das eingeklappte Projektmenü möglich. Über eine Schaltfläche lassen sich alle Container der aktuellen Hierarchieebene aufklappen, so 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 einer kompakten Weise, die einen schnellen Überblick über die Inhalte bietet ohne viel Scrollen zu müssen. Die einzelnen Inhalts-Elemente können mit Drag&Drop in ihrer 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 in der Ansicht zu filtern, z.B. um nur die Elemente 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 Elemente an. In der Spalte für die Kontext-Information ⑦ werden Detailinformationen zum aktuell ausgewählten Element dargestellt, die an die eingestellte 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 · 6.3.2, Erstellen · 6.3.3, Übersetzen · 6.3.4 und Prüfen · 6.3.5), die die Inhaltselemente manipulieren, gewechselt werden.

### 6.3.3 Texte erstellen



Wireframe 3: Erstellen der Texte

Wireframe 3 zeigt die Darstellung des GUIs zur Bearbeitung der Texte des Produkts. Die Inhaltselemente in der aktuellen 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 Längenbeschränkungen (falls vorhanden) mit entsprechendem 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.

### 6.3.4 Texte übersetzen



Wireframe 4: Übersetzen der Texte

Wireframe 4 zeigt die Darstellung des GUIs zur Übersetzung der Texte des Produkts. Hierzu wird der Inhalt des Textbausteins 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.

### 6.3.5 Prüfen



Wireframe 5: Überprüfen der Texte

Wireframe 5 zeigt die Darstellung des GUIs zur Kontrolle und Freigabe der Texte des Produkts. In dieser Ansicht werden die verschiedenen Status der Textbausteine bearbeitet. Hierzu sind die Textbausteine mit ihren Übersetzungen dargestellt ①. Über die mit Icons markierten Schaltflächen ② lassen sich die einzelnen Status 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 Status 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.

### 6.3.6 Kontext-Informationen



Wireframe 6: Kontext-Informationen

Wireframe 6 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 auf.
2. Änderungshistorie: Anzeige der vergangenen Änderungen, 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 Produkts, wie z.B. die Attribute, an.

Über das Suchfeld ② lässt sich die aktuelle Ansicht filtern. In der 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 die Diskussion mithilfe der Kommentarfunktion ist möglich ⑤.



---

Die in diesem Abschnitt gezeigten Wireframes sind die Vorlage für eine mögliche Implementierung. Bei der Umsetzung des Prototyps wurde versucht, dieser Vorlage weitestgehend zu folgen, wie in Abschnitt [6.5](#) · [S.68](#) beschrieben wird. Im nächsten Abschnitt wird jedoch zuerst auf die Verbindung des Anwendungsservers mit einem GUI eingegangen.

## 6.4 Anbindung des GUI an den Anwendungsserver

Die in Abbildung 8 · S.50 *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 Anwendungsserver wird in einer eigenen Komponente, dem *API-Adapter*, gekapselt, die über die RESTful-Schnittstelle JSON-Daten mit dem Anwendungsserver 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 Anwendungsserver, die nicht zwingend 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 [7, S.1–5 und S.65–72] ist dieses Prinzip ausführlich erläutert.

Ein Problem beim Datenaustausch mithilfe von JSON-Objekten ist, dass diese üblicherweise nur reine Daten enthalten, da einfach nur 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 in der Anwendung hart kodiert. Um zukünftigen Änderungen an der Schnittstelle ohne Änderungen auf Clientseite begegnen zu können, sollten die Antworten der Schnittstelle mit Metainformationen versehen sein, die es dem Client ermöglichen, ohne hart kodierte Zuordnung zwischen Endpunkt und Domänenobjekt auszukommen.

JSON-LD<sup>12</sup> 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 lässt sich die vollständige Datenstruktur einer Anwendung als Graph darstellen. Hierzu werden alle Objekte mit Relationen anno-

---

<sup>12</sup><http://json-ld.org/>

tiert und der Client kann entscheiden, welchen Relationen er folgt und welchen nicht. Da 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 Einstiegspunkt zu kennen von dem aus man sich durch den Graphen »hangeln« kann, indem man Objekt-Relationen folgt, deren Kontexte die Anwendung erkennt.<sup>13</sup>

---

<sup>13</sup><http://coderbyheart.de/blog/relationen-in-linked-data>

## 6.5 Implementierung des GUI

Im Prototyp wurde das GUI als reine JavaScript-Anwendung umgesetzt. Als Framework für den Aufbau der Anwendung wurde Backbone.js<sup>14</sup> eingesetzt.

### 6.5.1 Komponenten

Backbone.js stellt Basisklassen für Models, Views und Collections (Listen von Models) zur Verfügung, die mithilfe 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 angezeigt wird.

Um die einzelnen Bestandteile der Anwendung möglichst modular und die Ladezeiten so kurz wie möglich zu halten, kommt require.js<sup>15</sup> 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. Hierdurch wird zudem eine saubere Trennung zwischen der Darstellungslogik, die in JavaScript implementiert ist und dem zugehörigen HTML-Quellcode ermöglicht. Underscore.js<sup>16</sup>, auf dem Backbone.js aufbaut, liefert hierzu einfache Templating-Funktionen.

Zur Realisierung der GUI-Funktionalität wurde Twitter Bootstrap<sup>17</sup> eingesetzt, ein HTML5-Framework, das für viele GUI-Elemente bereits vorgefertigte Darstellungen bietet. Einige Elemente werden mithilfe von JavaScript mit erweiterter Funktionalität versehen, in Twitter Bootstrap kommt dabei JQuery<sup>18</sup> 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 Endgeräten sichergestellt. Mithilfe von CSS *media queries* können einfach Anpassung für bestimmte Bildschirmgrößen implementiert werden.

