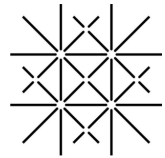




HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL



Fachhochschule Nordwestschweiz
Hochschule für Gestaltung und Kunst



UNI
BASEL

Thesis

Online-Styleguide

Master of Advanced Studies in Human Computer Interaction Design

Hochschule für Technik Rapperswil, Abteilung Informatik
Universität Basel, Fakultät für Psychologie
Hochschule für Gestaltung und Kunst

Stefano Morandi, Sébastien Niclasse und Chris Turtschi
Januar 2011

Coaches: Thomas Bircher und Toni Steimle

Abstract

Für Unternehmen ist die Motivation Styleguides für die Entwicklung von grafischen User-Interfaces zu erstellen und einzusetzen vielfältig. Durch den Einsatz von Styleguides lassen sich Entwicklungs- und Schulungskosten senken, die Kundenzufriedenheit erhöhen und den Bekanntheitsgrad einer Marke oder eines Produktes steigern. Kaum jemand, der sich mit User-Interfaces befasst, würde daher bezweifeln, dass Styleguides sinnvoll und nützlich sind. Jedoch werden in der Literatur diverse Gründe für ein „Scheitern“ von Styleguides angeführt. Einer davon ist, dass sie oft entweder nur in Papierform oder als elektronisches Dokument verfügbar sind, was deren Änderung oder Erweiterung erschwert. Ein Problem ist oft auch die mangelnde Akzeptanz des Styleguides seitens der Personen, die sie anwenden sollen, jedoch nicht in deren Erstellungsprozess involviert wurden. Im Kern bedeutet dies, dass mangelnde Kollaboration und Kommunikation von Erstellern und Anwendern von Styleguides ein Grund für das Scheitern sein können.

Diese Studie befasst sich mit der Theorie und Problematik von Styleguides, gibt Empfehlungen für deren Einsatz im Umfeld von Unternehmen und Projekten und zeigt ein Konzept für ein System zur kollaborativen Erstellung, Verwaltung und Verwendung projektspezifischer Styleguides, das anhand eines benutzerzentrierten Vorgehens erarbeitet wird.

Die Grundlage des Konzeptes bilden Recherchen in der Literatur zur Theorie und Problematik von Styleguides und Recherchen in der Praxis mittels Interviews mit Personen in verschiedenen Firmen, die in ihrer jeweiligen Arbeitsrolle mit Styleguides konfrontiert sind. Auf Basis der dabei gewonnen Daten, werden Personas bestimmt, Anforderungen definiert und diverse weitere Spezifikationen festgelegt. Daraus wird ein konzeptioneller Prototyp abgeleitet und evaluiert.

Danksagung

An dieser Stelle möchten wir uns ganz herzlich bei allen Testpersonen und Interviewpartnern folgender Firmen bedanken, die es uns mit ihrem Entgegenkommen und ihrem Wissen ermöglicht haben, einen tiefen Einblick in die Thematik zu gewinnen:

- Baloise, Basel
- Claudia, Basel
- Ginetta AG, Bern
- Netcetera, Zürich
- New Impact AG, Bern
- Roche Diagnostics Ltd., Rotkreuz
- SBB AG, Bern
- Siemens, Zürich
- SIX Swiss Exchange, Zürich
- Softcom, Fribourg
- Swisscom AG, Bern
- Sympany, Basel

Herzlichen Dank möchten wir unserem Coach Thomas Bircher aussprechen, der uns während der oft hektischen Zeit mit seinem Wissen und Erfahrung eine grosse Unterstützung war.

Ein grosses Dankeschön gilt auch den Firmen MST Systemtechnik AG und SBB AG für die Bereitstellung der Meetingräume und der überaus wichtigen Kaffeeinfrastruktur.

Je profite ici de remercier encore ma collègue de travail Marta pour ses conseils avisés en modélisation UML.

Schliesslich gilt unser grösster Dank unseren Partnerinnen, Familien und Bekannten, die uns während der gesamten Masterarbeit mit sehr viel Geduld und Rücksicht zur Seite standen, auch wenn häufig die Wochenenden für unsere Projektarbeit investiert wurden.

Inhaltsverzeichnis

Abstract	v
Danksagung	vii
Inhaltsverzeichnis	ix
Tabellen	xii
Abbildungen	xiii
1 Einleitung	1
1.1 Aufgabenstellung	2
1.2 Zielpublikum.....	3
1.3 Randbedingungen und Besonderheiten	3
1.4 Geheimhaltung und Rechte	3
1.5 Projektplanung.....	3
2 Von Menschen und Styleguides – Einführung ins Themengebiet	5
2.1 Human-Computer Interaction	6
2.2 Usability	7
2.3 Usability-Guidelines	8
2.3.1 Klassifizierungen von Usability-Guidelines	8
2.3.2 Schreibe Patterns – Keine Guidelines!	12
2.3.3 Beispiel einer guten Guideline.....	12
2.4 Styleguides	13
2.4.1 Konsistenz.....	13
2.4.2 Vorteile von Styleguides	14
2.4.3 Klassifizierungen von Styleguides	14
2.4.4 Inhalt von Styleguides	15
2.4.5 Styleguides im Software-Entwicklungsprozess.....	16
2.4.6 Projektspezifische Styleguides.....	18
3 Der Weg zum Konzept – Vorgehen und Methodologie	19
3.1 Chronologisches Vorgehen	19
3.1.1 Eintauchen in die Materie – Literaturrecherchen	20
3.1.2 Recherche über Benutzer und Aufgaben – Interviews.....	20
3.1.3 Verständnisaufbau der Benutzer und Aufgaben – Personas und Szenarios	21
3.1.4 Bändigung der Informationsflut und Ideen – Concept Model.....	21
3.1.5 Visualisierung von Ideen – Varianten und Prototypen.....	21
3.1.6 Analyse des Geschäftskontextes – Business Use-Cases	22
3.1.7 Formalisierung der Anforderungen – Features und Requirements	22
3.1.8 Formalisierung der Anwendungsfälle – Use-Cases und Aktoren.....	22
3.1.9 Evaluation bestehender Lösungen – Konkurrenzanalyse.....	23
3.1.10 Detailspezifikation der Applikation – Strukturdiagramm und Zustandsdiagramm	23
3.1.11 Evaluation des Konzeptes – Usability Testing.....	23
3.2 Ethische Aspekte des Vorgehens	24

4	Die Suche nach Erkenntnissen – Recherchen	25
4.1	Recherchen in der Literatur	26
4.1.1	Klassifizierung der Probleme und Lösungen in Themenbereiche.....	27
4.1.2	Themenbereich: Allgemeines	28
4.1.3	Themenbereich: Inhalt.....	28
4.1.4	Themenbereich: Struktur und Form	30
4.1.5	Themenbereich: Prozesse und Kollaboration	32
4.2	Recherchen in der Praxis	35
4.2.1	Analyse der Benutzer	35
4.2.2	Analyse der Prozesse	40
4.2.3	Fazits der empirischen Analyse.....	43
5	Die Zusammenfassung der Erkenntnisse – Aspekte, Features und Konkurrenzanalyse	47
5.1	Aspekte.....	48
5.1.1	Unterstützung der Kollaboration	48
5.1.2	Wiederverwendbarkeit von Inhalten.....	48
5.1.3	Inhalts- und Change-Management	49
5.1.4	Import und Export von Inhalten	49
5.1.5	Benachrichtigung der Benutzer	50
5.1.6	Flexibles Workflowmanagement	50
5.1.7	Sicherheit, Projekt- und Benutzerverwaltung.....	50
5.1.8	Visualisierung und Suche	51
5.1.9	Interoperabilität mit externen Applikationen	51
5.2	Features.....	52
5.3	Konkurrenzanalyse.....	54
6	Das Giessen des Fundamentes – Spezifikationen	57
6.1	Concept Model	58
6.1.1	Organisation und Struktur	59
6.1.2	Prozesse und Kollaboration	60
6.1.3	Visualisierung und Suche	61
6.2	Use-Case Spezifikation	62
6.2.1	Aktoren	62
6.2.2	Berechtigungen.....	62
6.2.3	Use-Case Model	63
6.3	Strukturdiagramm	64
6.3.1	Struktur eines Styleguides	64
6.3.2	Wiederverwendbarkeit, Versionierung und Metadaten	64
6.4	Zustandsdiagramm.....	65
6.4.1	Flexibilität	65
6.4.2	Zustände und Transitionen	66
6.4.3	Rollen und Berechtigungen	66

7	Vom Fundament zur Lösung – Konzept und Prototyp	67
7.1	Konzept „Online-Styleguide“	68
7.2	Konzeptioneller Prototyp	69
7.2.1	Grundlayout	69
7.2.2	Header	70
7.2.3	Ansichten <i>Styleguide</i> , <i>Suchresultate</i> und <i>Dashboard</i>	71
7.2.4	Gestaltungskonzept	77
7.2.5	Verwendete Patterns	78
8	Der Prototyp im Test – Evaluation	79
8.1	Methode	79
8.2	Umsetzung	79
8.3	Findings	80
9	Der Einsatz projektspezifischer Styleguides – Guidelines	83
10	Schlussfolgerungen	85
10.1	Schwierigkeiten und Limitationen	85
10.2	Reflexion der Methodologie	86
10.3	Diskussion der Ergebnisse	86
10.4	Weitere Schritte	87
10.5	Projektablauf und persönliche Fazits	88
	Anhänge	91
	Anhang A: Projektplanung	91
	Anhang B: Personas	91
	Anhang C: Business Use-Cases	91
	Anhang D: Features und Requirements	91
	Anhang E: Konkurrenzanalyse	91
	Anhang F: Concept Model	91
	Anhang G: Strukturdiagramm	91
	Anhang H: Zustandsdiagramm	91
	Anhang I: Use-Case Spezifikationen	91
	Anhang J: Screenshots Prototyp	91
	Anhang K: Walkthrough-Szenarien	91
	Anhang L: Walkthrough-Analyse	91
	Anhang M: Hintergrund zu Concept Models	91
	Anhang N: Glossar	91
	Literatur und Quellen	93

Tabellen

Tabelle 1: Arbeitshypothesen.....	2
Tabelle 2: Scope der Masterarbeit	2
Tabelle 3: Die wichtigsten Tätigkeiten aufgeteilt auf die Inception und Elaboration Phasen	4
Tabelle 4: Klassifizierung von Usability Guidelines (Dix, et al., 2004b)	10
Tabelle 5: Bezugspunkte der Konsistenz (Wilson, 2001).....	13
Tabelle 6: Vorteile von Styleguides (Gale, 1996).....	14
Tabelle 7: Klassifizierungen von Styleguides	14
Tabelle 8: Inhalt projektspezifischer Styleguides (Richter, et al., 2007 S. 53).....	15
Tabelle 9: Usability-Ziele projektspezifischer Styleguides (Quesenbery, 2001a).....	18
Tabelle 10: Tableau des activités réalisées.....	19
Tabelle 11: Répartition des interviews par Personas, taille et secteur d'activité des entreprises.....	20
Tabelle 12: Ethische Richtlinien im Vorgehen	24
Tabelle 13: Klassifizierung der Probleme und Lösungen in Themenbereiche.....	27
Tabelle 14: Aspects des interviews et priorités par rapport aux Personas	35
Tabelle 15: Description des variables comportementales (Behavior Variables).....	37
Tabelle 16: Types de méthodologie utilisées (résumé de la recherche empirique)	43
Tabelle 17: Types de recherche utilisées (résumé de la recherche empirique).....	45
Tabelle 18: Types de format des SG (résumé de la recherche empirique)	46
Tabelle 19: Récapitulatif des découvertes de la recherche empirique	46
Tabelle 20: Features des Systems.....	53
Tabelle 21: Liste des produits concurrents du Benchmarking	54
Tabelle 22: Description des System Actors	62
Tabelle 23: Gestion des permissions	62
Tabelle 24: Verschiedene Portlets für verschiedene Rollen.....	75
Tabelle 25: Verwendete Patterns.....	78
Tabelle 26: Ergebnisse zum Dashboard.....	81
Tabelle 27: Ergebnisse zur Verzeichnisstruktur in der linken Spalte.....	81
Tabelle 28: Ergebnisse zu einem Artikel.....	82
Tabelle 29: Ergebnisse zur Suchfunktion.....	82
Tabelle 30: Allgemeine Ergebnisse	82
Tabelle 31: Positive und negative Kriterien.....	82
Tabelle 32: Guidelines zum Prozess	83
Tabelle 33: Guidelines zur Struktur und Form.....	83
Tabelle 34: Guidelines zum Inhalt	84

Abbildungen

Abb. 1: Projektplanung anhand des RUP-Vorgehensmodells	4
Abb. 2: Einbettung von Styleguides im Kontext von HCI	5
Abb. 3: Usability im Kontext (Hübscher, 2008 S. 7)	7
Abb. 4: Klassifizierung von Usability-Guidelines (Mariage, et al., 2004).....	8
Abb. 5: The Bull's-Eye: A Framework for Web Application UI Design Guidelines (Beier, et al., 2003) .	10
Abb. 6: Interpretations- und Abstraktionsgrad nach Quellen (Mariage, et al., 2004)	11
Abb. 7: Interpretations- und Umsetzungsgrad nach Quellen (Mariage, et al., 2004).....	11
Abb. 8: Beispiel des Aufbaus einer Guideline (Usability.gov, 2003)	12
Abb. 9: Différences entre méthodologies, inspiré du cours „Vorgehensmodelle“ (Steimle, 2008)	17
Abb. 10: Benutzergruppen von Styleguides (Quesenbery, 2001a)	18
Abb. 11: Déroulement des activités dans le temps, inspiré du cours de Design (Bircher, 2009 S. 32) .	19
Abb. 12: Étapes du prototypage et du design, inspiré du cours de Design (Bircher, 2009)	21
Abb. 13: Ansprüche verschiedener Benutzergruppen an Styleguides (Quesenbery, 2001a)	31
Abb. 14: Analyse des Personas (Behavior Variable Map)	36
Abb. 15: Mapping des hypothèses et des résultats des Personas	38
Abb. 16: Vue d'ensemble des Business Use-Cases.....	40
Abb. 17: Create styleguide Business Use-Case	41
Abb. 18: Activité d'écriture des styleguides dans les méthodologies.....	42
Abb. 19: Activité de lecture des styleguides dans les méthodologies	42
Abb. 20: Résultats de l'analyse concurrentielle (Benchmarking).....	55
Abb. 21: Concept Model des Systems.....	58
Abb. 22: Concept Model - Organisation.....	59
Abb. 23: Concept Model - Struktur eines Styleguide	59
Abb. 24: Concept Model - Artikel und Tags	59
Abb. 25: Concept Model - Artikel, Kommentare und Wahlstimmen	59
Abb. 26: Concept Model - Styleguide, Artikel, Elemente und Versionen	59
Abb. 27: Concept Model - Aufträge	60
Abb. 28: Concept Model - Prozesse	60
Abb. 29: Concept Model - Version und Prozesse	60
Abb. 30: Concept Model - Änderungen	60
Abb. 31: Concept Model - Portlets und Dashboard	61
Abb. 32: Concept Model - Suchen und Filtern	61
Abb. 33: System Actors	62
Abb. 34: Use-Case Model (<i>User, Coordinator, Realizer</i>)	63
Abb. 35: Struktur eines Styleguide	64
Abb. 36: Zustandsdiagramm eines Styleguides.....	65
Abb. 37: Das Grundlayout zur Systemansicht „Styleguide“ und „Liste der Suchresultate“	69
Abb. 38: Das Speziallayout für die Systemansicht „Dashboard“	70
Abb. 39: Der Header-Bereich	70
Abb. 40: Die Styleguideansicht.....	71
Abb. 41: Metadaten in einer Lightbox	72

Abb. 42: Liste der Suchresultate.....	73
Abb. 46: Dashboardansicht	74
Abb. 43: Projektfilter	74
Abb. 44: Styleguidefilter	74
Abb. 45: Tagfilter	74
Abb. 47: Portlet „My Bookmarks“	76
Abb. 48: Portlet „My Watchlist“	76
Abb. 49: Portlet „Recently updated“	76
Abb. 50: Portlet „My Tasks“	76
Abb. 51: Portlet „My Requests“	76
Abb. 52: Portlet-Funktionen „close“, „minimize & maximize“ und „Lightbox“	77
Abb. 53: Lightbox Ansicht eines Portlets	77
Abb. 54: Déroulement du projet	88

1 Einleitung

Styleguides sind in der heutigen Zeit ein wichtiges Instrument um Gewährleistung für die Konsistenz einer Marke, eines Produkt oder ganzen Produktpaletten zu bieten und dadurch ein Bild von Geschlossenheit vermitteln. Gerade in der Web- und Softwareentwicklung, in der auch der Fokus dieser Masterarbeit liegt, ist es von Bedeutung, dass Styleguides konsequent angewendet werden um Inkonsistenzen, beispielsweise innerhalb von Softwareprogrammen, zu vermeiden. Es wird festgestellt, dass die Anwendung von Styleguides keine Selbstverständlichkeit ist und allzu oft, in Folge mangelnder Akzeptanz innerhalb der Fachgruppen der Web- und Softwareentwicklung, zum Scheitern verurteilt ist. Es können viele Gründe dafür verantwortlich gemacht werden, angefangen bei der fehlenden Kommunikation und der mangelnden Sensibilisierung seitens des Managements bezüglich des Nutzens von Styleguides, bis hin zur fehlenden Einbindung aller involvierten und betroffenen Personen in den Entstehungsprozess von Styleguides.

Zielsetzung

Ziel dieser Masterarbeit ist es, einerseits die Theorie über Styleguides zu ergründen und andererseits ein Konzept eines Systems zur kollaborativen Erstellung, Verwaltung und Verwendung von Styleguides zu erstellen. Ausserdem werden Empfehlungen zum Einsatz von Styleguides erarbeitet.

Aufbau der Thesis

Die Thesis gliedert sich in folgende Kapitel: Kapitel 1 behandelt die Aufgabenstellung, die Zielsetzung, den Scope und die Planung des Projektes. Kapitel 2 widmet sich der Einführung in die Thematik von Styleguides und deren Kontext. Das Vorgehen und die während des Projektes eingesetzten Techniken werden in Kapitel 3 erklärt. Die Recherchen in der Literatur und der Praxis zur Thematik von Styleguides ist Gegenstand von Kapitel 4. Die Erkenntnisse aus den Recherchen werden in Kapitel 5 zusammengefasst. Die daraus resultierenden Spezifikationen für das Konzept bilden den Schwerpunkt von Kapitel 6. In Kapitel 7 wird das Konzept auf Basis des konzeptionellen Prototyps erklärt. Die Evaluation des Prototyps bildet den Inhalt von Kapitel 8. Die Empfehlungen, welche das Projektteam zu projektspezifischen Styleguides in Bezug auf den Prozess, die Struktur und den Inhalt abgeben kann, ist Bestandteil von Kapitel 9. In Kapitel 10 werden die Schlussfolgerungen aus dem Vorgehen und dem Nutzen der Arbeit gezogen. Die persönlichen Fazits der Projektteilnehmer bildet den Abschluss dieses Kapitel.

Artefakte

Projektplan, Benutzerbeschreibungen, detaillierte Modelle, Spezifikationen, Glossar und weitere spezifische Artefakte sind in den entsprechenden Anhängen untergebracht.

1.1 Aufgabenstellung

Beim Einsatz von Styleguides ergeben sich, laut diversen Quellen in der Literatur, immer wieder Probleme die zu einem Scheitern von Styleguides führen können. Diese Probleme und die Tatsache, dass uns bislang kein System zum kollaborativen Styleguide-Management bekannt ist, haben uns dazu bewogen mit der Masterarbeit hier anzusetzen. Basis dafür bilden folgende Fragen:

- Was genau sind eigentlich Styleguides? Weshalb gibt es sie? Was ist ihr Nutzen? Wie werden sie eingesetzt? Welche Probleme gibt es bezüglich Styleguides?
- Lässt sich ein System konzipieren, das eine Kollaboration verschiedener, von projektspezifischen Styleguides betroffenen Personen erlaubt? Und wie könnte ein User-Interface für ein solches System aussehen?
- Lassen sich typische Benutzergruppen, sogenannte Personas, von Styleguides definieren? Welche sind es und welche Anforderungen und Bedürfnisse haben sie?
- Kann man eine Struktur definieren, die es erlaubt, die möglichen Inhalte von Styleguides wiederverwendbar zu machen?
- Lässt sich ein „generisches“ System konzipieren? Das heisst, ein System, das die Möglichkeit bietet, verschiedene Arten von Styleguides mit unterschiedlichem Inhalt und unter Berücksichtigung verschiedener Firmen- und Entwicklungsprozesse in denen sie erstellt und eingesetzt werden, zu verwalten und darzustellen?

Ausgehend von den Fragestellungen haben wir Hypothesen abgeleitet, die im weiteren Verlauf der Masterarbeit validiert werden sollen. Die Validierung erfolgt qualitativ und nicht quantitativ:

Hypothesen	Referenz
Es gibt drei Personas: Owner, Writer und User	HYP-1
Personen wollen mit einem kollaborativen Styleguide-Management-System arbeiten.	HYP-2
Styleguides oder Elemente davon können wiederverwendet werden.	HYP-3
Es ist möglich ein „generisches“ System für Styleguides zu entwerfen.	HYP-4

Tabelle 1: Arbeitshypothesen

Scope der Masterarbeit

Die nachfolgende Tabelle zeigt den Scope, das heisst, die inhaltliche Abgrenzung der Masterarbeit:

In Scope	Out of Scope
Recherchen zur Thematik von Styleguides im Web- und Softwareentwicklungsbereich	Analyse und Auswertung aller Arten und Typen von Usability-Guidelines
Ausarbeitung eines Konzepts für ein kollaboratives System zur Erstellung, Verwaltung und Benutzung von Styleguides	Funktionale Anforderungen aus Sicht der Ersteller und Auftraggeber von Styleguides
Identifikation der Benutzergruppen (Personas), die mit dem System interagieren	Umfragen und Statistiken zum Thema Styleguide
Erhebung der Anforderungen an das System	Technische Umsetzung des konzeptionellen Prototyps.
Konzeptioneller Prototyp aus Sicht eines Benutzers von Styleguides und beschränkt auf die wichtigsten Funktionalitäten (Anzeigen und Suchen von Inhalten)	
Empfehlungen zum Einsatz von Styleguides	

Tabelle 2: Scope der Masterarbeit

1.2 Zielpublikum

Diese Masterarbeit richtet sich hauptsächlich an folgende Gruppen:

- Personen, die sich mit der Definition oder Erstellung von Inhalten projektspezifischer Styleguides im Bereich der Web- und Softwareentwicklung verantwortlich zeigen.
- Lehrkräfte und Studenten, die sich vertieft mit der Thematik von HCI, Usability-Guidelines und Styleguides befassen.

1.3 Randbedingungen und Besonderheiten

Das Projektteam setzt sich aus Personen unterschiedlicher Muttersprachen zusammen. In Absprache mit dem Projektbetreuer sind daher Teile des vorliegenden Berichtes und einige Anhänge in französischer Sprache abgefasst.

Eine weitere Besonderheit ist, dass diese Masterarbeit eine reine Studienarbeit darstellt und kein eigentlicher Auftraggeber in Form eines Wirtschaftspartners dahintersteht. Es wird daher auf ein *Management Summary* verzichtet.

1.4 Geheimhaltung und Rechte

Die Ergebnisse dieser Masterarbeit sind frei verfügbar. Eine Ausnahme bilden sämtliche Artefakte aus denen Rückschlüsse auf spezifische Personen oder Firmen gezogen werden können.

1.5 Projektplanung

Das Vorgehensmodell für diese Masterarbeit bildet der *Rational Unified Process* (RUP, 2003). Die Hauptgründe dafür sind:

- **Iterative und inkrementelle Entwicklung**
Nach jeder Iteration ist es möglich, den Fortschritt des Projekts zu überprüfen und, falls erforderlich, Massnahmen zu ergreifen. Die Ergebnisse einer Iteration können in einer nächsten Iteration inkrementell erweitert und ergänzt werden.
- **UML-Notation**
Im softwaretechnischen Bereich ist UML eine gängige Notation, mit denen diverse Modelle, beispielsweise Business Use-Cases, Use-Cases, Aktoren, Klassen- und Zustandsdiagramme, erstellt und beschrieben werden können.
- **Flexibilität des Frameworks**
Mit einer individuellen Anpassung an ein Projekt wird ermöglicht, die Dauer der Phasen und die zu erstellenden Artefakte selber zu bestimmen und an diese Masterarbeit anzupassen.

Um die Benutzer des Systems besser zu verstehen, wird das Vorgehen nach RUP mit UCD-Techniken *Interview*, *Personas* und *Usability-Walkthrough* ergänzt.

Nachfolgende Grafik zeigt die einzelnen Phasen und Iterationen die während des Projektes durchlaufen werden. Der detaillierte Projektplan ist in *Anhang A: Projektplanung* ersichtlich.

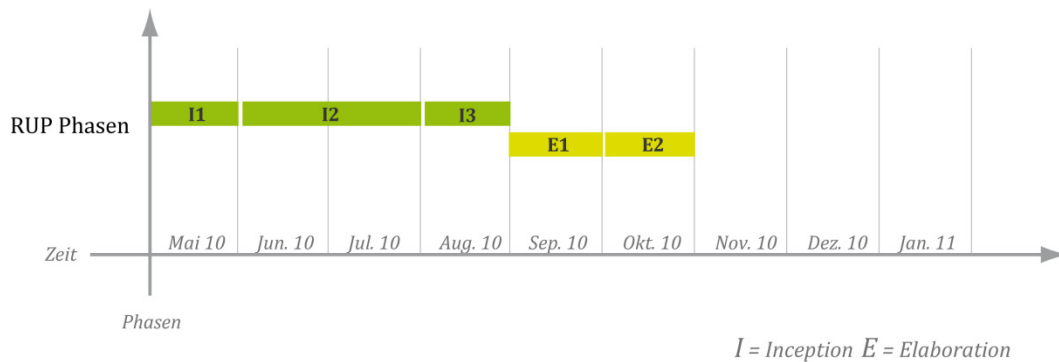


Abb. 1: Projektplanung anhand des RUP-Vorgehensmodells

Die nachfolgende Tabelle zeigt die wichtigsten Tätigkeiten und die daraus resultierenden Ergebnisse der einzelnen Phasen des Projektablaufs:

Phase	Tätigkeiten	Ergebnisse
Inception 1	Literaturrecherchen, Projektplanung und Definition der Hypothesen	Projektplan, Addendum, Scope, Vorbereitung der Arbeitsartefakte
Inception 2	Durchführung der Interviews, Erstellung: Personas, Business Use-Case Model und Concept Model	Personas, Verständnis für Problemdomäne anhand des Concept Model, Business Use-Case Model
Inception 3	Features und Requirements definieren, Use-Case definieren, Concept Model fertigstellen	Features und Requirements, Use-Case Spezifikationen, Concept Model, Zustandsdiagramm
Elaboration 1	Spezifikationen, Prototyp erstellen, Usability Walkthrough durchführen	Strukturdiagramm, Statusdiagramm, low-fidelity Prototyp, Usability-Szenarien
Elaboration 2	Interaktiver Prototyp erstellen	Interaktiver Prototyp (wurde aus Zeitgründen nicht mehr gemacht)

Tabelle 3: Die wichtigsten Tätigkeiten aufgeteilt auf die Inception und Elaboration Phasen

Zur Besprechung und Überprüfung des Projektstatus und zur Abstimmung und Koordination der nächsten Schritte werden monatliche Coach-Meetings durchgeführt.

2 Von Menschen und Styleguides – Einführung ins Themengebiet

Unsere Masterarbeit befasst sich mit der Konzeption eines Systems zur kollaborativen Erstellung, Verwaltung und Benutzung von Styleguides für Projekte innerhalb eines Unternehmens. Um jedoch zu begreifen was Styleguides sind, weshalb sie existieren und was deren Sinn und Zweck ist, muss der Kontext in dem Styleguides eingebettet sind näher erläutert werden.

Dieses Kapitel befasst sich daher nicht nur mit Styleguides selbst, sondern ebenfalls mit dem Umfeld, aus denen Styleguides hervorgehen.

Styleguides im Kontext von Human-Computer Interaction (HCI)

Die Einbettung von Styleguides im Kontext veranschaulicht folgende Abbildung:

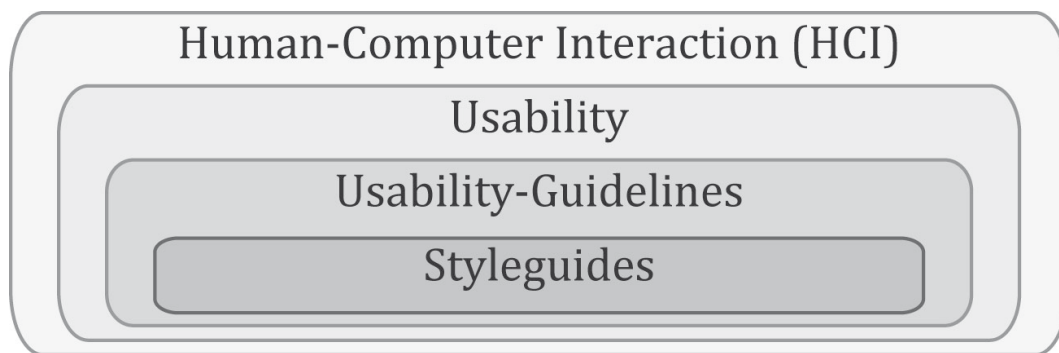


Abb. 2: Einbettung von Styleguides im Kontext von HCI

HCI ist ein interdisziplinäres Studienfeld, das sich mit der Interaktion zwischen Menschen und Computern befasst. Das Ziel von HCI ist die Verbesserung dieser Interaktion durch die Steigerung der Benutzbarkeit von Computern und deren Anpassung an die Bedürfnisse der Benutzer (Wikipedia, 2010).

In HCI eingebettet ist die **Usability**. Sie ist ein Schlüsselkonzept von HCI und setzt sich mit der Benutzbarkeit von Systemen bei der Durchführung menschlicher Tätigkeiten mit Hilfe von Computern auseinander. Der ISO-Standard 9241-11 (ISO, 1998) definiert Usability als den Grad, für welche bestimmte Benutzer ein bestimmtes Ziel innerhalb eines bestimmten Nutzungskontextes effektiv, effizient und zufriedenstellend erreichen.

Das Wissen, welches im Gebiet der Usability existiert, wird typischerweise in Form von **Usability-Guidelines** festgehalten. Usability-Guidelines sind ein „Ausdruck“ dieses Wissens in Form von Prinzipien, Richtlinien und/oder Empfehlungen (Mariage, et al., 2004).

Styleguides wiederum bilden eine Sammlung von Usability-Guidelines und/oder funktionalen oder nicht-funktionalen Anforderungen zur Sicherstellung der Konsistenz des Interfaces einer Applikation oder einer Familie von Applikationen für einen bestimmten Kontext (Vanderdonckt, 1999).

2.1 Human-Computer Interaction

Die Kurzform des englischen Begriffes *Human-Computer Interaction* lautet HCI. Bisweilen werden, alternativ dazu, andere englische Begriffe wie *Man-Machine Interaction* (MMI) oder *Computer-Human Interaction* (CHI) verwendet. Die deutschsprachigen Entsprechungen dazu lauten *Mensch-Computer Interaktion* oder *Mensch-Maschine Interaktion*. In der Thesis wird der Begriff HCI benutzt.

Geschichte

Der Begriff HCI findet erst seit den 80er Jahren weite Verbreitung. Die Wurzeln von HCI reichen jedoch weiter zurück. Ernsthafte und systematische Studien über die menschliche Arbeitsleistung in Fabriken wurden, mit Fokus auf manuelle Tätigkeiten, seit Beginn des letzten Jahrhunderts durchgeführt. Aber erst der zweite Weltkrieg, in Bezug auf die Produktion effizienterer Waffensysteme, steigerte das Interesse an der Interaktion zwischen Menschen und Maschinen. Dieses Interesse führte 1949 zur Gründung der *Ergonomics Research Society*. Das Studienfeld der Ergonomie, auch *Human Factors* genannt, befasst sich mit dem Benutzerverhalten in Bezug auf jegliches System, ob mechanisch, manuell oder computerbasiert. Durch die Verbreitung von Computern spezialisierten sich Forscher auf das Studium der Interaktion zwischen Menschen und Computern. Sie fokussierten sich dabei auf physiologische, psychologische und theoretische Aspekte dieses Prozesses. Daraus entstand der Begriff *Human-Computer Interaction* (Dix, et al., 2004a S. 1-8).