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

---

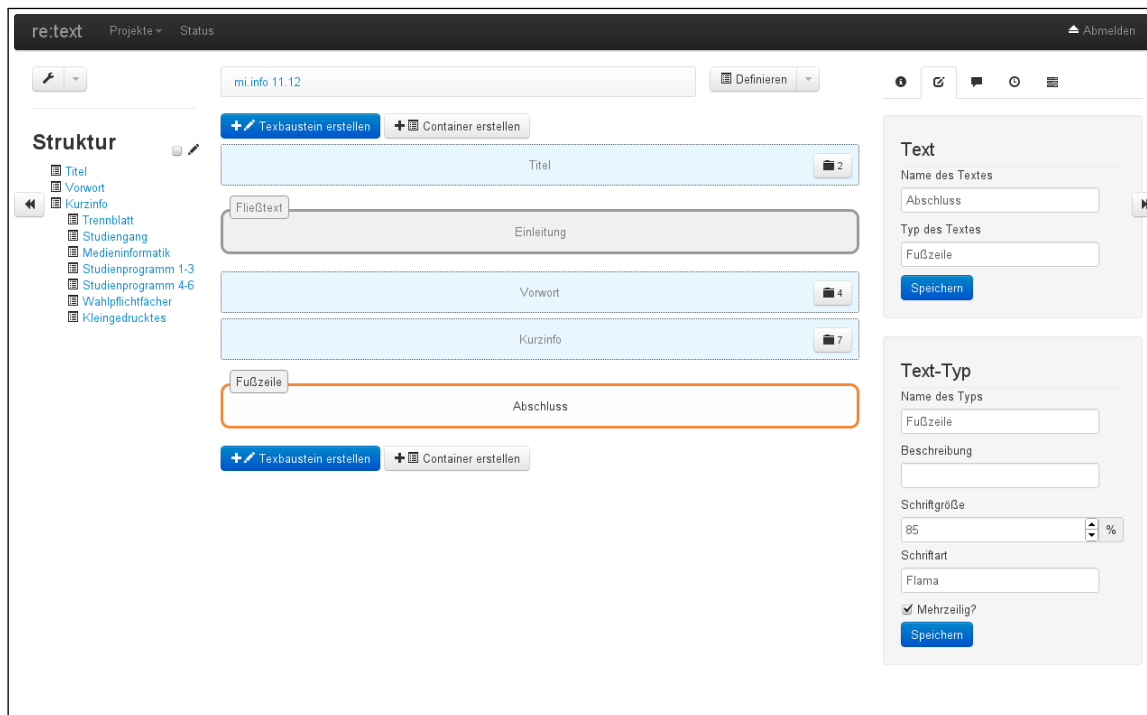
<sup>14</sup><http://backbonejs.org/>

<sup>15</sup><http://requirejs.org/>

<sup>16</sup><http://underscore.org/>

<sup>17</sup><http://twitter.github.com/bootstrap/>

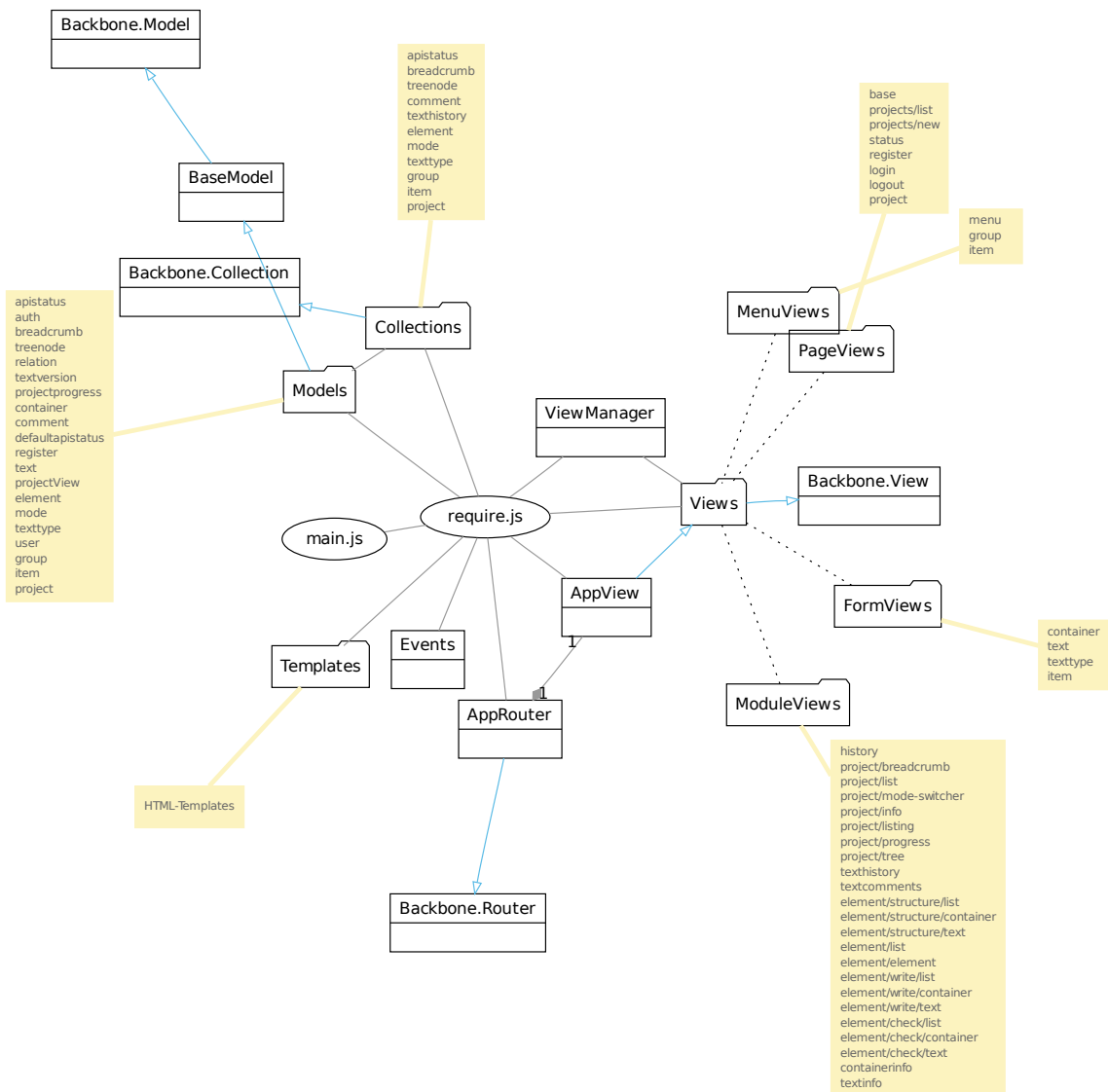
<sup>18</sup><http://jquery.com/>



Screenshot 1: Browserbasiertes GUI des Prototyps (vgl. Abschnitt 6.3.2 · S.59)

Screenshot 1 zeigt den GUI-Aufbau des Prototyps beispielhaft in der Ansicht zur Definition des Produkts. Die Anordnung der Elemente des Entwurfs 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.1 · S.I.

### 6.5.2 Klassendiagramm



### Abbildung 10: GUI-Klassendiagramm des Prototyps

Abbildung 10 zeigt das Klassendiagramm der browserbasierten Anwendung. Zentraler Einstiegspunkt ist die JavaScript-Datei `main.js`, die die Anwendung initialisiert. Dazu lädt sie mithilfe 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 den Templates nach. Abhängigkeiten wie JavaScript-Objekte und Templates werden dabei zentral von `require.js` bereitgestellt, das damit die Aufgabe eines Dependency-Injection-Containers (DI-Container) 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 [6.7.2 · S.75](#).

---

Das browserbasierte GUI ist zwar eine eigenständige Anwendung, ohne einen Anwendungsserver lässt sich hiermit jedoch nicht arbeiten, da das GUI lediglich zum Darstellen und Modifizieren von Daten verwendet wird. Die Persistenz und vor allem die Sicherung von Datenzugriffen, Berechtigungen, sowie viele weitere Funktionen werden im Anwendungsserver implementiert, der im nächsten Abschnitt beschrieben wird.

## 6.6 Entwurf des Anwendungsservers

In diesem Abschnitt werden die in Abbildung 8 · S.50 grau eingefärbten Komponenten des Anwendungsservers beschrieben.

### 6.6.1 API, Cronjobs, CLI, ...

In der API-Komponente, die als RESTful-API implementiert ist, sind über eine Routing-Tabelle alle Controller registriert. Da das System mit einer vollständigen Abdeckung aller Operationen durch API-Endpunkte implementiert ist, wird diese auch für lokal ausgeführte Skripte 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.

### 6.6.2 Dependency-Injection-Container

Die einzelnen Komponenten des Anwendungsservers sind nach dem Prinzip der losen Kopplung verbunden (vgl. [6, S.62]). Hierzu werden die einzelnen Komponenten als Services in einem 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, sondern von außen »hereingebracht« werden, ist leicht zu modifizieren und vor allem leicht zu testen, da sich Abhängigkeiten für Tests leicht durch Mock-Objekte austauschen lassen (vgl. [5, Kap.2]).

### 6.6.3 Controller

Zentrale Komponente bilden die Controller. Hier ist die Business-Logik der Anwendung implementiert. Die einzelnen Operationen sind in verschiedenen Controllern implementiert, so ist das System leicht erweiter- und wartbar. Die Controller werden von der API-Komponente entsprechend 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-Komponente an den Client zurückgeschickt wird.

### 6.6.4 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 weder Logik enthalten noch von anderen Komponenten abhängig sind. Siehe hierzu auch das Domänenmodell auf S.52.



### 6.6.5 Persistenz, ORM

Mit dieser Komponente werden die innerhalb der Anwendung erzeugten Daten persistiert. Mithilfe eines Object-Relational-Mappers (ORM) werden die Domänendaten, 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 eine entsprechende Komponente ersetzt.

### 6.6.6 Import/Export & Benachrichtigung

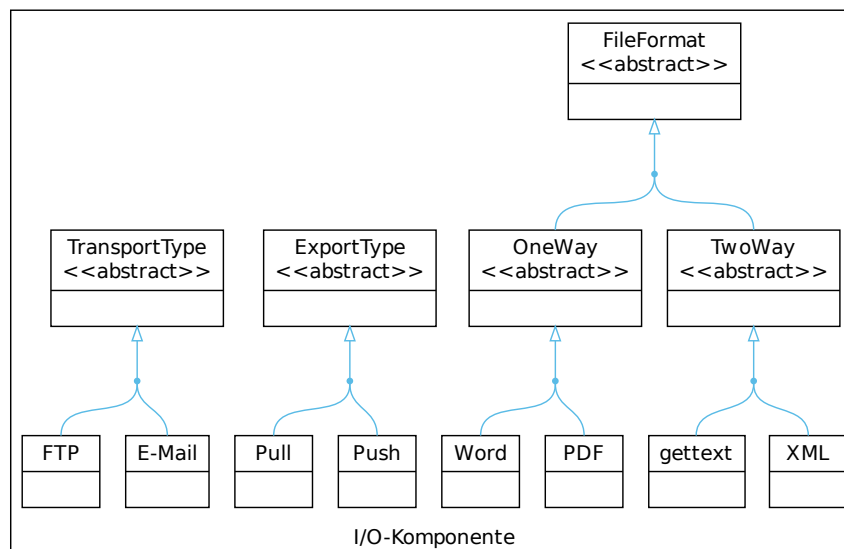


Abbildung 11: I/O-Komponente

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 für die Implementierungen des Transportwegs (z.B. *E-Mail* oder *FTP*), der Art des Ex- oder Importes (synchron, asynchron bzw. *Push* oder *Pull*) und das Dateiformat (vgl. Abb. 11). Aus diesen drei Teilen lassen sich beliebige Kombinationen erstellen. Beispiele sind synchrone Exporte, wie den Download eines Exports über die Schnittstelle oder 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.

### 6.6.7 Jobs

Zeitaufwändige und wiederkehrende Operationen, 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-Servers eingestellt und dort durch registrierte Job-Runner verarbeitet.

### 6.6.8 Workflow

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

### 6.6.9 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.



Die in diesem Abschnitt vorgestellten Komponenten bilden gemeinsam ein umfangreiches, komplexes System, mit dem sich alle Anforderungen abbilden lassen. Der im nächsten Abschnitt beschriebene Prototyp verwendet aufgrund seines sehr überschaubaren Funktionsumfangs nur einen Teil der genannten Komponenten. Es wird jedoch darauf geachtet, die Implementierung möglichst nahe an dieser Vorlage zu realisieren.

## 6.7 Implementierung des Anwendungsservers

Für den Prototyp wurde das PHP-Framework **Symfony2**<sup>19</sup> eingesetzt, das von sich aus bereits viele Anforderungen des vorangegangenen Abschnitts erfüllt.

### 6.7.1 Kern-System

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: mithilfe einer Routen-Konfiguration werden Controller für bestimmte Pfade in der Anfrage-URL registriert und dann bei Bedarf vom Framework automatisch instanziiert und aufgerufen. Das folgende Listing zeigt dies am Beispiel des `LoginController`s, 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() { ... }  
}
```

Die Controller schreiben ihre Antwort in ein Objekt, das vom Framework als Antwort gesendet wird.

Oberstes Prinzip bei der Entwicklung des Frameworks war die Modularisierung und die Wiederverwendbarkeit, so sind alle Komponenten ohne harte Abhängigkeiten mithilfe eines DI-Containers verbunden. Dadurch sind **Symfony2**-Anwendungen leicht zu warten und auch zu testen, entsprechende Unit-Testing-Komponenten werden bereits mitgeliefert.

### 6.7.2 API

Die API wurde RESTful implementiert und deckt alle Funktionen des Anwendungsservers ab. Sie folgt dem CRUD-Paradigma und unterstützt dementsprechend die jeweiligen Methoden: `POST` zum Erzeugen von Objekten (`CREATE`), `GET` zum Lesen von Objekten (`READ`), `PUT` zum Aktualisieren bestehender Objekte (`UPDATE`) und `DELETE` zum Löschen von Objekten. Generell werden Daten innerhalb der Anfragen als JSON-Objekte kodiert. URL-Parameter oder eine Kombination von beidem sind ebenfalls zulässig. Eine Aufstellung aller implementierten API-Endpunkte findet sich in Anhang [A.2](#) · S.V.

---

<sup>19</sup><http://symfony.com/>

Die ausgelieferten JSON-Objekte enthalten Zusatzinformationen zum Kontext und zu Relationen, wie in Abschnitt 6.4 · S.66 beschrieben.

```
{  "@context":"http://jsonld.retext.it/Container",
  "@id":"/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.