Interdisziplinarität

HCI ist ein interdisziplinäres Studienfeld und umfasst daher verschiedene Wissensbereiche. Es gibt allerdings keine abschliessende Liste aller Wissensbereiche, die einen Einfluss auf HCI haben:

„The ideal designer of an interactive system would have expertise in a range of topics: psychology and cognitive science to give her knowledge of the user’s perceptual, cognitive and problem-solving skills; ergonomics for the user’s physical capabilities; sociology to help her understand the wider context of the interaction; computer science and engineering to be able to build the necessary technology; business to be able to market it; graphic design to produce an effective interface presentation; technical writing to produce the manuals, and so it goes on.“ (Dix, et al., 2004a S. 4)

Definition

Da keine Übereinstimmung über die einflussnehmenden Wissensbereiche herrscht, existiert auch keine allgemeinverbindliche Definition von HCI (Hewett, et al., 1996). In der Literatur wird aber oft folgende Definition verwendet:

„Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them.“ (Hewett, et al., 1996)

Anliegen

Die drei Hauptanliegen von HCI sind Menschen, Computer und die Tätigkeiten, die mit Hilfe eines Systems durchgeführt werden. Damit die Tätigkeit des Benutzers durch das System unterstützt werden kann, muss die Benutzbarkeit des Systems gewährleistet sein. Dies eröffnet ein viertes Anliegen: die *Usability* (Dix, et al., 2004a S. 5).

2.2 Usability

Usability, oder zu Deutsch „Benutzbarkeit“, ist eines der Schlüsselkonzepte von HCI. Damit ein Produkt erfolgreich ist, muss die Usability gewährleistet sein. Dix et al. (2004a S. 5-6) definieren drei Kriterien die erfüllt sein müssen, damit ein Produkt erfolgreich ist:

- **Das Produkt muss *nützlich* sein**
Das Produkt muss die Anforderungen erfüllen, z.B. Musik abspielen, Essen kochen, ein Dokument formatieren.
- **Das Produkt muss *benutzbar* sein**
Tätigkeiten müssen sich mit dem Produkt einfach und natürlich erledigen lassen, ohne die Gefahr von Fehlern, etc.
- **Das Produkt muss *benutzt* werden**
Die Menschen müssen es einsetzen *wollen*, es muss attraktiv sein, Spass erzeugen, etc.

Allerdings ist die Usability von Produkten nicht allgemeingültig. Das heisst, ein Produkt, das für eine bestimmte Person in einem bestimmten Umfeld benutzbar ist, kann für eine andere in demselben Umfeld nicht zu gebrauchen sein.

Dementsprechend wird Usability nach ISO-Standard 9241-11 (ISO, 1998) wie folgt definiert:

„The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use“

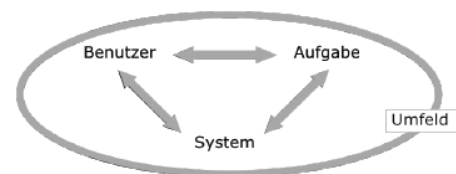


Abb. 3: Usability im Kontext (Hübscher, 2008 S. 7)

Die Definition zeigt, dass Usability nicht ausschliesslich eine Eigenschaft des Produktes selbst ist (Richter, et al., 2007 S. 4).

Sie ist immer im Kontext von Benutzer, Aufgabe, System und Umfeld zu betrachten.

Um die Usability von Systemen zu beurteilen, wird sie oft in verschiedene Kriterien oder Komponenten unterteilt. Diese Unterteilung wird, je nach Quelle, unterschiedlich vorgenommen:

- Nielsen (2003) beschreibt fünf Usabilitykriterien: Learnability, Efficiency, Memorability, Errors und Satisfaction.
- Quesenbery (2004) beschreibt folgende Kriterien: Effective, Efficient, Engaging, Error Tolerant, Easy to Learn.
- Dix et al. (2004a) definiert folgende Hauptprinzipien: Learnability, Flexibility und Robustness. Diese Prinzipien werden weiter in verschiedene Unterprinzipien unterteilt.

Allerdings ist nicht jedes Kriterium gleich wichtig ist. Je nach Art oder Benutzer sind die Kriterien verschieden zu gewichten und müssen in einer Balance zueinander stehen (Quesenbery, 2004).

Welche Kriterien und Prinzipien zu befolgen sind, um die Usability eines System zu gewährleisten, ist ein zentrales Problem und hängt eng mit dem Kontext, der zu entwickelnden Applikation zusammen. Damit Entwickler und Designer eines User-Interfaces die richtigen Entscheidungen, welche die Usability eines Systems betreffen, fällen können, muss ihnen dieses Wissen in einer bestimmten Form zur Verfügung stehen. Dieses Wissen findet sich üblicherweise in Form von **Usability-Guidelines** (Mariage, et al., 2004).

2.3 Usability-Guidelines

Vanderdonckt (1999) definiert Usability-Guidelines, oder kurz Guidelines, als „[...] a design and/or evaluation principle to be observed in order to get and/or guarantee the usability of a UI for a given interactive task to be carried out by a given user population in a given context.“

Aus dieser Definition wird ebenfalls die Verwandtschaft zur Definition der Usability gemäss ISO-Standard 9241-11 (ISO, 1998) deutlich. Da Usability im Kontext zu betrachten ist, sind entsprechend auch Usability-Guidelines kontextabhängig.

Problematik der Terminologie

Usability-Guidelines existieren in verschiedenen Formen und lassen sich in zahlreichen Quellen finden (Mariage, et al., 2004). Allerdings gibt es in der Literatur leider keine Übereinstimmung bezüglich der Terminologie, die im Umfeld von Usability-Guidelines anzutreffen ist. Einerseits benutzen Autoren unterschiedliche Begriffe um den gleichen Sachverhalt zu beschreiben, andererseits werden zwar dieselben Begriffe benutzt, es wird aber etwas anderes darunter verstanden.

Um einige Beispiele zu nennen:

Mariage et al. (2004) versteht unter dem Begriff *Design rules* konkrete und spezifische Vorgaben, die das Design eines User-Interfaces betreffen. Für Dix et al. (2004a) wird jedoch unter demselben Begriff die Gesamtheit aller möglichen Arten von Usability-Guidelines verstanden, also auch unspezifische Guidelines oder Prinzipien.

Für konkrete und spezifische Vorgaben, die bei Mariage et al. (2004) unter dem Begriff *Design rules* zusammengefasst werden, benutzt Mayhew (1999) den Begriff *Standards*. *Standards* wiederum werden von Dix et al. (2004a), von Richter et al. (2007) und Mariage et al. (2004) als Sammlung von Gestaltungsvorgaben zum Zwecke der Standardisierung von User-Interfaces bezeichnet.

2.3.1 Klassifizierungen von Usability-Guidelines

Aufgrund der unterschiedlichen Definitionen der Begriffe ist eine klare Einteilung von Usability-Guidelines nur bedingt möglich. Eine mögliche Klassifizierung von Usability-Guidelines kann anhand ihrer Art und ihrer Quelle vorgenommen werden (Mariage, et al., 2004):

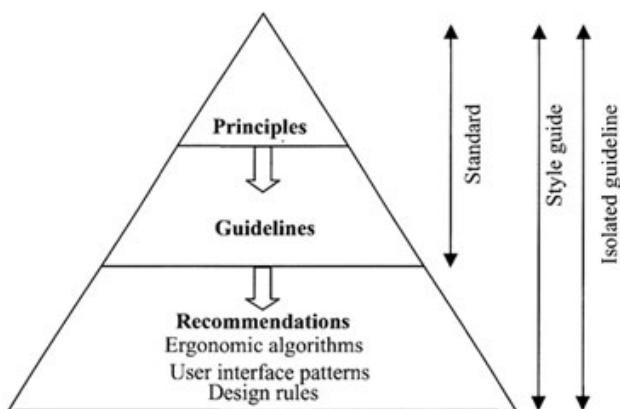


Abb. 4: Klassifizierung von Usability-Guidelines (Mariage, et al., 2004)

Die Art (im Diagramm in fetter Schrift dargestellt) entspricht dem Abstraktionsgrad, den Usability-Guidelines aufweisen. Dieser reicht von generell gehaltenen Prinzipien, die einen hohen Interpretationsaufwand erfordern, bis hin zu spezifischen Vorgaben, die keine Interpretation mehr zulassen.

Als **Quelle** (auch **ergonomische Quellen** genannt) wird der Ursprung bezeichnet, aus denen Usability-Guidelines entstammen. Quellen sind beispielsweise Styleguides, Standards oder Patterns.

2.3.1.1 Klassifizierung nach Art

Principles sind Prinzipien oder Grundsätze (auch **Heuristik** oder **goldene Regeln** genannt), die konzeptionelle Designentscheidungen während des Entwicklungsprozesses lenken. Sie widerspiegeln das Wissen über die Wahrnehmung, das Lernvermögen und das Verhalten von Menschen in Bezug auf deren Interaktion mit Computern. Sie sind in *generischer* Art und Weise formuliert, damit sie für viele Fälle angewendet werden können (z. B. „*use consistent images and metaphors with the external world in appearance and the behavior.*“). Einige Beispiele solcher Prinzipien sind die „*Ten Usability Heuristics*“ (Nielsen, 2005) oder die „*Eight Golden Rules of Interface Design*“ (Shneiderman, 2004).

Guidelines basieren auf den *Principles*, sind aber spezifisch für einen Designbereich. Ein Beispiel für eine Guideline im Webbereich wäre: „*Use a consistent look and a visual language inside the site.*“. Einige Guidelines bedürfen allerdings einer Interpretation und müssen angepasst werden, um den spezifischen Bedürfnissen einer Organisation oder eines bestimmten Designs gerecht zu werden.

Recommendations (auch **Conventions** genannt) legen eine konzeptionelle Entscheidung für einen bestimmten Anwendungsbereich fest. Sie sind eindeutig und lassen keinen Interpretationsspielraum zu (z. B. „*Provide search, help, and home page links at the bottom of each page.*“). *Recommendations* werden aus existierenden *Principles* und *Guidelines* unter Berücksichtigung der Anforderungen und des Nutzungskontextes abgeleitet. *Recommendations* umfassen *Design rules*, *Ergonomic algorithms* und *User interface patterns*.

2.3.1.2 Klassifizierung nach Quelle

Design rules bestehen aus einem Set von funktionalen Anforderungen und spezifizieren das Design eines bestimmten Interfaces. Sie sind sehr konkret formuliert und erfordern keine Interpretation durch den Entwickler oder Designer. Dies bestärkt daher deren durchgängige Anwendung.

Ergonomic algorithms integrieren üblicherweise einfache *Design rules* in einer systematischen Prozedur, die schneller anwendbar ist als einzelne Regeln. Die Verwendung solcher Algorithmen gibt Designern eine grössere Flexibilität bei der Anwendung solcher *Design rules*. Solche Algorithmen existieren normalerweise eher in Form von Software als auf Papier. Beispielsweise berechnet die Software DESIGN ADVISOR (Faraday, 2000), anhand eines Algorithmus, den visuellen Weg des Auges zwischen Elementen auf einer Webseite abhängig von deren Art, Größe, Farbe und Position.

Isolated guidelines (auch **Guideline set** oder **Compilation of guidelines** genannt) enthalten Empfehlungen für ein User-Interface. Diese können entweder spezifische Aspekte der Usability beschreiben (z.B. der Einsatz von Farbe bei einer Suche) oder einen grösseren Kontext im Bereich von HCI abdecken (z.B. die Erwartungskonformität von Elementen einer Webseite). *Isolated guidelines* enthalten teilweise Beispiele, Kommentare und Begründungen. *Isolated guidelines* entstammen normalerweise einem Konsens zwischen Leuten, die mit Guidelines arbeiten, bzw. davon betroffen sind (z.B. HCI-Experten, Benutzer, Entwickler, Designer). Wenn solche Guidelines empirisch getestet und validiert sind, werden sie oft in Tagungsbänden oder Fachzeitschriften veröffentlicht. Ein Beispiel dafür sind die „*Guidelines for Multimedia on the Web*“ (Nielsen, 1995).

User interface patterns verfolgen einen anderen Ansatz, um mehrere isolierte Guidelines in ein verständliches und anwendbares Regelwerk zusammenzufassen. Sie bieten eine umfassende Lösung für eine Reihe ähnlicher Designprobleme. Im Gegensatz zu Guidelines haben Patterns den Vorteil, stets in einem Nutzungskontext zu stehen, innerhalb dem sie angewendet werden sollen.

Standards sind eine Sammlung von Richtlinien, die für eine Standardisierung von User-Interfaces sorgen sollen. Sie enthalten normalerweise keine konkreten Vorgaben, wie ein bestimmtes Design umgesetzt werden soll, sondern nur *principles* und *guidelines*. Herausgeber sind sowohl nationale als auch internationale Organisationen wie Regierungen (z.B. über gesetzliche Verordnungen), Militär und zivile oder industrielle Einrichtungen. Beispiele dafür sind die ISO (International Standardization Organization) und das W3C (World Wide Web Consortium). Ein Beispiel für einen Standard sind die „*Web Content Accessibility Guidelines (WCAG) 2.0*“ (W3C, 2008).

Styleguides umfassen eine Sammlung von Usability-Guidelines verschiedener Abstraktionsgrade (*principles*, *guidelines* und *recommendations*) und/oder funktionale und nichtfunktionale Anforderungen, welche die Konsistenz einer Sammlung unterschiedlicher User-Interfaces sicherstellen soll. Diese Sammlung kann spezifisch für ein Betriebssystem (z.B. Windows, Mac-OS), für einen Softwarehersteller (z.B. IBM, Nokia), eine Domäne (z.B. Webseiten, Web-Applikationen, Telekommunikation), ein Unternehmen, eine Produktfamilie (z.B. Office-Applikationen) oder ein einzelnes Produkt, bzw. Projekt sein.

2.3.1.3 Andere Arten der Klassifizierung

Die vorhergehende Klassifizierung von Usability-Guidelines sind nur einige der Möglichkeiten. Eine weitere Art der Klassifizierung von Usability-Guidelines bieten Beier et al. (2003):

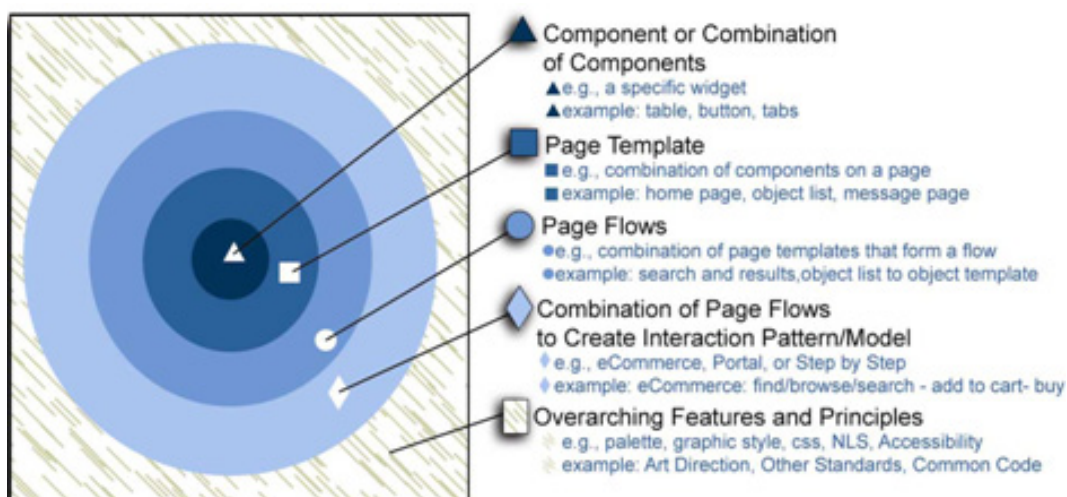


Abb. 5: The Bull's-Eye: A Framework for Web Application UI Design Guidelines (Beier, et al., 2003)

Die Guidelines werden anhand des *Bull's-Eye Framework* klassifiziert. Dabei werden sie, ihrer Abstraktionsgrade entsprechend, in fünf Schichten eingeteilt. Diese reichen von generellen Guidelines, bis hin zu spezifischen Guidelines für bestimmte Elemente.