### 6.7.3 Persistenz

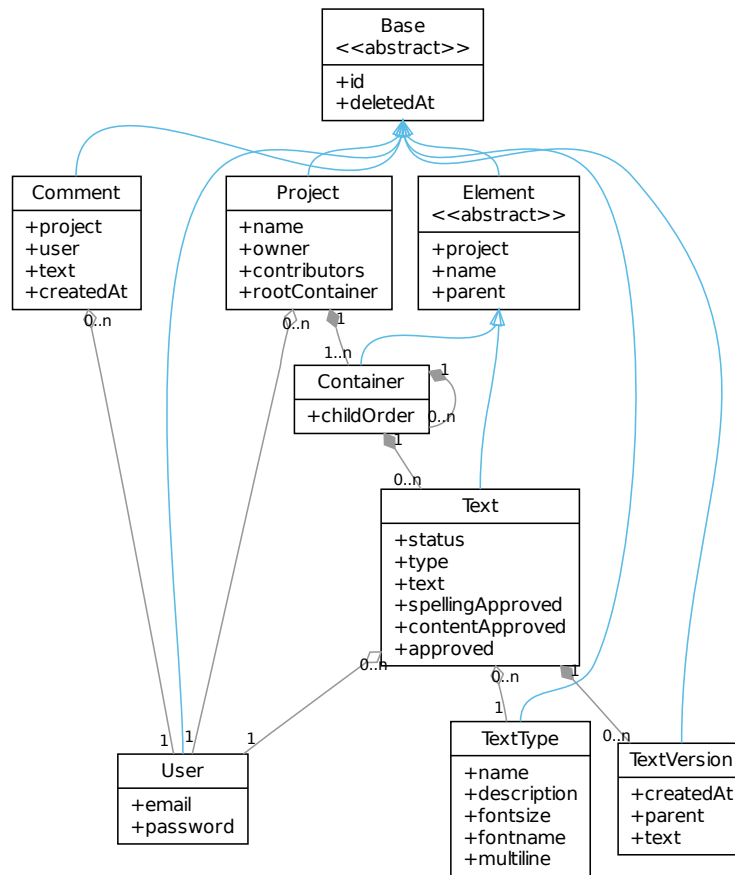


Abbildung 12: Persistierte Objekte im Prototyp

Zum Persistieren der Daten kommt **MongoDB**<sup>20</sup>, eine No-SQL-Datenbank, zum Einsatz. Diese ermöglicht das unkomplizierte Speichern auch komplexer Dokumentenstrukturen als JSON-Objekt. Die Wahl einer nicht-relationalen Datenbank bietet den Vorteil, dass aufwändiges Zusammensetzen und Zerlegen von Dokumenten entfallen kann. In der Datenbank werden die in Abbildung 12 gezeigten Domänenobjekte gespeichert. Dieses Modell ist deutlich einfacher als das in Abschnitt 6.2 · S.52 vorgestellte Domänenmodell, da im Prototyp nur wenige, entscheidende Funktionen implementiert wurden. Das Laden- und Speichern ist durch den Einsatz eines Object-Document-Mappers (ODM) für Doctrine 2 und MongoDB<sup>21</sup> weitestgehend automatisiert.

<sup>20</sup><http://www.mongodb.org/>

<sup>21</sup><https://github.com/doctrine/mongodb-odm>

## 6.7.4 Klassendiagramm

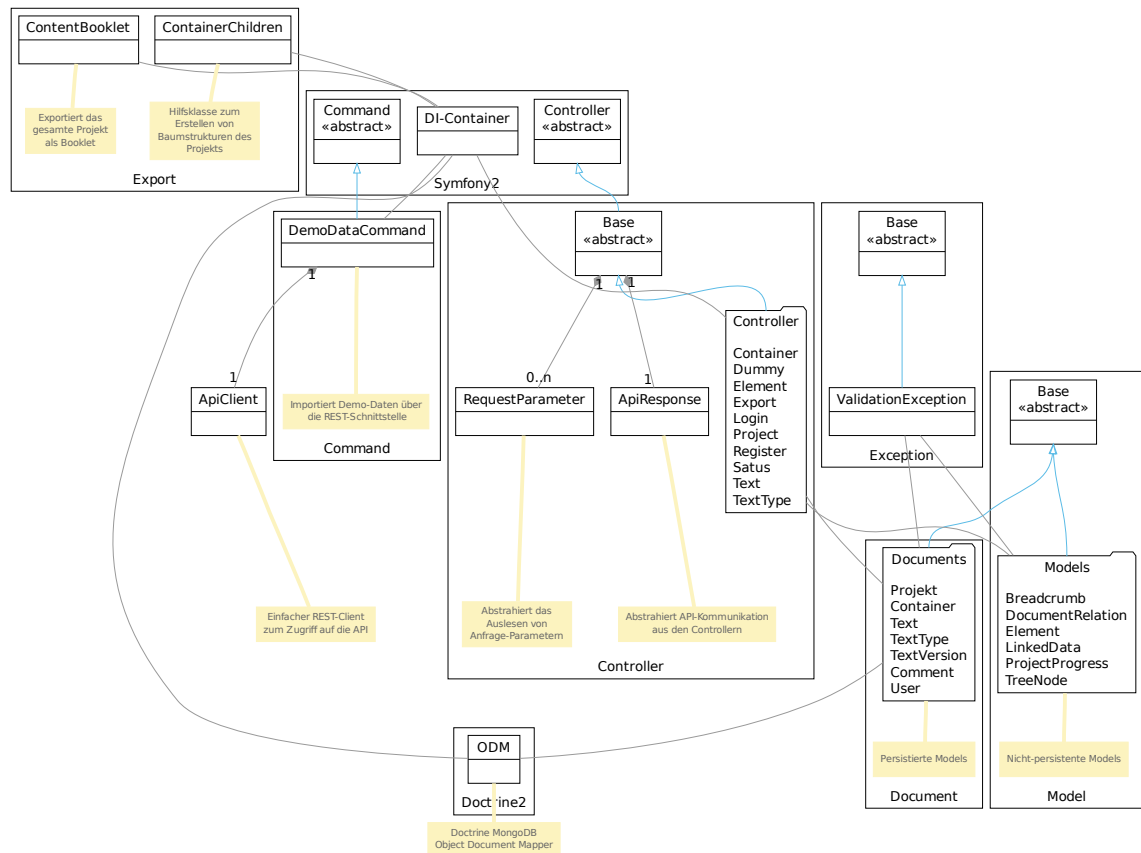


Abbildung 13: Anwendungsserver-Klassendiagramm des Prototyps

Abbildung 13 zeigt das Klassendiagramm der Implementierung des Prototyps. Zentrale Komponente ist der DI-Container des Symfony2-Frameworks. Dieser instanziiert je nach Pfad der Anfrage einen Controller und ruft dort die entsprechende Methode auf. 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 Repräsentation der Daten die Klassen des Packages Documents, die Controller können diese selber erzeugen oder bekommen diese als Anfrage-Ergebnisse vom ODM übergeben.

Funktionen, die an mehreren Stellen oder über unterschiedliche Arten verwendet werden, wurden in eigene Services ausgelagert (`ContentBooklet` und `ContainerChildren`). Diese werden innerhalb der Controller verwendet, können aber auch über das CLI gestartet werden.

Symfony2 stellt eine einfache Möglichkeit zum Erstellen von Operationen über die Kommandozeile zur Verfügung. Im Prototyp wurde das Anlegen von Demo-Projekt-Daten mithilfe dieser Möglichkeit realisiert. Das `DemoDataCommand` wird von der Konsole aus gestar-

tet und verwendet den `ApiClient`, um über HTTP mit der Schnittstelle des Anwendungsservers zu kommunizieren.



Die Implementierung eines Prototyps dient zur Verifizierung eines Entwurfs; im folgenden Abschnitt wird der Prototyp in einem realitätsnahen Szenario begutachtet.

## 6.8 Implementierter Workflow am Beispiel des Studiengangsflyers

Die in den vorangegangenen Abschnitten vorgestellte Implementierung des Prototyps 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 des Wintersemesters 2011/2012 hat 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.