Eine weitere Art der Klassifizierung bieten Dix et al. (2004b). Unter *Design rules* werden alle Arten von Usability-Guidelines verstanden. Diese werden in drei Typen unterteilt:

Typ der <i>Design rules</i>	Charakteristiken
Principles	Abstrakte Design Rules, geringe Autorität, hohe Allgemeingültigkeit
Standards	Spezifische Design Rules, hohe Autorität, limitierter Anwendungsbereich
Guidelines	Geringe Autorität, grösserer Anwendungsbereich

Tabelle 4: Klassifizierung von Usability Guidelines (Dix, et al., 2004b)

2.3.1.4 Interpretationsaufwand und Umsetzung

Nahezu alle Arten von Guidelines bedürfen einer Interpretation, damit sie im spezifischen Kontext einer Problemstellung (z.B. ein Designproblem) Verwendung finden können. Folgende Abbildung veranschaulicht den Interpretationsbedarf und Abstraktionsgrad von Guidelines in Abhängigkeit ihrer Quelle (Mariage, et al., 2004):

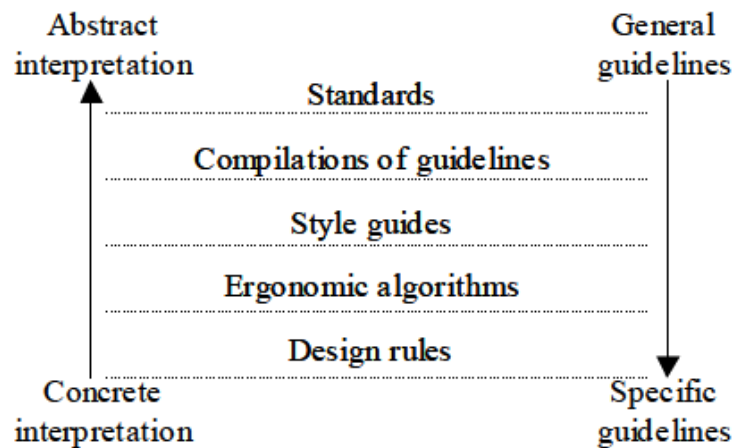


Abb. 6: Interpretations- und Abstraktionsgrad nach Quellen (Mariage, et al., 2004)

Unspezifische Guidelines, wie sie beispielsweise in Standards zu finden sind, bedürfen eines hohen Interpretationsaufwandes, haben jedoch für ein breites Spektrum von User-Interfaces ihre Gültigkeit. Spezifische Guidelines benötigen keine Interpretation, sind dadurch aber auch stark in ihrem Anwendungsbereich eingeschränkt. (Dieses und andere Probleme werden im Kapitel 4 *Die Suche nach Erkenntnissen – Recherchen* ausführlicher behandelt.)

Folgende Abbildung zeigt den gleichen Sachverhalt in einer anderen Form (Mariage, et al., 2004):

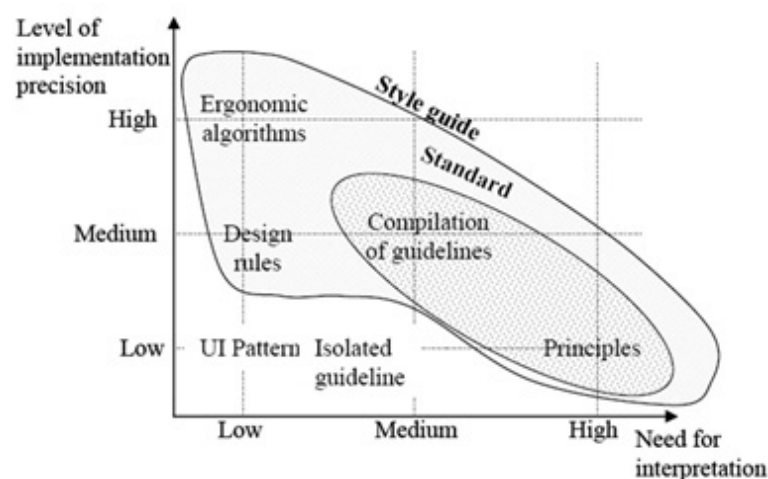


Abb. 7: Interpretations- und Umsetzungsgrad nach Quellen (Mariage, et al., 2004)

In dieser Grafik wird zusätzlich veranschaulicht, wie viele Details bezüglich der Umsetzung, bzw. Implementation von Usability-Guidelines von welchen Quellen geliefert werden.

2.3.2 Schreibe Patterns – Keine Guidelines!

Patterns und Guidelines sind eng verwandt. Viele Patterns replizieren die Information, die bereits in Form von Guidelines existiert. Der Unterschied zwischen Patterns und Guidelines liegt primär in der Perspektive und der Repräsentation der Information (Henninger, 2001).

Genau in der Perspektive und Repräsentation der Information liegt jedoch das Problem von Guidelines. In Kapitel 2.3.1.4 *Interpretationsaufwand und Umsetzung* wurde der Aufwand für die Interpretation als einer der Mängel von Usability-Guidelines identifiziert. Ein weiterer Mangel ist der fehlende Kontext von Guidelines. Die Anwender einer Guideline wissen somit nicht, bei welchem Problem eine bestimmte Guideline angewendet werden soll (Mariage, et al., 2004).

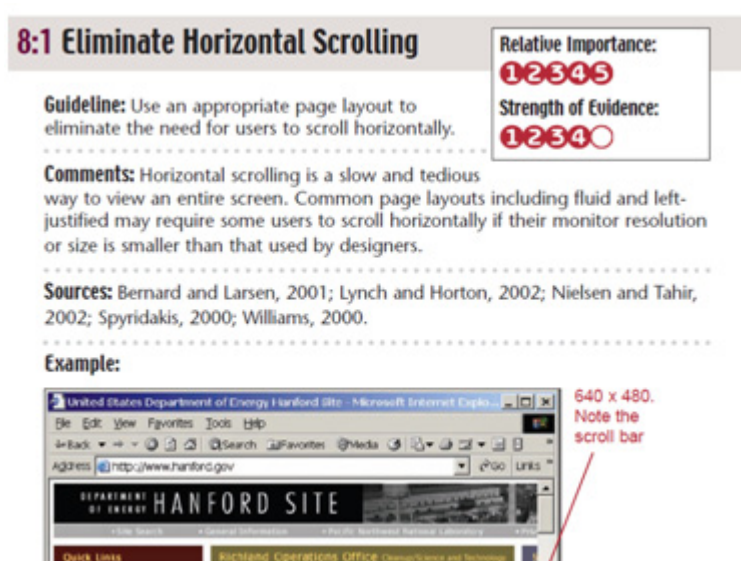
Aufgrund ihres Aufbaus, leiden Patterns nicht unter diesen Mängeln. Die Idee hinter Patterns ist, wiederkehrende Probleme mitsamt ihres Kontextes und Kräften, die auf das Problem wirken, zu erfassen und dafür eine generelle Lösung anzubieten (Henninger, 2001). Ein typisches Pattern beschreibt das Problem, die dafür entsprechende Lösung (mit Beispielen), die Überlegungen hinter der Lösung, verwandte Patterns und andere relevante Details, wie zum Beispiel Resultate von Usability-Tests (Spool, 2003). Der Nutzungskontext eines Patterns ist somit immer klar gegeben.

Aufgrund dieses Vorteils, wird in der Literatur oft die Verwendung von Patterns anstatt Guidelines propagiert. Griffiths et al. (2003), Tidwell (1999) und Spool (2003) beschreiben ihre jeweiligen Überlegungen und Beweggründe dafür.

Eine empirischen Studie (Koukouletsos, et al., 2007), in welcher Guidelines und Patterns gegenübergestellt werden, scheint die Überlegenheit von Pattern gegenüber Guidelines zu bestätigen. In dieser Studie wird festgestellt, dass Patterns, insbesondere für Neulinge im Bereich der Entwicklung oder des Designs, einen Vorteil gegenüber Guidelines haben können.

2.3.3 Beispiel einer guten Guideline

Es gibt keine Regeln bezüglich des Aufbaus von Guidelines, wie sie formuliert werden sollten oder welche Informationen sie bieten müssen. An dieser Stelle wird jedoch auf ein gutes Beispiel eines klaren Aufbaus verwiesen (Usability.gov, 2003):



Die Guideline hat einen Titel, der mit einem Satz erklärt, worum es dabei geht. Darunter folgen einige weiterführende Erklärungen und Kommentare.

Abgeschlossen wird die Guideline mit Angaben bezüglich der Forschungsliteratur, welche die Guideline untermauert, Beispielen, sowie einer Bewertung zur Wichtigkeit und der empirischen Evidenz der Guideline.

Abb. 8: Beispiel des Aufbaus einer Guideline (Usability.gov, 2003)

2.4 Styleguides

Ohnemus (1997) definiert Styleguides als: „[...] *a basis for development and is a collection of principles, guidelines and conventions brought together into a single medium to define a unified look and feel for products and services. Taken together, this information structure provides the foundation for good design.*“

Gemäss der Definition ist das Ziel von Styleguides die Konsistenz von Produkten und Dienstleistungen sicher zu stellen. Konsistenz ist ein wichtiges Kriterium für die Usability eines Produktes, bzw. einer Applikation oder Systems.

2.4.1 Konsistenz

Für Ohnemus (1997) reduziert Konsistenz den Lernaufwand, verbessert die Leistung und senkt die Fehlerrate. Laut Gale (1996) bieten konsistente Applikationen folgende Vorteile:

- Sobald Benutzer eine Applikation kennen und beherrschen, kann dieses Wissen auf andere Applikationen übertragen werden. Dies minimiert den Lernaufwand und regt Benutzer an, andere Applikationen zu verwenden.
- Konsistente Applikationen helfen dem Personal, gegenüber Kunden kompetent und selbstbewusst zu erscheinen. Dies verbessert den Eindruck des Kunden bezüglich der Qualität der Dienstleistung.
- Konsistenz zwischen Applikationen erlaubt es den verschiedenen Projektteams, Quellcode wiederzuverwenden. Dies reduziert den Implementationsaufwand und die Wartungskosten von Applikationen.

Konsistenz ist allerdings keine eindimensionale Eigenschaft eines Systems, welche entweder erfüllt oder nicht erfüllt ist. Konsistenz steht immer in Relation zu etwas „Anderem“ (Dix, et al., 2004a S. 265). Wilson (2001) führt einige Punkte auf, die in Bezug zur Konsistenz stehen:

Bezugspunkte der Konsistenz	
• Konsistenz bezüglich der Benutzererwartungen • Konsistenz bezüglich der Terminologie • Konsistenz bezüglich der Interaktion • Konsistenz bezüglich dem Erscheinungsbild	• Konsistenz innerhalb derselben Applikation • Konsistenz bezüglich verwandter Applikationen • Konsistenz bezüglich der Fehlermeldungen • Konsistenz bezüglich Icons

Tabelle 5: Bezugspunkte der Konsistenz (Wilson, 2001)

Konsistenz ist demzufolge ein komplexes Thema und muss unter verschiedenen Blickpunkten betrachtet werden. Trotzdem ist die Sicherstellung der Konsistenz von Applikationen, aufgrund der Vorteile, ein wichtiges Ziel, zu dessen Erreichung Stylguides ein Mittel sind (Hart, 2000).

2.4.2 Vorteile von Styleguides

Durch die Sicherstellung der Konsistenz bieten Styleguides sich eine ganze Reihe von Vorteilen, die aus verschiedenen Perspektiven betrachtet werden können (Gale, 1996):

Vorteile aus Sicht des Endbenutzers	Vorteile aus Sicht der Entwicklung	Vorteile aus Sicht des Unternehmens
• Weniger Fehler und Frustration • Gesteigertes Vertrauen in das System • Geringerer Lernaufwand • Bessere Stimmung und Motivation • Höhere Effizienz • Höhere Produktivität • Geringere Ablehnung gegenüber neuen Technologien	• Weniger „Neuerfindungen“ • Weniger willkürliche Designentscheidungen • Erlaubt die Erstellung wiederverwendbarer Software • Geringerer Entwicklungsaufwand • Beibehaltung der Kontrolle bezüglich Look & Feel	• Höhere Kundenzufriedenheit und Dienstleistungsqualität • Steigerung des Bekanntheitsgrades der Marke • Steigerung des Bekanntheitsgrades des Produktes • Geringere Trainingskosten • Entlastung der Hotline (Unterstützung bei Problemen) • Steigerung der Mitarbeiterbindung • Steigerung der Akzeptanz von Benutzern für neue Systeme

Tabelle 6: Vorteile von Styleguides (Gale, 1996)

2.4.3 Klassifizierungen von Styleguides

In Kapitel 2.3.1.2 *Klassifizierung nach Quelle* wurde gezeigt, dass Styleguides spezifisch für einen bestimmten Kontext sind. Dieser Kontext kann die Plattform (z.B. Betriebssystem), der Hersteller einer Software (IBM, Microsoft, Apple, etc.), die Domäne (z.B. Webbereich, Telekommunikation), das Unternehmen und entsprechend, die in einem Unternehmen erstellte Produktfamilie, ein einzelnes Produkt oder ein Projekt sein.

Allerdings gestaltet sich die Klassifizierung von Styleguides ähnlich problematisch wie die von Usability-Guidelines (2.3 *Usability-Guidelines*). In der Literatur gibt es keine Übereinstimmung anhand welcher Styleguides *eindeutig* eingeteilt werden könnten.

Folgende Tabelle zeigt einige der verschiedenen Klassifizierungen von Styleguides und die dafür vergebenen Bezeichnungen:

Kontext	Mayhew (1999 S. 312)	Quesenbery (2001a)	Richter et al. (2007 S. 50)	
Plattform	Platform Style Guide	Plattform Guide	Hersteller- oder plattformabhängiger Styleguide	
Hersteller	-			
Domäne	-	Design Guide	-	
Unternehmen	Corporate Style Guide	Corporate Application Guide, Corporate Application Style Guide	Unternehmens- Styleguide	
Produktfamilie	Product Family Style Guide			
Produkt	Product Style Guide		Projekt-Styleguide	
Projekt				

Tabelle 7: Klassifizierungen von Styleguides

2.4.4 Inhalt von Styleguides

In der Literatur gibt es diverse Beispiele, welche Informationen in Styleguides enthalten sein sollten. Laut Richter et al. (2007 S. 53) sollte ein Styleguide Informationen zu folgenden Themen bieten:

Thema	Beschreibung
Technologische Rahmenbedingungen und Zielgruppe	Beschreibt auf welche Systeme sich der Styleguide bezieht und wer die Empfänger des Styleguides sind.
Software-Ergonomie	Allgemeingültige Regeln, welche im konkreten Fall zu berücksichtigen sind (z.B. Anzahl Menüeinträge) sowie Regeln bezüglich der spezifischen Zielgruppe und Anwendung (z.B. vollständig über die Tastatur bedienbare Dialoge).
Anwendungsregeln	Beschreibt, welche Elemente in welchen Situationen verwendet werden.
Verhalten der GUI-Elemente	Beschreibung der Reaktion des Systems (z.B. Selektion eines Eintrages, Deaktivierung von Controls).
Navigation	Beschreibung der Navigationselemente (Menüs, Links, etc.).
Visuelles Design	Farbschema, Farbkontraste, Schriften, Layout, Icons, etc. Auch Corporate-Design-Aspekte werden hier referenziert.
Technische Umsetzbarkeit	Hinweise auf die technische Umsetzung, z.B. Referenz auf verfügbare GUI-Komponenten.
Terminologie	Begriffe und Bezeichnungen der Benutzeroberfläche, wie wird der Benutzer angesprochen, welche Fachbegriffe werden verwendet. Formulierung von Fehlermeldungen.
Tastaturbedienung	Shortcuts, Kommandos, Tabulator-Reihenfolge, Default-Buttons in einem Dialog.