Mit dem Prototyp ist es möglich, den Flyer als Projekt anzulegen. Im nächsten Schritt wird die Struktur des Flyers festgelegt. Die Anwendung unterscheidet dazu zwischen Containern und Textbausteinen. Container sind rein strukturelle Elemente, die wiederum weitere Container und Textbausteine enthalten können. Für den Flyer bietet es sich an, einzelne Kapitel (Titel, Vorwort, Kurzinfo, Termin & Öffnungszeiten, Personen, Orientierung) als Container auf oberster Ebene anzulegen und darunter jeweils weitere Unterteilungen nach Abschnitten vorzunehmen. Innerhalb dieser Abschnitte werden dann die einzelnen Textbausteine definiert. Zu den Textbausteinen können auch Zusatzinformationen zum Text-Klasse (Schriftart, Schriftgröße, ein- oder mehrzeilig) hinterlegt werden. Die Elemente lassen sich mittels Drag&Drop in der Reihenfolge anpassen.

Sobald die Bestandteile des Produkts angelegt wurden kann parallel bereits mit der Erstellung der Texte begonnen werden. Diese können in den jeweiligen Bausteinen je nach Klasse als ein- oder mehrzeiliger Text hinterlegt werden. Änderungen an den Inhalten werden gespeichert und sind als Änderungsverlauf abrufbar. Über eine Kommentarfunktion ist der Austausch über die Texte innerhalb der Anwendung möglich.

In der Ansicht zur Freigabe können die einzelnen Texte überprüft und abgenommen werden. Es wird zwischen den Status für Rechtschreibung, Inhalt und der allgemeinen Freigabe unterschieden. Bei Statusänderungen wird automatisch ein Kommentar erzeugt, der beim Ablehnen durch den Benutzer mit Zusatzinformationen versehen werden kann. Die einzelnen Status werden aggregiert und als Gesamt-Status für das gesamte Projekt sowie zur Übersicht und zum schnellen Zugriff in der Projektstruktur angezeigt.

Als Beispiel für die Verwendung der Projektdaten in externen Systemen lässt sich das Projekt als *Content-Booklet* im HTML- oder PDF-Format exportieren. Dieses Dokument enthält alle Information in strukturierter Form, so wie sie im System angelegt wurden und listet neben den Texten auch die jeweiligen Zusatzinformationen auf. Mithilfe des *Content-Booklet* können Produzenten einfach mit der neuesten Version der Texte für das Produkt versorgt werden und es liefert einen Überblick über alle Bestandteile des Produkts. Ein Ausschnitt aus diesem Booklet findet sich in Anhang [A.3](#) · S.VII.



---

Der Prototyp ist in der Lage, bereits mit diesem einfachen Funktionsumfang einen praktischen Mehrwert für ein Produkt wie den Studiengangsflyer zu bieten. Die Texte können gemeinsam und nachvollziehbar erfasst werden, der Export als *Content-Booklet* kann verschiedenen Mitarbeitern bei der Erstellung des fertigen Produkts konkrete Hilfestellung bieten.

## 6.9 Zusammenfassung

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

Der Entwurf wurde mithilfe einer prototypischen Implementierung, in der die wichtigsten Abläufe abgebildet werden, anhand eines realen Projekts überprüft. Dabei wurde gezeigt, dass sich die vorgeschlagenen Prinzipien in der Darstellung wie in der Implementierung mit aktuell verfügbaren Technologien problemlos umsetzen lassen. Zudem wurde gezeigt, dass das Konzept der Trennung zwischen Anwendungsserver und GUI keine negativen Auswirkungen auf die Verwendbarkeit einer Anwendung hat. Für die Realisierung weiterer Funktionen im Sinne des Entwurfes sind keine Hindernisse aufgetreten.



Der in diesem Kapitel vorgestellte Entwurf liefert damit die Basis für die mögliche Entwicklung einer vollwertigen Lösung und bietet für einzelne Bestandteile bereits mögliche Technologieempfehlungen.

## 7 Fazit

Diese Bachelor-Thesis hat sich mit der Fragestellung beschäftigt, wie es dazu kommt, dass trotz aller technischen Fortschritte im Bereich der Informationstechnologie bei der Organisation von Texten für Medienprodukte auf Arbeitsweisen zurückgegriffen wird, die inzwischen überholt sein sollten und wie eine bessere Lösung für diese Aufgaben aussehen könnte.

Es wurde gezeigt, dass die gebräuchlichen Werkzeuge, Microsoft Word und Excel, in der alltäglichen Arbeit in Agenturen zu vielerlei Problemen führen, sie aber verwendet werden, weil die Nutzer zum einen deren Gebrauch gewöhnt sind und zum anderen die Werkzeuge scheinbar über alle notwendigen Funktionen für diese Aufgabe verfügen. In einer ausführlichen Analyse wurde diese Annahme jedoch widerlegt und im Einzelnen gezeigt, welche problematischen Auswirkungen der Einsatz monolithischer Dateiformate und dezentraler Speicherung in den komplexen Abläufen in Zusammenhang mit der Erstellung von Medienprodukten haben.

Aufbauend auf dieser Erkenntnis und unter Zuhilfenahme von Personas, die auf Interviews mit zwölf Branchenexperten basieren, wurde eine Lösung konzipiert, die versucht, die genannten Probleme zu beseitigen und den Anforderungen der Personas zu genügen. Hierzu wurde ein zentraler Anwendungsserver vorgeschlagen, auf den mit spezialisierten, an die jeweiligen Bedürfnisse der Benutzer angepassten, GUIs zugegriffen wird. Mit deren Hilfe werden die Texte der Produkte definiert, geschrieben, korrigiert, kontrolliert, freigegeben und veröffentlicht.

Für die wichtigsten Bestandteile der Lösung, den Anwendungsserver und das browserbasierte GUI, wurde die konkrete Architektur entworfen und detaillierte Gestaltungsrichtlinien mithilfe von Wireframes festgelegt.

Zur Validierung des Entwurfs wurde schließlich ein Prototyp umgesetzt, der die wichtigsten Funktionen anhand eines Beispiel-Projekts implementiert. Die Implementierung zeigte, dass das Konzept funktioniert, der Entwurf realisierbar ist und bereits die prototypische Fassung konkreten Mehrwert bietet.