Tabelle 8: Inhalt projektspezifischer Styleguides (Richter, et al., 2007 S. 53)

Quesenbery (2001a) geht einen Schritt weiter und definiert zusätzlich noch eine Reihe weiterer Informationen, die ein Styleguide enthalten sollte:

- Erklärungen zur Benutzung des Styleguides selbst und Erklärungen zu den Prozessen, die den Styleguide als Dokument betreffen (z.B. Änderungen, Reviews, etc.).
- Angaben zur Methodologie, die für die Erstellung des Styleguides eingesetzt wird (z.B. ein UCD-Prozess).
- Angaben zur eigentlichen Zielgruppe der Applikation selbst (z.B. Usability-Goals, Benutzerprofile und -charakteristiken).
- Informationen zu übergeordneten Designprinzipien und Metaphern, welche das User-Interface betreffen.
- Referenzen zu anderen Informationsquellen (z.B. zum Entwicklungsprozess der Applikation, zu Sitzungsprotokollen, zu anderen verwendeten Styleguides oder Guidelines).

Beispiele für eine mögliche Gliederung von Styleguides bieten jeweils „*A Collaborative Approach to Developing Style Guides*“ (Gale, 1996) und „*Building a Better Style Guide*“ (Quesenbery, 2001a).

2.4.5 Styleguides im Software-Entwicklungsprozess

2.4.5.1 *Rappels sur les méthodologies de développement*

Selon Fowler (2005) les premières méthodologies de développement ont été mises en place à la suite d'une période où la conception de Software était pour beaucoup synonyme de chaos. La problématique principale était l'ampleur prise par certains systèmes. Plus ils étaient conséquents, plus la gestion et le débogage étaient délicats.

Ces méthodologies, inspirées de l'ingénierie, avaient pour objectif de rendre le développement plus prévisible et efficace. Les processus disciplinés étaient perçus comme bureaucratiques et n'ont ainsi jamais bénéficiés de bonne réputation. Highsmith (1999) leur a d'ailleurs attribué le nom de méthodologies „monumentales“. Elles n'ont de plus pas montré des résultats concluants.

Les méthodologies dites „légères“ sont apparues en réaction aux méthodologies lourdes. Dénommées maintenant Agile, selon Fowler (2005): Ces nouvelles méthodes sont un habile compromis entre trop de processus et aucun processus formel, fournissant juste le compte nécessaire pour obtenir un succès raisonnable. La différence immédiate est qu'elles sont moins orientées documentation.

Adaptatifs contre prédictifs

Les méthodes lourdes planifient de grandes parties de processus ou d'application sur le long terme et supportent mal les changements (d'où le terme prédictif, de prévoir). „*Agile methods are adaptive rather than predictive*“ (Fowler, 2005). Ces méthodes adaptatives s'ajustent aux changements de besoins et de processus, contrairement aux premières. L'adjectif adaptatif renvoie à leur capacité à se modifier elles-mêmes selon la situation.

Les itérations, le développement itératif et incrémentiel

Le développement itératif est connu depuis longtemps. Son but est de mieux contrôler un processus imprévisible (Fowler, 2005). Il consiste en la production régulière d'une version fonctionnelle du système, mais intégrant une partie des fonctionnalités seulement. Ces versions permettent ainsi un retour d'information sur la situation actuelle. Les méthodes adaptatives sont obligatoirement itératives, ce qui n'est pas nécessairement le cas des méthodes prédictives.

Le développement itératif est une stratégie de re travail planifié où chaque période est utilisée pour réviser ou améliorer des parties du système (Cockburn, 2008). En opposition au développement incrémentiel qui emploie une stratégie de paquetage et de planification où les différentes parties du système sont développées à différent moment et sont intégrées une fois terminées. Cockburn (2008) préconise une utilisation simultanée des deux méthodes, les projets restent libres d'utiliser l'une, l'autre, les deux ou aucune des deux méthodes.

Différences des méthodologies

La graphique ci-dessous montre le déroulement de ces méthodologies dans le temps:

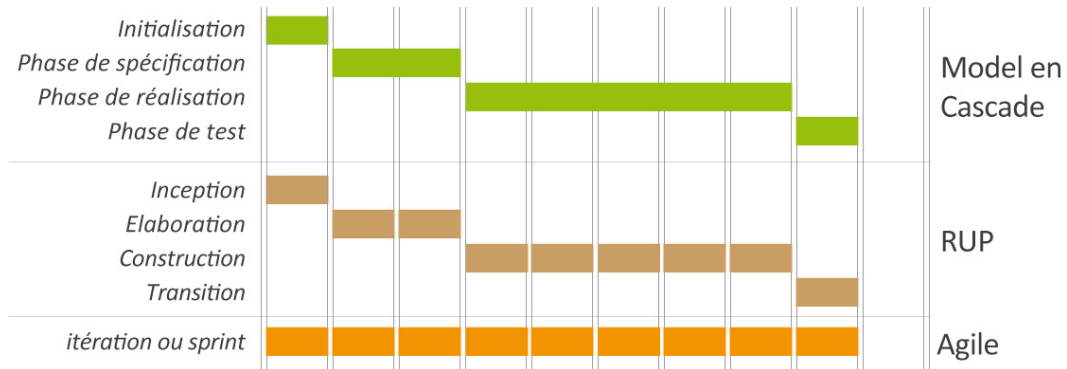


Abb. 9: Différences entre méthodologies, inspiré du cours „Vorgehensmodelle“ (Steimle, 2008)

La première partie du diagramme représente une itération unique d'un processus en cascade. Les spécifications sont décrites en une fois, avant la construction, puis les tests. Il s'agit de l'exemple d'une méthodologie lourde et prédictive. Tout ce qui va être réalisé est décrit a priori (prédictif).

La seconde partie du graphique montre un développement itératif RUP. Fowler (2005) décrit RUP comme étant basé sur les cas d'utilisation, itératif et centré sur l'architecture. *„My experience with RUP is that its problem is its infinite variability. I've seen descriptions of RUP usage that range from rigid waterfall with 'analysis iterations' to picture perfect agile.“* (Fowler, 2005) Ici la configuration se situe entre Agile et modèle en cascade, entre itératif et prédictif.

Enfin, la dernière partie montre un développement Agile, itératif et adaptatif. Dans chaque itération de nouvelles fonctionnalités sont ajoutées ou améliorées, implémentées et testées.

2.4.5.2 Les Styleguides dans Agile

La méthodologie Agile privilégie un système qui fonctionne plutôt qu'une documentation complète (Beck, et al., 2001). Ainsi, créer un document de grande complexité avant le développement est par définition non-Agile (Edmonds, 2008). Edmonds (2008) ajoute: *„Codifying conventions as they develop insures a top notch user experience, facilitates efficient implementation, and provides the key design criteria when it comes time to do UI refactoring.“*. Donc codifier certaines conventions améliore la qualité, aide et réduit les risques d'inconsistances lors de développement incrémentiel.

Le statut du styleguide peut varier: *„As an application develops, the leader vs follower status of the style guide can vary“* (Edmonds, 2008). Il peut être parfois meneur ou suiveur selon la situation, c'est-à-dire documenter des décisions de design ou des parties d'application à posteriori ou spécifier à priori l'interface pour le faciliter, pour assurer sa consistance par exemple.

Kreitzberg et al. (2009) ajoutent encore qu'il n'est pas nécessaire que le design initial soit „gravé dans la pierre“. Ils ajoutent cependant que l'adaptation de l'interface utilisateur n'est pas aussi aisée qu'un changement de code.

2.4.5.3 Les styleguide dans les autres méthodologies

Le Styleguide est un artefact central dans *„The Usability Engineering Lifecycle“* de Mayhew (1999). En plus des décisions de design, ce document contient les requirements et d'autres informations nécessaires pour que l'application garde toute sa consistance.

2.4.6 Projektspezifische Styleguides

Der Scope dieser Masterarbeit bezieht sich auf die kollaborative Erstellung und Anwendung von projektspezifischen Styleguides. Mit projektspezifischen Styleguides sind Styleguides gemeint, die innerhalb eines Unternehmens für ein Produkt oder eine Familie von Produkten im Zuge eines Projektes erstellt werden. Dieses Kapitel widmet sich daher ausschliesslich der Theorie bezüglich dieser Art von Styleguides.

2.4.6.1 Usability-Ziele projektspezifischer Styleguides

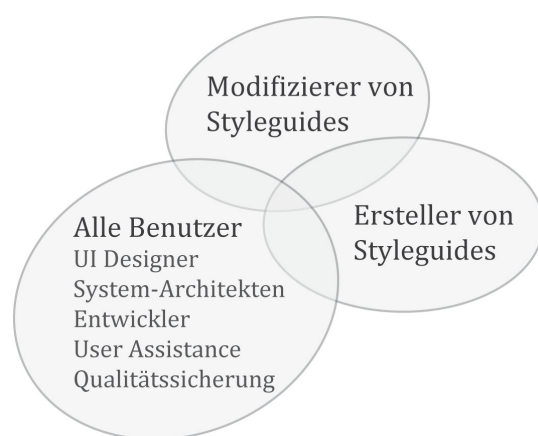
Quesenbery (2001a) definiert Usability-Ziele von projektspezifischen Styleguides und bezieht sich dabei auf die Definition von Usability gemäss dem ISO-Standard 9241-11 (ISO, 1998):

Usability Charakteristik	Ziel
Effizienz	Verbesserte Qualität: Der Zeitaufwand für das Design einer Benutzeroberfläche wird reduziert, weil grundlegende Guidelines klar dokumentiert sind, Werkzeuge wieder- verwendet werden und „Best Practice“-Richtlinien für Entscheidungen vorhanden sind.
Effektivität	Verbesserter Prozess: Weil gemeinsame Guidelines vorhanden sind, arbeiten verschiedene User-Interfaces besser zusammen. Erste Designentwürfe sind effektiver. Es finden weniger Nacharbeiten statt, da weniger Usabilityprobleme gelöst werden müssen.
Zufriedenstellung	Verbesserte Usability: Für Benutzer, wie auch für Designer einer Applikation, wird die „User-Experience“ verbessert. Designer haben die Befriedigung der Erstellung exzellenter User-Interfaces, die Benutzer profitieren von einer besseren Usability.

Tabelle 9: Usability-Ziele projektspezifischer Styleguides (Quesenbery, 2001a)

2.4.6.2 Ersteller, Benutzer und Stakeholder

Ersteller, Benutzer und Stakeholder von Styleguides werden in der Literatur verschieden definiert. Gale (1996) definiert drei Gruppen von Personen, die mit Styleguides konfrontiert sind. Als *Endbenutzer* definiert Gale den Anwender einer Applikation, die mit Hilfe von Styleguides erstellt worden ist. Als *IT-Entwickler* werden Personen bezeichnet, die Applikationen mit Hilfe von Styleguides entwickeln. Sie sind also die „direkten“ Benutzer der Styleguides. Mit *Organisation* oder *Business* werden die Stakeholder von Styleguides bezeichnet. Gale definiert jedoch keine Gruppe, die für die Erstellung der Styleguides selbst verantwortlich ist.



Quesenbery (2001a) definiert andere Gruppen. Die Abbildung zeigt, dass es drei Gruppen von Benutzern gibt: Die Ersteller von Styleguides, diejenigen, die Styleguides modifizieren und die eigentlichen Benutzer von Styleguides. Diese Gruppen können, müssen sich aber nicht überlappen.

Quesenbery definiert keine Gruppe von Stakeholdern, das heisst, denjenigen, welche die Erstellung eines Styleguides in Auftrag geben und für den Prozess verantwortlich sind.

Abb. 10: Benutzergruppen von Styleguides (Quesenbery, 2001a)

3 Der Weg zum Konzept – Vorgehen und Methodologie

Ce chapitre explique les techniques utilisées dans le cadre de ce projet. Après une brève vue d'ensemble des principales activités, les différentes tâches sont passées en revues. Cette partie complète le déroulement du projet et le choix de la méthodologie expliqués précédemment dans le chapitre 1.5 *Projektplanung*.

Dans la deuxième partie de ce chapitre nous aborderons les aspects éthiques du projet.

3.1 Chronologisches Vorgehen

Dans chaque tâche sont listés les méthodes utilisées, les documents entrants et sortants, ainsi qu'une brève description du déroulement et la justification du choix des méthodes. Contrairement à l'aspect séquentiel de la présentation, certaines tâches sont itératives et s'étalent sur toute durée du projet. Les tâches peuvent être regroupées en six activités principales :

Activités	Description
Recherche	Recherche dans la littérature (3.1.1)
Interview	Exécuter les interviews, les débriefings (3.1.2)
Modélisation	Analyse des Personas (3.1.3), Concept Model (3.1.4), Business Use-Cases (3.1.6), Features et Requirements (3.1.7), System Use-Cases (3.1.8), analyse de la concurrence (3.1.9), diagramme de classe et d'état (3.1.10)
Prototypage	Prototypage (3.1.5)
Evaluation	Usability Walkthrough (3.1.11)
Documentation	Ecriture du rapport

Tabelle 10: Tableau des activités réalisées

Le graphique ci-dessous illustre l'aspect itératif des tâches à l'intérieur des activités:

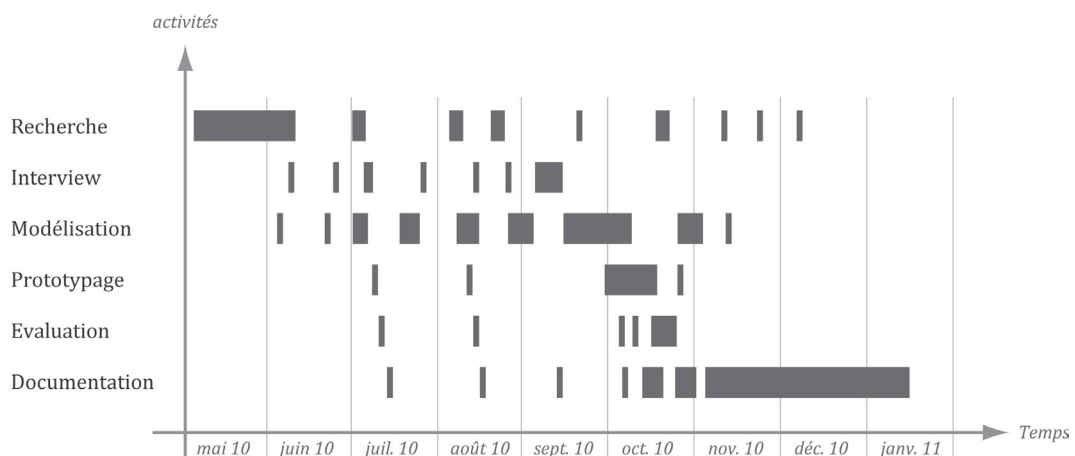


Abb. 11: Déroulement des activités dans le temps, inspiré du cours de Design (Bircher, 2009 S. 32)

3.1.1 Eintauchen in die Materie – Literaturrecherchen

Technique	Recherche et lecture
Input	Littérature du domaine, articles, en partie dans des livres et sur le Web
Output	Idées, problématiques, hypothèse de base des Personas
Description	Cette partie regroupe toute la recherche effectuée dans ce travail. Le domaine de recherche comprend: les différents types de styleguide, les problèmes liés aux styleguides, le processus de développement software avec des styleguides, la taxonomie, la recherche dans les documents, les outils de collaboration. Nous avons lu et analysé plus de septante textes, provenant de livres, d'articles sur internet et de blogs. Ils ont été trouvés à l'aide de moteur de recherche, dans les références d'articles et par notre coach. Cette étape a permis de formuler les hypothèses, d'orienter notre travail et de définir le scope.

3.1.2 Recherche über Benutzer und Aufgaben – Interviews

Technique	Interview ethnographique condensée, interview semi-structurée et débriefing
Input	Hypothèses, problématique, catalogue de questions
Output	Débriefing (caractères et Findings)
Description	Nous avons choisi les interviews semi-structurées et les interviews ethnographiques condensées comme techniques d'acquisition des données. Ces derniers nous ont permis d'observer la manière dont était utilisé les styleguides. Sans être aussi coûteux en temps que l'étude de terrain. Cette technique est un bon compromis entre efficacité et qualité des données récoltées. Des interviews semi-structurées ont été réalisées si le styleguide ne pouvait pas être montré, utilisé ou dans tout autre cas. Sur seize interviews réalisées, dix interviews sont semi-structurées et six ethnographiques condensées. La table ci-dessous montre la répartition des interviews par Persona, par taille d'entreprise et par secteur d'activité de l'entreprise. Nous distinguons deux secteurs, les prestataires de service informatique (développement informatique, agence de design.) et les autres secteurs (transport, banque, assurance).