# A Anhang

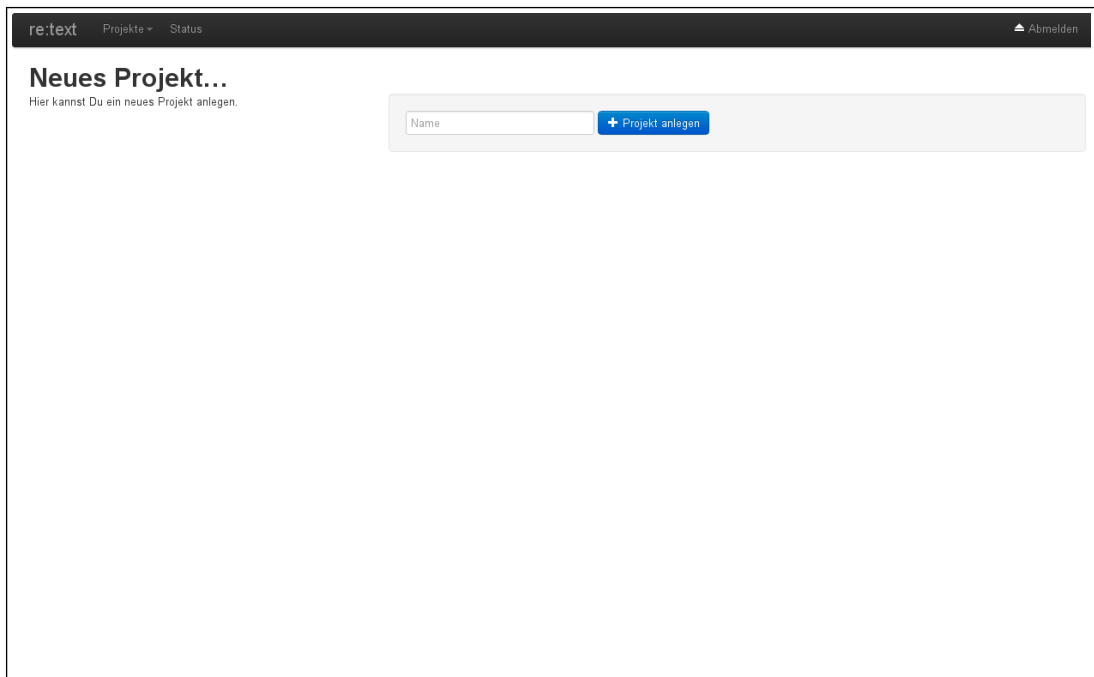
## A.1 Screenshots

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

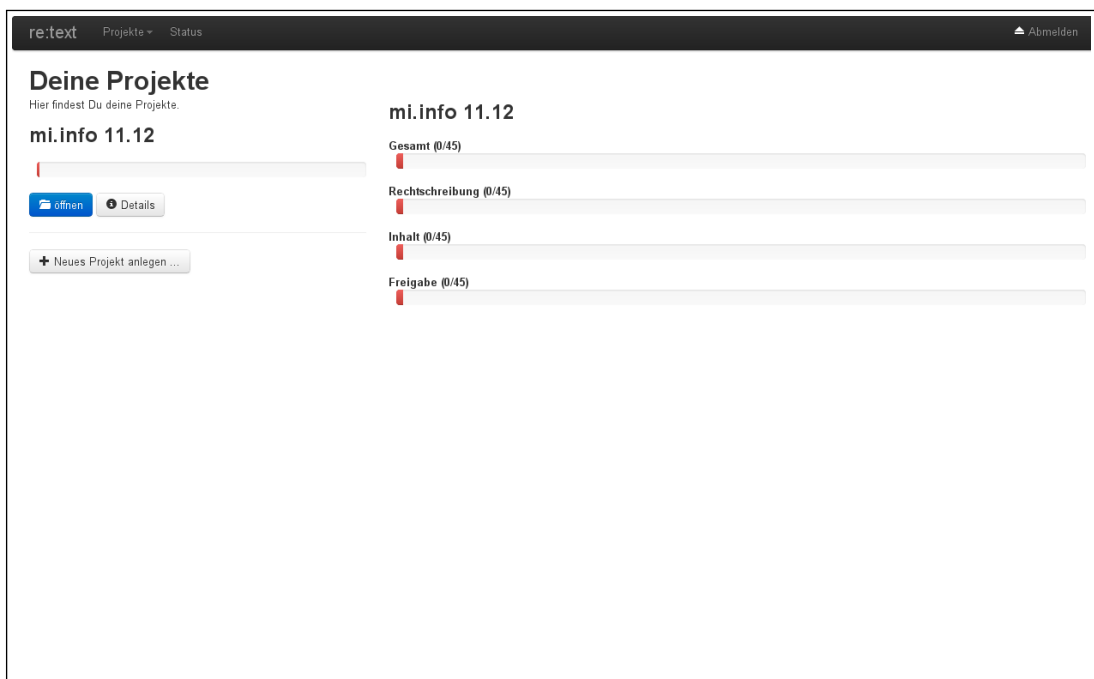
Screenshot 2: Registrierung für neue Benutzer