Personas	Interviews dans les PME	Interviews dans les GE	Branche IT	Branche autre	Total par Persona
Owner	0	3	0	3	3
Writer	2	1	2	1	3
User	4	6	4	6	10
Total	6	10	6	10	16

Tabelle 11: Répartition des interviews par Personas, taille et secteur d'activité des entreprises

Seul deux interviews ont été réalisées en binôme, le reste par une personne seule. Toutes les interviews ont duré une heure environ. La majorité des interviews se sont déroulés sur le lieu de travail des personnes interrogées.

Chaque interview a été débriefé en groupe selon la méthode „*interpretation session*“ de Holtzblatt et al. (2005) que nous avons adapté. Trois types de données ont été protocolés, les informations sur le caractère, sur les tâches et les découvertes (Findings). Cette méthode a permis d'éviter de biaiser les résultats et les interprétations personnelles. Elle a surtout facilité le partage de l'information dans le groupe. Les débriefings ont été faits en partie avec SKYPE.

3.1.3 Verständnisaufbau der Benutzer und Aufgaben – Personas und Szenarios

Technique	Persona, scénario de contexte
Input	Débriefing, hypothèse des Personas
Output	Personas, scénarios contextuels
Description	La technique centrée sur l'utilisateur des Personas (Cooper, et al., 2007), nous a permis de comprendre l'utilisateur, son contexte et ses motivations. Elle a complété le Framework RUP avec le facteur humain de l'application. Nous l'avons choisi pour les bonnes expériences que nous avons déjà faites auparavant. Nous avons commencé par élaborer une hypothèse de base ([HYP-1] à trois Personas), sur la base des rôles connus et de la littérature. Les variables comportementales ont ensuite été définies (dix à quinze par Personas). Une fois les interviews exécutées et débriefées, chaque interview a été mappée sur les variables. L'analyse des Personas a été réalisée en deux itérations, d'abord pour les Personas Owner et Writer et ensuite la Persona User. Les scénarios de contexte ont été créés et intégrés à la description des Personas. Ces scénarios décrivent comment interagissent les Personas avec la future application (Soll-Szenarien).

3.1.4 Bändigung der Informationsflut und Ideen – Concept Model

Technique	Concept Modeling
Input	Findings, Personas, recherche
Output	Concept Model
Description	Pour garder une vue d'ensemble sur l'importante masse d'informations, nous avons utilisé le „<i>Concept Model</i>“ de Brown (2010b). Ce modèle permet de relier les différents concepts, idées, rôles, etc. dans un seul diagramme. Il sert de catalyseur pour les prochaines activités. Il a été également développé en plusieurs itérations. Cette technique est expliquée plus en détail dans <i>Anhang M: Hintergrund zu Concept Models</i>.

3.1.5 Visualisierung von Ideen – Varianten und Prototypen

Technique	Prototype papier, prototype Wireframe
Input	Toutes les informations à disposition
Output	Prototypes de l'application
Description	Nous avons utilisé deux types de prototypages pour le projet. Le prototypage „<i>low-fidelity</i>“ à l'aide de papier et de crayon et le prototypage Wireframe avec un outil de Mockups. La création du prototype s'est déroulée sur quatre itérations, réparti sur plus de la moitié du temps du projet. Le graphique ci-dessous illustre les différentes étapes de design et décision en fonction des itérations:

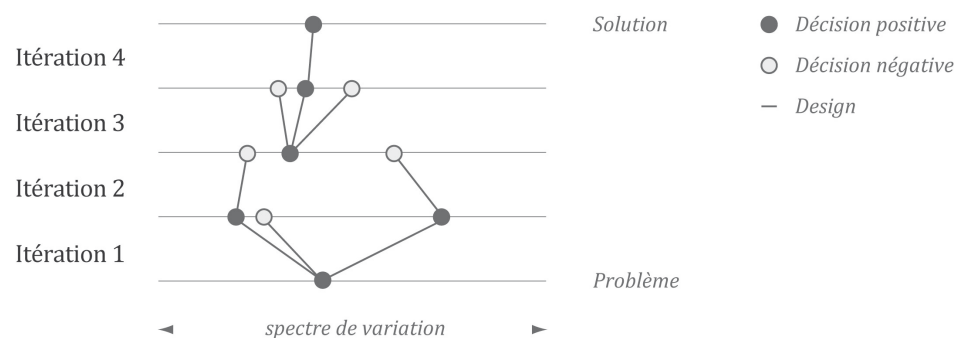


Abb. 12: Étapes du prototypage et du design, inspiré du cours de Design (Bircher, 2009)

1^{er} itération: trois variantes de prototype ont été développées (une par personne). Le but de cette phase était de définir rapidement la structure et le concept de l'application.

2^{ème} itération: les deux variantes les plus différentes ont été encore affinées. Grâce aux nouvelles idées, recherches et informations des interviews, les deux variantes de prototypes papiers ont été abandonnées et un nouveau concept a été défini.

3^{ème} itération: trois nouvelles variantes ont été élaborées (une par personne). Suite à une nouvelle évaluation de team, une seule variante a été choisie. Une fois la structure décidée, le prototype a été réalisé avec l'outil de prototypage Wireframe BALSAMIQ.

4^{ème} itération: le prototype a été amélioré et certains détails affinés.

3.1.6 Analyse des Geschäftskontextes – Business Use-Cases

Technique	Modélisation UML, Business Modeling
Input	Findings, scénarios, littérature
Output	Business Use-Cases (BUC)
Description	Les Business Use-Cases sont un type d'objet UML, qui a pour but de modéliser les processus métiers d'une entreprise. Nous avons choisi cette méthode pour clarifier le contexte d'utilisation des styleguides au sein des entreprises, mais également pour définir les processus dans lesquels les styleguides sont utilisés. UML a été choisi pour sa clarté et parce que c'est un standard reconnu. Le développement des BUC a été réalisé en trois itérations, à la fin de chacune d'elle une révision a été effectuée. Les BUC ont été modélisés avec l'outil Rational Software Modeler (RSM) de IBM.

3.1.7 Formalisierung der Anforderungen – Features und Requirements

Technique	Requirements engineering
Input	Findings, Personas, scénarios, littérature, recherche
Output	Catalogue de Features, catalogue des besoins (Requirements)
Description	Les Features permettent de définir ce que doit accomplir l'application à un haut niveau, ainsi que l'importance et la priorité de ces objectifs. Dans notre cas elles n'ont pas été définies au début avec un client, mais après l'analyse des utilisateurs. Car notre projet a un caractère de recherche, sans mandant. En plus des interviews, nous avons étudié les recommandations de la littérature afin de définir ces objectifs. Pour les formuler toutes les découvertes et autres informations ont été rassemblées, regroupées, agrégées pour former une dizaine d'objectifs principaux. Les Requirements sont la base de toute spécification. Les besoins fonctionnels et non fonctionnels ont été énoncés à partir des Findings et des idées personnelles.

3.1.8 Formalisierung der Anwendungsfälle – Use-Cases und Aktoren

Technique	Modélisation UML, System Use-Case
Input	Scénarios contextuels, Personas, Findings, Business Use-Case(BUC)
Output	Use-Case Model (UCM), System Use-Case (SUC)
Description	Les System Use Cases sont la clef de voûte du Framework RUP. Nous avons formalisé les tâches que doivent effectuer les acteurs sur l'application. Ce choix s'inscrit dans la continuité des BUC. Les UCM permettent d'avoir d'une part, une vue d'ensemble sur les fonctions de l'application et d'autre part, de montrer quels acteurs utilisent quels SUC. Grâce aux scénarios contextuels et aux BUC, nous avons posé et décrit les SUC. Seul deux SUC ont été décrits en détail, ceux qui ont été réalisés dans le prototype et qui font partie du scope. Les spécifications détaillées des SUC (cas principal et variantes) sont décrites sous forme textuelle, les diagrammes sont faits avec l'outil RSM de IBM.

3.1.9 Evaluation bestehender Lösungen – Konkurrenzanalyse

Technique	Analyse de la concurrence, analyse comparative (Benchmarking)
Input	Features, Requirements, Persona
Output	Tableau comparatif
Description	Pour toutes nouvelles applications, il est important de connaître l'état de la concurrence. C'est ce que nous avons cherché à savoir, une fois les objectifs de l'application clarifiés. Nous avons ainsi réalisé un Benchmarking des produits concurrents. Sept produits ont été évalués sur la base de onze critères, chacun pondérés de 0 à 5. Les critères sont issus des Features. La personne clé pour la décision de l'utilisation d'un outil pour les styleguides est le manager. C'est pourquoi la pondération a été définie du point de vue de la Persona Ottmar Owner (<i>Anhang B: Personas</i>). Chaque produit est noté sur une échelle de 0 à 5 en fonction des informations disponibles sur la page du constructeur. Aucun outil n'a été testé ou installé pour ce Benchmarking. Le score est calculé pour chaque produit selon la formule ci-contre, où S est le score du produit, P est la pondération du Critère i, C_i est la note obtenue pour le Critère i. $S = \sum_{i=1}^{11} P_i \cdot C_i$ Le score est ensuite recodé sur une échelle en pourcent grâce à la règle de trois ci-contre, où R est le résultat, S est le score du produit, M est le meilleure score parmi les produits (le Benchmark). $R = \frac{S \cdot 100}{M}$ Cette transformation permet de mieux comparer les concurrents. L'application de référence est notre projet „Online-Styleguide“. Il remplit en théorie tous les critères et obtient ainsi le score maximum de 100%. Les critères ainsi que leur pondération sont disponible en <i>Anhang E: Konkurrenzanalyse</i>.

3.1.10 Detailspezifikation der Applikation – Strukturdiagramm und Zustandsdiagramm

Technique	Spécification détaillée, modélisation UML
Input	Use-Case, littérature, Features et Requirements
Output	Diagramme de classe (structure), diagramme d'état
Description	Dans le cadre des spécifications détaillées nous avons utilisé deux diagrammes UML supplémentaires. Le diagramme de classe, pour mettre en valeur le modèle de données et sa structure, ainsi que le diagramme d'état, pour décrire le cycle de vie des styleguides. Comme les autres artefacts, ils ont été créés de manière itérative.

3.1.11 Evaluation des Konzeptes – Usability Testing

Technique	Usability Walkthrough
Input	Prototype Wireframe, scénarios de tests
Output	Listes de points négatifs et positifs
Description	Pour tester le prototype, nous avons mis en place des Usability Walkthrough. Cette autre technique de design centrée sur l'utilisateur nous permet de vérifier que le design est conforme aux attentes de nos Personas. Nous avons choisis et écrits les scénarios de tests, un pour chaque Persona (Analyste et Développeur). Les évaluations se sont déroulées toujours par paire, une personne pour prendre les notes, une autre pour simuler l'ordinateur et diriger le test. En tout cinq évaluations ont eu lieu, d'une durée d'une heure. La répartition des utilisateurs par Persona était de quatre pour les développeurs et une personne pour l'analyste. Les résultats des évaluations sont intégrés directement dans la thèse au chapitre 8 <i>Der Prototyp im Test – Evaluation</i>.

3.2 Ethische Aspekte des Vorgehens

Qualitative Erhebungen von Daten ist ein wesentlicher Aspekt der UCD-Methodologie. Interviews, Contextual Inquiries oder Usability-Walkthroughs sind einige der Techniken, bzw. Aktivitäten zur qualitativen Erhebung von Daten. Für diese Aktivitäten ist der direkte Einbezug von Personen als Teilnehmer von Untersuchungen jedoch unabdingbar.

Bei der Durchführung von Aktivitäten die Teilnehmer involvieren, spielen juristische und ethische Aspekte eine zentrale Rolle (Courage, et al., 2005 S. 95-104). Der Durchführende ist einerseits für die physische und psychische Unversehrtheit der Teilnehmer, andererseits ist auch für den Schutz der Firma vor Klagen, seitens verärgelter Teilnehmer, verantwortlich. Ausserdem ist er für die Korrektheit und Verlässlichkeit der erhobenen Daten verantwortlich. Der beste Weg, dieser vielschichtigen Verantwortung gerecht zu werden, ist die Einhaltung ethischer Richtlinien.

Ethische Richtlinien

Für unsere Masterarbeit, insbesondere für Aktivitäten, die Teilnehmer involvierten, haben wir uns, in Anlehnung an Courage et al. (2005 S. 97-102), an folgende ethische Richtlinien gehalten:

Richtlinie	Beschreibung
Das Recht auf Information	Vor einer Untersuchung haben wir potentielle Teilnehmer schriftlich über die jeweilige Untersuchung und den Ablauf informiert. Nach erfolgter Zusage haben wir die Teilnehmer, kurz vor Beginn der anstehenden Aktivität, nochmals mündlich über deren Sinn und Zweck informiert.
Das Recht zum Abbruch	Die Teilnehmer wurden aufgeklärt, dass sie jederzeit, auch während einer laufenden Aktivität, das Recht zum Abbruch haben.
Erlaubnis für Aufzeichnungen	Vor einer Aufzeichnung haben wir die Teilnehmer explizit nach ihrem Einverständnis gefragt.
Erschaffen einer angenehmen Erfahrung	Wir sind, soweit möglich, auf die Wünsche der Teilnehmer bezüglich Ort und Zeit eingegangen. Wir haben auf einen respektvollen und höflichen Umgang während den Erhebungen geachtet.
Anonymität	Wir haben den Teilnehmern absolute Vertraulichkeit und Anonymität garantiert. Bei der Durchführung der Analysen wurden Pseudonyme (z.B. fiktive Namen bei Personas) und Identifikationsnummern verwendet. Somit ist kein direkter Rückschluss zu den jeweiligen Identitäten der Teilnehmer möglich. Dasselbe Prinzip haben wir auch für sämtliche erstellten Artefakte angewendet.
Angemessene Entschädigung	Wir haben die Teilnehmer für Ihren Aufwand entsprechend honoriert. Dies geschah beispielsweise anhand eines „Dankeschön“ in Form von Pralinen.
Gültige und verlässliche Daten	Bei den Untersuchungen haben wir stets darauf geachtet, keine Suggestivfragen zu stellen oder die Teilnehmer auf andere Weise in ihren Entscheidungen oder Denkweisen zu beeinflussen. Jeweilige Informationen wurden „ungeschminkt“ übernommen und nach bestem Wissen und Gewissen ausgewertet.
Aufbewahrung der Daten	Die Rohdaten werden nur solange als nötig aufbewahrt. Mit Abschluss dieses Projektes werden die Rohdaten gelöscht.

Tabelle 12: Ethische Richtlinien im Vorgehen

4 Die Suche nach Erkenntnissen – Recherchen

Die Durchführung umfassender Recherchen ist für den Erfolg eines Projektes entscheidend. Sinn und Zweck unserer Recherchen war es, möglichst viele Erkenntnisse über unterschiedliche Aspekte von Styleguides zu gewinnen. Allerdings nicht nur über Styleguides selbst, sondern ebenfalls über das Umfeld, in dem Styleguides eingesetzt werden, welche Prozesse dabei eine Rolle spielen und welche Personen involviert sind.

Zum Erlangen dieser Erkenntnisse haben wir zwei Wege verfolgt. Einerseits haben wir verschiedene Quellen in der Literatur untersucht und analysiert, andererseits haben wir empirische Recherchen mit verschiedenen Personen in Form von Interviews durchgeführt.

Dieses Kapitel gliedert sich daher in zwei Teile.

Der erste Teil beleuchtet die in der Literatur beschriebenen Problembereiche im Zusammenhang mit Styleguides. In diesem Abschnitt werden ebenfalls Lösungen und Empfehlungen aus der Literatur zu diesen Problembereichen präsentiert.

In einem zweiten Teil werden die aus den empirischen Recherchen gewonnenen Daten aufgezeigt und anhand von Business Use-Cases, Personas und Fazits aus den Untersuchungen beschrieben.

4.1 Recherchen in der Literatur

Laut einer Umfrage von Quesenbery (2001b), werden Styleguides in Unternehmen mehrheitlich mit Erfolg eingesetzt. Quesenbery bemisst diesen Erfolg anhand folgender Punkte, die für die Mehrheit der befragten Teilnehmer zutrifft:

- Der Styleguide hat die in ihn gesetzten Ziele erfüllt.
- Der Styleguide ist zu einem täglich benutzten Werkzeug geworden.
- Der Styleguide ist eine respektierte und anerkannte Quelle für Informationen.
- Die Benutzeroberflächen sind konsistenter geworden.
- Die Benutzeroberflächen sind benutzbarer geworden.
- Es herrscht Übereinstimmung mit den Guidelines, die im Styleguide enthaltenen sind.

Styleguides also ein voller Erfolg?

Quesenbery betont allerdings, dass die Kommentare der Umfrageteilnehmer grosse Schwierigkeiten widerspiegeln. Diese Schwierigkeiten reichen von der Integration von Styleguides im Entwicklungsprozess, über mangelnden Gebrauch seitens der Entwickler, bis hin zu mangelhafter Unterstützung seitens des Managements für den Einsatz von Styleguides. Dies legt den Schluss nahe, dass sehr viele Faktoren einen Einfluss darauf haben, ob Styleguides in einem Unternehmen letztendlich mit Erfolg eingesetzt werden können.

Im Zuge der Erstellung eines Styleguides ist es für viele Unternehmen das erste Mal, dass sie mit der Thematik von HCI und Design konfrontiert sind. Obwohl der Einsatz von Styleguides viele Vorteile mit sich bringt (*Tabelle 6: Vorteile von Styleguides (Gale, 1996)*), gibt es eine Reihe von Fallstricken, denen es auszuweichen gilt (Gale, 1996). Diese Fallstricke bezeichnet Gale mit dem Ausdruck „*The style guide problem*“. Laut Gale haben schon viele Unternehmen mehrmals versucht das „Styleguide-Problem“ zu lösen und er listet mehrere Gründe auf, die zu ihrem Scheitern geführt haben.

In der Literatur befassen sich allerdings auch zahlreiche andere Quellen mit den Gründen, die letztendlich zu einem Scheitern von Styleguides führen können. Diverse Autoren bieten aber auch Vorschläge an, mit denen Schwierigkeiten umschifft und Probleme gelöst werden können.

Es gibt nicht *das* Problem und *die* Lösung

Probleme und Lösungen lassen sich nicht isoliert voneinander betrachten. Probleme stehen in Beziehung zueinander und können somit entweder ein Auslöser oder selbst eine Konsequenz anderer Probleme sein. Ebenso verhält es sich mit Lösungen. Manche Lösungen sind übergreifend und bieten eine Lösung verschiedener, voneinander abhängiger Probleme.

Nicht alle Probleme sind lösbar

Anzumerken ist im Weiteren, dass einige Probleme intrinsisch sind. Das heisst, es handelt sich dabei um Probleme, die in der Natur von Styleguides oder generell von Guidelines selbst liegen. Deshalb gibt es für diese Art von Problemen auch keine Lösungen. Der Vollständigkeit halber werden sie aber dennoch aufgeführt.

4.1.1 Klassifizierung der Probleme und Lösungen in Themenbereiche

In der Literatur existieren zahlreiche Quellen, die sich mit Problemen und Lösungen in Bezug auf Styleguides auseinandersetzen. Einige Autoren setzen sich mit deren Inhalt auseinander, andere wiederum befassen sich mit Themen bezüglich der Kollaboration und zum Einführungsprozess von Styleguides in Unternehmen. Einige Autoren fokussieren sich auf die Benutzung von Styleguides, während andere wiederum die Struktur von Styleguides in den Mittelpunkt rücken.

Um einen Überblick über die verschiedenen Themenbereiche, zu denen Probleme und Lösungen beschrieben werden, zu erhalten, ist eine Klassifizierung notwendig. Eine mögliche Klassifizierung dieser Themenbereiche wird von Vogt (2001) präsentiert. Nach Durchsicht diverser Quellen kommt Vogt zum Schluss, dass sich die Probleme und Lösungen in zwei Themenbereiche einteilen lassen. Der erste Bereich betrifft Themen in Bezug auf den Inhalt von Styleguides. Der zweite Bereich betrifft Themen in Bezug auf die „Handhabung“ von Styleguides als Dokumente. Unter Handhabung werden dabei Themen zu Prozessen, Verantwortlichkeiten und Kollaboration verstanden.

Allerdings gibt es noch eine Reihe weiterer Autoren, die sich zu Problembereichen äussern und Lösungen bieten, die von Vogt nicht behandelt werden. In diesem Zusammenhang ist es wichtig zu erwähnen, dass sich einige Autoren nicht direkt mit Styleguides, sondern mit Guidelines generell auseinandersetzen. In Kapitel 2.3.1.2 *Klassifizierung nach Quelle* wurde gezeigt, dass Styleguides eine Sammlung von Guidelines sind, bzw. auf Guidelines basieren. Daher gelten angesprochene Probleme und Lösungen bezüglich Guidelines ebenfalls für Styleguides.

Auf der Basis des Klassifizierungsschemas von Vogt, wird eine erweiterte Klassifizierung verwendet. Probleme und Lösungen zu Styleguides lassen sich grundsätzlich in vier Themenbereiche gliedern:

Themenbereich	Beschreibung
Allgemeines	Betrifft Probleme und Lösungen, die sich allgemein mit Styleguides als „Konzept“ oder „Idee“ auseinandersetzen.
Inhalt	Darunter sind Probleme und Lösungen zu verstehen, die sich mit dem Inhalt von Styleguides auseinandersetzen.
Struktur und Form	Darunter sind Probleme und Lösungen zu verstehen, welche die Struktur und Organisation von Informationen in Styleguides, sowie dessen Form (z.B. online, auf Papier) betreffen.
Prozesse und Kollaboration	Betrifft Probleme und Lösungen, die in Bezug zu Erstellungs- und Einführungsprozessen, der Integration in Entwicklungsprozessen, der Zusammenarbeit verschiedener Stakeholder und die Verwendung von Styleguides stehen.

Tabelle 13: Klassifizierung der Probleme und Lösungen in Themenbereiche

Die nachfolgenden Kapitel beschreiben die Probleme und Lösungen innerhalb der oben definierten Themenbereiche in tabellarischer Form. In den jeweiligen Tabellen steht „P“ für erwähnte Probleme und „L“ für jeweilige Lösungsvorschläge entsprechender Autoren.

Probleme und Lösungen zu den Themenbereichen sind zusätzlich mit **Referenzen** versehen ([LIT-XX-X]). Darauf wird insbesondere in Kapitel 5 *Die Zusammenfassung der Erkenntnisse – Aspekte, Features und Konkurrenzanalyse* Bezug genommen. Die nachfolgend aufgeführten Lösungsvorschläge bilden auch einen Teil der Empfehlungen zum Einsatz von Styleguides, die in Kapitel 9 *Der Einsatz projektspezifischer Styleguides – Guidelines* gegeben werden.

4.1.2 Themenbereich: Allgemeines

	Usability und Konsistenz	LIT-AP-1
P	Styleguides sind ein Mittel, um die Konsistenz eines Produktes sicherzustellen (Mariage, et al., 2004). Quesenbery (2001a) und Gale (1996) betonen aber, dass die Einhaltung von Konsistenz alleine nicht genügt, um benutzbare Applikationen zu erstellen. Quesenbery beschreibt dies folgendermassen: <i>„All too often, businesses ask for a style guide as a means to create a common look and feel, in the belief that it will solve usability problems and establish consistency between applications – only to be disappointed in the results. Even if such a style guide is followed carefully, the resulting interfaces may not meet usability goals.“</i>	
	Durch die Einhaltung von Styleguides ist zwar die Konsistenz einer Applikation sichergestellt, jedoch nicht, dass sie die relevanten Funktionalitäten aufweist, die von Endbenutzern gefordert und erwartet werden (Gale, 1996).	
	Gale (1996) und Wilson (2001) kritisieren, dass viele Firmen in Styleguides oft die Lösung sämtlicher Usability-Probleme sähen. Styleguides sind aber kein Ersatz für iteratives Design und Usability-Tests.	
L	Laut Gale (1996) liegt die Sicherstellung der Usability, die über die Grenzen eines Styleguides hinausgehen, in der Verantwortung der entsprechenden Geschäfts- und IT-Abteilungen. Es ist daher eine Frage der richtigen Prozesse und Vorgehensweisen.	
	Wilson (2001) schlägt vor, dem Management nachdrücklich klar zu machen, dass ein Styleguide nicht die Lösung sämtlicher Usability-Probleme ist.	

	Generalisierbarkeit	LIT-AP-2
P	Myers (1993) beschreibt, dass die Geschichte von HCI zeigt, dass sich Guidelines nicht unbedingt generalisieren lassen. Das bedeutet, dass einige Guidelines nur für spezifische Situationen anwendbar sind, nicht aber generell gelten. Vogt (2001) zieht daraus den Schluss, dass, wenn sich Guidelines nicht generalisieren lassen, die Grundlage „generischer Styleguides“ in Frage gestellt ist.	
L	Für diese Problematik existiert in der Literatur keine spezifische Lösung. Myers (1993) Aussage ist, dass sich dieses Problem in der nächsten Zukunft nicht lösen lassen kann.	

4.1.3 Themenbereich: Inhalt

	Fokus, Detaillierungsgrad und Kontext	LIT-IN-1
P	Probleme können entstehen, wenn Styleguides entweder zu detailliert sind oder versuchen, zu vielen verschiedenen Applikationstypen oder Benutzern gerecht zu werden. Dies führt dazu, dass der Fokus des Styleguides unklar wird und deshalb für eine Reihe von Applikationen nicht geeignet ist (Gale, 1996).	
	Vogt (2001) beschreibt, dass einige Styleguides, besonders solche für Betriebssysteme, zu detaillierte Beschreibungen von Komponenten enthalten, die sowieso nicht von einem Entwickler beeinflusst werden können oder sollen. Ausserdem bestehen einige Styleguides nur in einer Auflistung von Komponenten eines Interfaces, es fehlen aber Beschreibungen bezüglich des Nutzungskontextes dieser Komponenten.	
	Auch Quesenbery (2001a) beschreibt, dass viele Styleguides derart formal beschrieben sind, dass man nicht mehr erkennt, für welchen Kontext darin enthaltene Guidelines eigentlich anzuwenden sind.	
L	Vogt (2001) präsentiert folgende Lösungsvorschläge: • Ein Styleguide sollte definieren, welchen Zweck er verfolgt und wofür er nicht gedacht ist. • Allzweck-Styleguides sind nicht praktikabel, deshalb sollen Styleguides spezifisch für einen bestimmten Einsatzzweck entwickelt werden. Dies reduziert auch deren Umfang. • Anstatt bekannte Komponenten einer Benutzeroberfläche im Detail zu beschreiben, sollten Styleguides auf deren Nutzungskontext und Kombination mit anderen Komponenten eingehen. Um einem Designer zu helfen die richtigen Entscheidungen zu treffen, sollte ein Styleguide den Kontext für die Verwendung von Designelementen beschreiben. Ein Styleguide sollte Vergleichstabellen für Optionen eines Designelementes (z.B. Vor- und Nachteile von Tabellen/Listen) enthalten (Quesenbery, 2001a).	
	Aus Styleguides sollte klar hervorgehen, welche Guidelines befolgt werden müssen und welche nur eine Empfehlung sind (Wilson, 2001).	

Konflikte und Korrektheit	LIT-IN-2
P Vogt (2001) beschreibt verschiedene Arten von Konflikten, die bei der Benutzung von Styleguides auftreten können. Beispielsweise können für ein spezifisches Design- oder Usabilityproblem mehrere in einem Styleguide beschriebene Prinzipien in Frage kommen, die alle für diesen Fall gültig sind. Der Konflikt besteht dadurch, dass aus dem Styleguide nicht klar ersichtlich wird, welches Prinzip zur Lösung des Problems gewählt werden soll. Tidwell (1999) beschreibt dies folgendermassen: <i>„No style guide can infallibly tell you how to strike a balance between two opposing high-level principles“</i>. Aber auch wenn keine Konflikte im Styleguide existieren, kann man ihm nicht blind vertrauen (Vogt, 2001). Die Basis für diese Aussage liefert Vanderdonckt (1999), der beschreibt, dass einige Styleguides auf fehlerhaften oder umstrittenen Guidelines basieren. Laut Spool (2002) existieren viele Guidelines, die nicht getestet und somit empirisch belegt sind. Die Befolgung einer solchen Guideline kann sogar zu einer Verschlechterung der Usability führen. Spool sagt dies folgendermassen: <i>„This means that following untested guidelines is like drinking water from an unidentified source. It might quench your thirst, but it could also make you very ill.“</i> L Laut Vogt (2001) sind diese Probleme ein intrinsisches Problem wofür es keine Lösung gibt.	
Inkonsistenz	LIT-IN-3
P Ein Problem kann aufgrund widersprüchlicher oder inkonsistenter Information innerhalb eines Styleguide entstehen. Inkonsistenzen in einem Styleguide beschädigen dessen Glaubwürdigkeit (Wilson, 2001). L Wilson (2001) schlägt als Lösung vor, einen Styleguide sorgfältig zu reviewen und sicherzustellen, dass alle darin enthaltenen Beispiele konsistent zu den entsprechenden Guidelines sind.	
Unvollständigkeit	LIT-IN-4
P Laut Vanderdonckt (1999) ist jede Sammlung von Guidelines unvollständig. Ein Styleguide kann deshalb keine abschliessenden Antworten für sämtliche Fragestellungen liefern. L Die Unvollständigkeit eines Styleguides ist laut Vanderdonckt (1999) ein intrinsisches Problem, wofür es keine Lösung gibt.	
Vergänglichkeit	LIT-IN-5
P Der schnelle Fortschritt technischer Möglichkeiten in Bezug auf Betriebssysteme, Entwicklungswerkzeuge, Plattformen, Programmiersprachen und Schnittstellen, führt dazu, dass Styleguides schnell veralten, da neue Trends nicht schnell genug berücksichtigt werden (Tidwell, 1999). L Laut Vogt (2001) gibt es für dieses Problem keine Lösung, da nicht zu erwarten ist, dass sich der Fortschritt der Technologien oder Werkzeuge verlangsamt.	
Illustrationen und „Negativbeispiele“	LIT-IN-6
P Laut Vanderdonckt (1999) sind Guidelines mangelhaft illustriert. Nach Einschätzung von Vogt (2001) lässt sich diese Kritik allerdings nicht generalisieren. Einige Guidelines lassen sich nur sehr schwer illustrieren. Beispielsweise wenn sie sich nicht auf visuelle Elemente beziehen. Manche Styleguides sind sehr gut illustriert, während andere wenig bis gar keine Illustrationen aufweisen. Viele Styleguides weisen einen Mangel an Negativbeispielen auf. Negativbeispiele zeigen, wie man etwas <i>nicht</i> machen sollte. Der Vorteil von Negativbeispielen ist, dass man sich leicht an schlechten Beispielen orientieren kann. Dadurch können Fehler auf einem User-Interface schneller identifiziert und dadurch vermieden werden. Negativbeispiele folgen dem Prinzip, dass man aus Fehlern lernt (Vogt, 2001). L Dieses Problem ist laut Vogt (2001) gut zu lösen, da Illustrationen einfach hinzugefügt werden können. Sie müssen aber mit Sorgfalt erstellt werden, da sie konform zum gesamten Styleguide selbst sein müssen und nicht nur für eine bestimmte Guideline, die sie illustrieren. Aus der Illustration selbst muss klar hervorgehen, ob es sich dabei um eine Illustration eines „Negativbeispiels“ handelt oder nicht.	