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

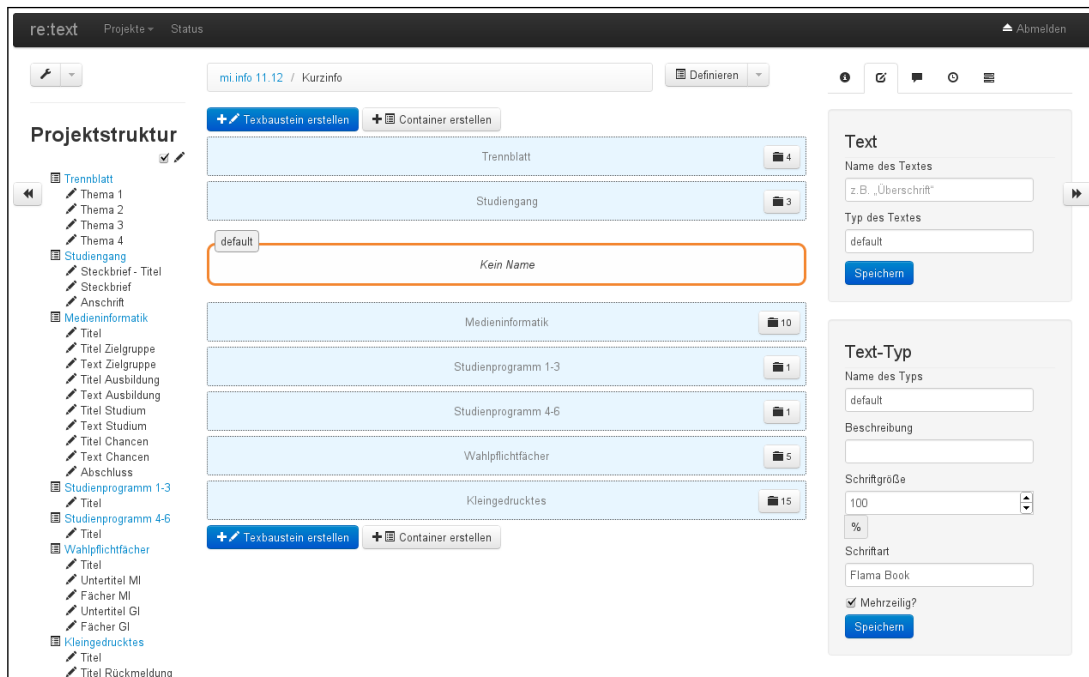
Screenshot 3: Login



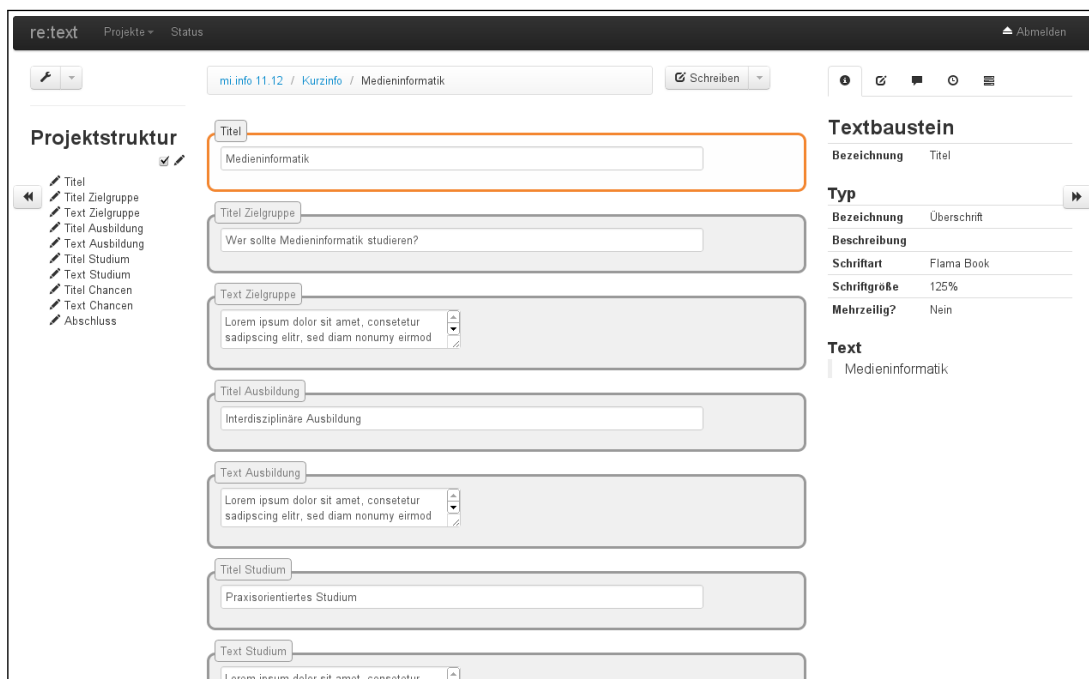
Screenshot 4: Projekt anlegen



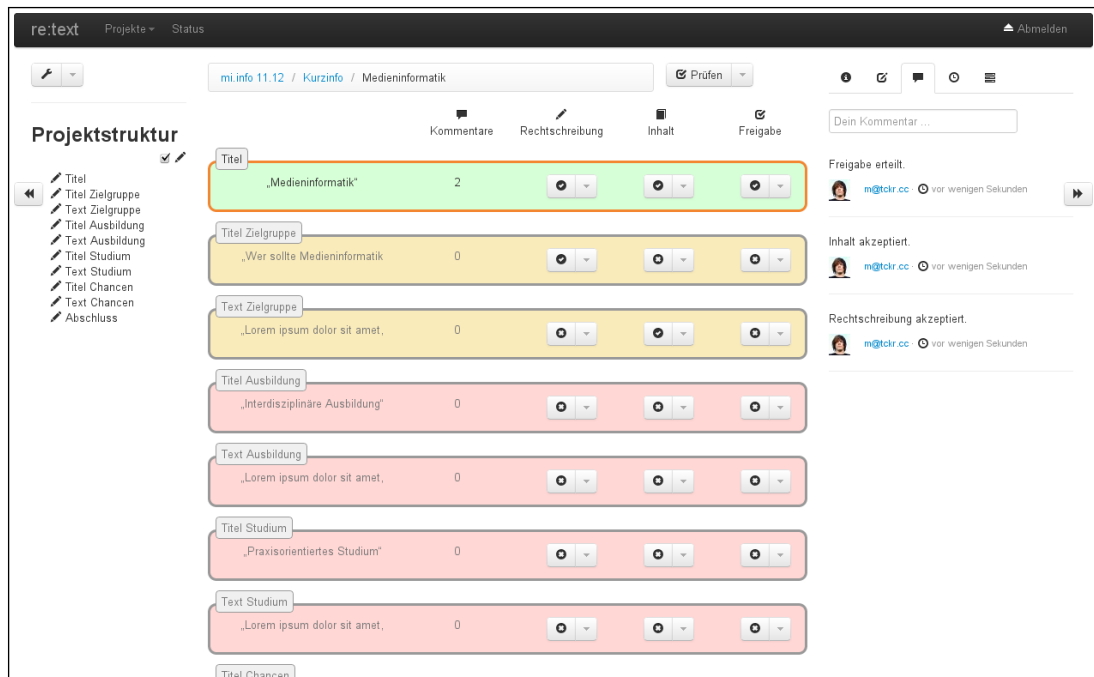
Screenshot 5: Projekt anzeigen, mit Darstellung des Projekt-Fortschritts



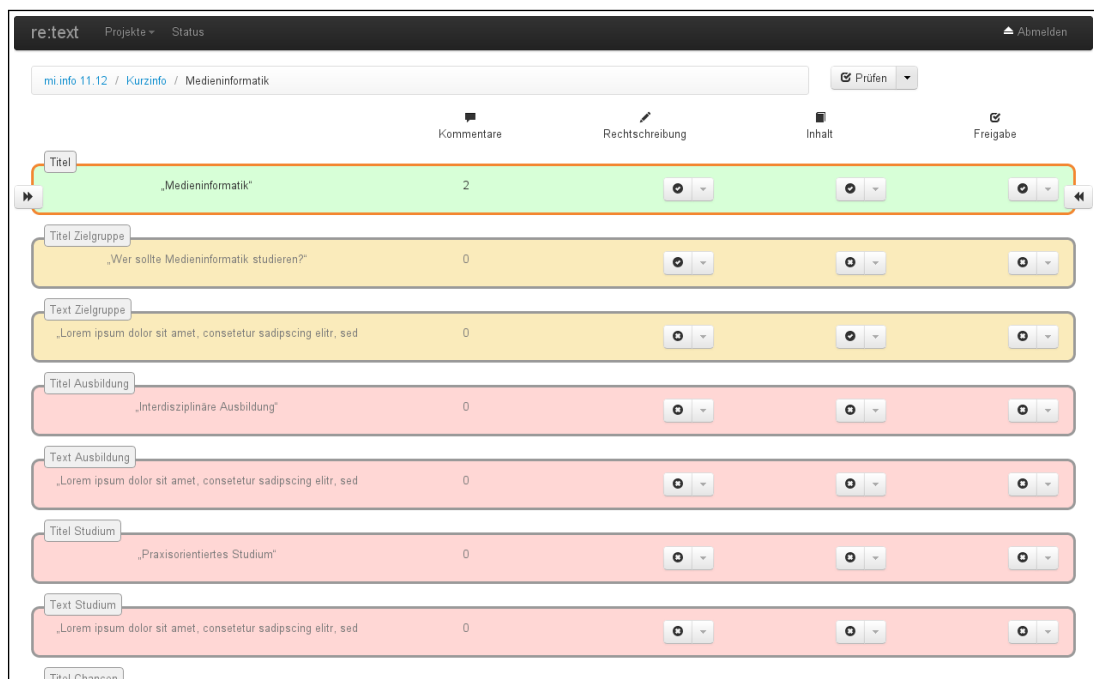
Screenshot 6: Produktstruktur definieren, vgl. 6.3.2 · S.59



Screenshot 7: Texte erstellen, vgl. 6.3.3 · S.61



Screenshot 8: Qualitätssicherung, vgl. 6.3.5 · S.63



Screenshot 9: Darstellung im Vollbild-Modus mit ausgeblendeten Seitenleisten, vgl. 6.3.1 · S.57



## A.2 API-Endpunkte

### A.2.1 Benutzer

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

### A.2.2 Projekt

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

### A.2.3 Elemente

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

### A.2.4 Container

<b>Methode</b>	<b>Pfad</b>	<b>Beschreibung</b>
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

### A.2.5 Texte

<b>Methode</b>	<b>Pfad</b>	<b>Beschreibung</b>
POST	/text	Text anlegen
GET	/text/<id>	Text laden
PUT	/text/<id>	Text aktualisieren
DELETE	/text/<id>	Text löschen
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

### A.2.6 Text-Typen

<b>Methode</b>	<b>Pfad</b>	<b>Beschreibung</b>
GET	/texttype	Liste mit Text-Typen eines Projekts laden
GET	/texttype/<id>	Text-Typ laden
PUT	/texttype/<id>	Text-Typ aktualisieren

### A.2.7 Export

<b>Methode</b>	<b>Pfad</b>	<b>Beschreibung</b>
GET	/export/contentbooklet.pdf	Content-Booklet eines Projekts als PDF exportieren
GET	/export/contentbooklet.html	Content-Booklet eines Projekts als HTML exportieren

## A.3 Content-Booklet

### Content-Booklet: mi.info 11.12

Datum: 19.06.2012

<b>Abschnitt: Titel</b>  ID 4fe03d2d820b901f1a000001 Pfad / Titel Unterelemente 2	
<b>Text: Titel</b>  ID 4fe03d2e820b90ce1a000004 ■ mi.info Typ Titel (Flama Medium, 200%, einzeilig)	
<b>Text: Untertitel</b>  ID 4fe03d2e820b901e1a000006 ■ 11.12 Typ Titel - Subline (Flama Book, 100%, einzeilig)	

<b>Abschnitt: Vorwort</b>  ID 4fe03d2d820b90ce1a000000 Pfad / Vorwort Unterelemente 4	
<b>Text: Überschrift</b>  ID 4fe03d2e820b901f1a000006 ■ Unendliche Weiten - wir schreiben das J Jahr 2011 Typ Überschrift (Flama Book, 125%, einzeilig)	
<b>Text: Text</b>  ID 4fe03d2e820b90ce1a000007 Typ default (Flama Book, 100%, mehrzeilig)	<p>»Computer, sag mir alles über mein Studium«, so oder so ähnlich hätte Captain Kirk sich sicherlich informiert, bevor er seine Crew in das Abenteuer der Medieninformatik geführt hätte. Alles nur ScienceFiction? Viele Dinge unserer täglichen Mediennutzung waren vor kurzem noch Entwürfe von Science-Fiction Romanen. Ob es der Begriff Cyberspace ist, der 1984 durch den Roman Neuromancer von William Gibson geprägt wurde, oder das iPad, das sein Vorbild im PADD genannten Device des Raumschiffs Enterprise hat, man findet viele Anleihen, die Schritt für Schritt in unser tägliches Leben Einzug gehalten haben.</p> <p>Wie wir in 50 Jahren mit Medien umgehen werden, kann sicherlich niemand voraussagen. Somit ist es wichtig, dass Sie neben den aktuellen Trends auch immer die dahinterliegenden Konzepte betrachten. Sie werden in Ihrem Studium daher neben den aktuellen Technologien viele dieser grundlegenden Konzepte kennen lernen: Wie kann ich Anwendungen strukturiert programmieren? Wie findet Kommunikation und Interaktion mit neuen Medien statt? Wie kann ich die Usability von Anwendungen verbessern? Wie werden Daten ausgetauscht und verwaltet? Wie kann ich virtuelle Welten erstellen und programmieren? Wie ...?</p> <p>Viele Fragestellungen, die Sie im Laufe Ihres Studiums klären können. Experimentieren Sie dabei auch mit unterschiedlichen Ansätzen und setzen Sie Technologien kreativ ein. Erst dadurch kann wieder etwas Neues entstehen.</p> <p>In diesem Sinne wünsche ich Ihnen im Namen des gesamten MI-Teams einen guten Start und eine spannende Reise durch die unendlichen Weiten der Medieninformatik.</p>
<b>Text: Signatur</b>  ID 4fe03d2f820b901e1a000009 ■ Prof. Dr. Jörg Berdux Typ kursiv (Flama Book, 100%, einzeilig)	
<b>Text: Signatur Untertitel</b>  ID 4fe03d2f820b901f1a000009 ■ Studiengangsleiter Medieninformatik Typ klein (Flama Book, 85%, einzeilig)	

Abbildung 14: Auszug aus dem Content-Booklet des Studiengangsflyers

## Literaturverzeichnis

- [1] D. Brown. *Communicating Design: Developing Web Site Documentation for Design and Planning*. Voices That Matter. New Riders, 2010.
- [2] M. Cohn. *User Stories Applied: For Agile Software Development*. The Addison-Wesley Signature Series. Addison-Wesley, 2004.
- [3] 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.
- [4] Roy T. Fielding and Richard N. Taylor. Principled design of the modern Web architecture. volume 2, pages 115–150, New York, NY, USA, May 2002. ACM.
- [5] S. Freeman and N. Pryce. *Growing Object-Oriented Software, Guided by Tests*. Addison-Wesley signature series. Addison Wesley, 2009.
- [6] J.T. Howerton. Service-Oriented Architecture and Web 2.0. *IT Professional*, 9(3):62–64, Mai-Juni 2007.
- [7] A. MacCaw. *JavaScript Web Applications*. O'Reilly Series. O'Reilly Media, 2011.
- [8] H. Schanze and S. Pütz. *Metzler Lexikon Medientheorie, Medienwissenschaft: Ansätze, Personen, Grundbegriffe*. Metzler, 2002.
- [9] Workflow Management Coalition. *Terminology & Glossary*, 3.0 edition, Februar 1999.

## Online-Quellen

- [10] Adobe Systems Incorporated. Professionelle Software für Redaktionen | Adobe InCopy CS5.5. <http://1.tckr.cc/HMQNo8> · Zugriff am 12. April 2012.
- [11] Sean Coates. PHP as a templating language. <http://1.tckr.cc/KG5ZSR> · Zugriff am 14. Mai 2012.
- [12] Microsoft Corporation. Zusammenarbeit an Dokumenten und gemeinsame Dokumenterstellung. <http://1.tckr.cc/Ja5QX0> · Zugriff am 17. April 2012.
- [13] Brett Hill. Check it out! Easily enable document version control in SharePoint Online. <http://1.tckr.cc/IUMZQu> · Zugriff am 13. April 2012.
- [14] Jakob Nielsen. 10 heuristics for user interface design. <http://1.tckr.cc/IWoaFd> · Zugriff am 10. Mai 2012.

- [15] Corporate Vice President of the Developer Division at Microsoft Corporation S. Somasegar. Key Software Development Trends. <http://1.tckr.cc/II1jNz> · Zugriff am 3. Mai 2012.
- [16] WoodWing. Wie funktioniert das? | WoodWing.com. <http://1.tckr.cc/HJMWTI> · Zugriff am 12. April 2012.