Sinnhaftigkeit und Zweckmässigkeit		LIT-IN-7
P	Der Sinn und Zweck von Prinzipien oder Regeln, die in einem Styleguide beschrieben sind, ist nicht immer klar (Vogt, 2001). Auch laut Quesenbery (2001a) wird in einem Styleguide oftmals die „Essenz“ eines Designelementes oder eines Interaktionsstils nicht gut genug vermittelt. Solche Styleguides sind üblicherweise schwierig zu interpretieren und umzusetzen (Myers, 1993). Beispielsweise wird in (Oracle, 2001) eine Guideline beschrieben, die sich auf den Abstand zwischen Label und Textfeld bezieht: <i>„Insert 12 pixels between the trailing edge of a label and the component it describes when labels are right-aligned. When labels are left-aligned [...]“</i> .	
	Die Begründung, weshalb der Abstand genau zwölf Pixel und nicht zehn oder elf betragen soll, ist daraus nicht ersichtlich. Laut Vogt führt dies dazu, dass ein Entwickler eine solche Guideline einfach ignoriert und nach Gutdünken verfährt.	
L	Quesenbery (2001a) schlägt folgende Lösungen vor: • Gib Designelementen eine Begründung: Eine Guideline soll begründen, weshalb sie die Benutzbarkeit fördert und was ihr Zweck ist. • Verbinde Guidelines mit Benutzeranalysen: Falls Ergebnisse aus Benutzeranalysen oder Usability-Tests vorliegen, welche eine Guideline untermauern, sollten sie im Styleguide integriert werden.	

4.1.4 Themenbereich: Struktur und Form

Index und Querverweise		LIT-SF-1
P	Ein mangelhafter Index ist ein häufiges Problem bei Styleguides. Es gibt nicht genügend Indexbegriffe oder Querverweise (Wilson, 2001).	
L	Wilson (2001) empfiehlt zur Lösung der Probleme zum Index und Querverweisen den Gebrauch mehrerer Synonyme für „Objekte“ im Styleguide. Man sollte sich für die Erstellung eines Index Zeit nehmen und die Hilfe eines professionellen Indexierers in Anspruch nehmen.	

Umfang		LIT-SF-2
P	Wilson (2001) und Gale (1996) beschreiben beide den Umstand, dass der Umfang von Styleguides teilweise sehr gross ist und hunderte von Seiten enthalten kann. Solche Styleguides sind in aller Regel schwierig zu benutzen, zu lesen und zu durchsuchen. Gale beschreibt weiter: <i>„It is ironic that many of the existing Style Guides, aimed at producing usable systems, are themselves difficult to use“</i> .	
L	Wilson (2001) schlägt zur Lösung des Umfangs von Styleguides vor, sich nur auf die wirklich kritischen Bereiche zu beschränken. Ein Styleguide sollte nicht versuchen jede einzelne Thematik zu behandeln.	

Form		LIT-SF-3
P	Einer der grössten Fehler ist es, einen Styleguide als ein Sachbuch anstatt eines Nachschlagewerkes zu betrachten. Nur Autoren und Reviewer werden den ganzen Styleguide durchlesen, alle anderen Benutzer werden nach Informationen suchen, wann immer sie sie benötigen (Quesenbery, 2001a). Laut Hart (2000) sind gedruckte Styleguides zu statisch und nutzen die Vorteile moderner Computertechnologien nicht aus. Diese statischen Styleguides, insbesondere sehr umfangreiche, sind schwierig zu benutzen und lassen sich schlecht in Firmenprozesse integrieren.	
L	Quesenbery (2001a) präsentiert folgende Lösungsvorschläge: • Stelle den Styleguide online zur Verfügung: Es soll von Anfang an geplant werden, einen Styleguide online zur Verfügung zu stellen. • Setze Hyperlinks ein: Mittels Hyperlinks sollen Inhalte und Beispiele miteinander verlinkt werden. • Erstelle Referenzierungen: Themen sollten so organisiert werden, dass die wichtigsten Informationen jeweils am Anfang stehen. Jede Thematik soll nur auf einer einzigen Hypertextseite abgehandelt werden und mittels Hyperlinks auf andere Themen verweisen. Hart (2000) plädiert ebenfalls für den Einsatz moderner Computertechnologie zur Überwindung von statischen Styleguides. Damit liessen sich Styleguides besser in Firmenprozesse einbinden.	

Struktur und Organisation**LIT-SF-4**

- P Laut Quesenbery (2001a) ist es, ohne zu wissen wofür und für wen ein Styleguide erstellt wird, unmöglich eine geeignete Struktur und entsprechende darunterliegende Inhalte zu erstellen. Eines der Probleme ist, dass Styleguides die verschiedenen Benutzergruppen (Abb. 10: Benutzergruppen von Styleguides (Quesenbery, 2001a)) nicht berücksichtigen. Diese haben unterschiedliche Ansprüche an den Inhalt von Styleguides, die sich zusätzlich noch in Abhängigkeit der Projektphase ändern:

User Group	Design Concept - Analysis Stages	Detailed Design - QA Stages
All	What are the rules I must follow in my work?	Have I followed those rules?
UI Designers (Architecture or Concept)	How should the interface be structured, both for layout and navigation?	Does the interface fit any structural guidelines?
System Architects	What are the implications for the architecture of any UI decisions?	Can all of the UI requirements be met?
UI Designers (Detailed Functions)	How do individual functions fit into the program structure? What screen/page or interaction templates are available?	What are the rules for layout, color, etc. that the design must support? How should design elements be specified?
Developers	How should the UI code be structured to meet design requirements? What controls or widgets are needed?	What color, styles, etc. are available to use? How can I match them to the design specifications?
User Assistance	How will assistance be offered to users? What standard UI elements are used? How should they be documented?	What assistance does a user need for this function/page/screen? What method should be used to include it?
Quality Assurance	Are there templates or standard approaches which can be tested globally?	Does the interface meet all standards?

Abb. 13: Ansprüche verschiedener Benutzergruppen an Styleguides (Quesenbery, 2001a)

Vanderdonckt (1999) sieht ebenfalls ein Problem in Bezug auf die Organisation von Inhalten. Verschiedene Rollen in der Entwicklung haben unterschiedliche Ansprüche an diese Organisation:

- **Task Analyst:** Bevorzugt eine Organisation nach Aufgaben, z.B. „Finden von Information“, „Form Fill-In“, „Multi-Criteria Search“.
- **Projektleiter:** Interessiert sich für *Highlevel*-Guidelines und verwendete Metaphern eines Interfaces.
- **Ergonome:** Bevorzugt eine Organisation anhand kognitiver Prinzipien, von denen Guidelines abgeleitet sind.
- **Designer:** Erwartet eine Organisation in einer Weise, die es ihm erlaubt, Guidelines schnell zu isolieren, um Konflikte zwischen ihnen zu lösen.
- **Programmierer:** Bevorzugt eine Organisation nach Widgets.
- **UI-Tester:** Bevorzugt Guidelines, die als messbare Größen formuliert sind.

Vanderdonckt (1999) zieht daraus folgenden Schluss: „*We recall that any guidelines organisation remains impossible to simultaneously address all demands for all task types.*“.

- L In der untersuchten Literatur gibt es keine Lösungsvorschläge, die sich spezifisch mit der Problematik der Strukturierung und Organisation der Information auseinandersetzen. Allerdings bieten Quesenbery (2001a) und Gale (1996) jeweils ein „Template“ eines Styleguides.

4.1.5 Themenbereich: Prozesse und Kollaboration

Lösungsvorschläge zur Problematik bezüglich Prozessen und Kollaboration verfolgen ganzheitliche Ansätze. Deshalb lassen sich die Lösungsvorschläge nicht entsprechend der unterschiedlichen Problembereiche gliedern, ohne sie aus dem Kontext zu reißen. Nachfolgend werden deshalb nur die jeweiligen Probleme aufgeführt. Die Lösungsvorschläge werden im Anschluss darauf beschrieben.

4.1.5.1 Problembereiche

Einführungsprozesse

LIT-PK-1

Laut Gale (1996) ist eines der Probleme, dass kaum Überlegungen angestellt werden, wie Styleguides im Unternehmen eingeführt werden sollen. Es gibt kaum Unterstützung und keinen Ansporn sie einzusetzen. Bei der Einführung von neuen oder geänderten Styleguides wird kaum Überzeugungsarbeit geleistet, insbesondere nicht bei den betroffenen Personen (Johnson, 2010). Wilson (2001) kritisiert ebenfalls, dass bei der Einführung von Styleguides kaum dafür geworben wird.

Entwicklungsprozesse

LIT-PK-2

Gale (1996) beschreibt, dass viele Styleguides nicht als Teil eines Entwicklungsprozesses integriert sind. Daher berücksichtigen sie auch nicht die verschiedenen Verwendungsarten von Styleguides.

Management

LIT-PK-3

Wilson (2001) kritisiert, dass das Management eines Unternehmens häufig keine genauen Vorstellungen der Vorteile von Styleguides haben und ebenfalls keine Vorgaben erlässt, die das Einhalten von Styleguides fordert.

Aber auch wenn Vorgaben zum Einsatz von Styleguides von höchster Stelle kommen, ist ein Erfolg nicht garantiert. Aufgezwungene Styleguides, die nicht auf Überzeugung seitens deren Benutzer stossen, werden deshalb nur halbherzig oder gar nicht erst angewendet (Quesenbery, 2001a). Gale (1996) drückt dies folgendermassen aus: *„The most effective Style Guides are not those that are imposed on developers by senior management, but those that have been developed with the developers themselves so that they perceive the Style Guide as being theirs.“*

Verantwortlichkeiten

LIT-PK-4

Durch die Erstellung und Einführung von Styleguides sind die Verantwortlichkeiten für die Usability der Applikationen nicht mehr klar gegeben. Die Entwicklungsteams nehmen an, dass dadurch jemand anders die Verantwortung für die Usability ihres Systems übernimmt und nicht mehr sie selber dafür zuständig sind. Dies trifft vor allem zu, wenn Styleguides von einer externen Partei entwickelt werden, welche die Entwickler nicht mit einbeziehen. Den Entwicklern fehlt dann häufig eine Bezugsperson, welche Fragen beantwortet und den Styleguide, falls nötig, anpasst (Gale, 1996).

Kollaboration und Akzeptanz

LIT-PK-5

Gale (1996) kritisiert, dass Styleguides oft ohne Einbezug derer Benutzer und ohne Berücksichtigung des Entwicklungsumfeldes, in denen sie eingesetzt werden sollen, erstellt werden. Styleguides können dadurch für die Benutzer unpraktisch zu benutzen und für das Entwicklungsumfeld nicht integrierbar werden. Entwickler sträuben sich gegen eine Verwendung von Styleguides, bei denen sie nichts beitragen konnten.

Wilson (2001) führt weiter aus, dass es keinen einfachen Weg gibt, Widersprüche in einem Styleguide zu diskutieren und zu klären.

Aktualisierungen und Änderungen

LIT-PK-6

Laut Gale (1996) ist eines der Probleme, dass Styleguides nicht aktualisiert werden. Entweder weil keiner realisiert, dass Styleguides aktualisiert werden müssen oder weil man nicht weiss, wer dafür verantwortlich dafür.

Wilson (2001) kritisiert, dass es keine gute Methode gibt, Benutzer über Änderungen oder Aktualisierungen von Styleguides aufmerksam zu machen.

Umsetzbarkeit

LIT-PK-7

Probleme können dadurch entstehen, dass Styleguides die Entwicklungsplattform, auf der eine Applikation entwickelt wird, nicht berücksichtigen. Dadurch können Prinzipien und Standards, die in einem Styleguide beschrieben sind, nicht umgesetzt und implementiert werden (Wilson, 2001).

4.1.5.2 Lösungsvorschläge nach Autoren

Vorschläge von Quesenbery (2001a)

LIT-QUES

Wenn ein Styleguide effektiv sein soll, sollte er nicht nur gute Guidelines enthalten, sondern muss ebenfalls benutzt werden und akzeptiert sein. Der Prozess, den es dafür braucht, ist genauso wichtig wie das eigentliche Resultat – der Styleguide selbst. Quesenbery macht folgende Vorschläge für den Prozess:

- **Starte früh:** Damit alle relevanten Stakeholder eines Designs involviert werden, sollten Styleguides schon früh im Entwicklungsprozess erstellt werden.
- **Mache die entstehende Arbeit den Benutzern zugänglich:** Ein Styleguide sollte schon während dessen Entstehung breit verfügbar gemacht werden. Dies ermöglicht Benutzern eine Einflussnahme in Form von Kommentaren und Fragen. Dadurch steigt die Akzeptanz eines Styleguides. Um jedoch zu vermeiden, dass unfertige Styleguides vorzeitig eingesetzt werden, müssen sie einen klaren Status aufweisen.
- **Mache den Prozess transparent:** Diskussionsthemen und Kommentare sollten im Styleguide integriert sein, damit die Stakeholder den Verlauf der Arbeit mitverfolgen können. Auch frühe Konzepte oder Varianten sollen sichtbar sein, sollten aber klar als solche gekennzeichnet werden.
- **Mache den Kontext sichtbar:** Es sollte aus einem Styleguide klar hervorgehen, in welchem Kontext er eingesetzt werden soll, bzw. was dessen Sinn und Zweck ist.

Vorschläge von Gale (1996)

LIT-GALE

Gale beschreibt einen fünfstufigen Ansatz eines *kollaborativen* Prozesses zur Erstellung von Styleguides:

- **Sensibilisierung der Entwickler und Endbenutzer:** Damit Sinn und Zweck des Styleguides von Entwicklern und Endbenutzern einer Applikation verstanden wird, sollen sie, in Zusammenarbeit mit dem Management, dafür sensibilisiert werden.
- **Konsensbildung:** Von einer Arbeitsgruppe, bestehend aus Entwicklern und Endbenutzern, sollen Kommentare und andere Inputs zum Styleguide eingeholt werden. Damit werden sie Teil des Prozesses.
- **Dokumentieren des Styleguides:** Erst *nachdem* ein bestimmter Grad an Konsens erreicht worden ist, soll der Styleguide dokumentiert werden. Dies soll die Akzeptanz und das Verständnis des Styleguides bei den Entwicklern sicherstellen und dafür sorgen, dass die besten Ideen in den Styleguide einfließen.
- **Training und Unterstützung:** Styleguides sollen parallel mit Training und „unterstützendem Material“ eingeführt werden. Die Arbeitsgruppe, die den Styleguide erstellt hat, soll beides anbieten.
- **Erschaffen eines Umfeldes, das eine Weiterentwicklung des Styleguides ermöglicht:** Ein Styleguide ist kein statisches Dokument. Es soll deshalb ein Umfeld erschaffen werden, in dem es möglich ist, einen Styleguide mit der Zeit weiter zu entwickeln (z.B. um neue Ideen oder Anforderungen zu integrieren).

Gale beschreibt eine Reihe weiterer Aspekte, die als Teil seines Gesamtansatzes zu verstehen sind:

- **Integration in den Entwicklungsprozess:** Damit ein Styleguide erfolgreich ist muss er den Entwicklungsprozess, in dem er verwendet wird, berücksichtigen.
- **Von Anfang an einsetzen:** Oft werden Styleguides lediglich gebraucht um die Konformität eines User-Interfaces zu überprüfen. Obwohl dies eine legitime Verwendung ist, ist es besser, einen Styleguide *von Anfang an* für ein neues Projekt zu benutzen.
- **Rahmen des Anwendungsbereiches:** Der Anwendungsbereich des Styleguides muss auf die relevanten Plattformen und Funktionalitäten beschränkt sein. Es ist unmöglich einen Styleguide zu erstellen, der für alle möglichen Systeme und Funktionalitäten verwendet werden kann.
- **Kollaboration mit den Entwicklern:** Ein Styleguide muss zusammen mit den Entwicklern erstellt werden. Dies stellt sicher, dass ein Styleguide benutzbar und umsetzbar ist. Dies steigert die Akzeptanz für dessen Gebrauch. Wenn Entwickler realisieren, dass Styleguides tatsächlich Produkte verbessern können, werden sie auch für Folgeprojekte eingesetzt.
- **Nicht nur ein Dokument:** Ein Styleguide sollte von verschiedenen unterstützenden „Kommunikationsmitteln“ begleitet werden. Dies können beispielsweise interaktive Demonstrationen sein, Checklisten zur Überprüfung der Konformität, Referenztabellen zum schnellen Nachschlagen von Informationen, wieder-verwendbarer Quellcode und Wettbewerbe zum Entdecken von Fehlern auf User-Interfaces.
- **Verantwortlichkeiten sicherstellen:** Es muss klar sein, wer für die Wartung und Aktualisierung eines Styleguides verantwortlich ist. Dies kann eine Gruppe oder eine einzelne Person sein.

Vorschläge von Wilson (2001)

LIT-WILS

Wilson gibt folgende Ratschläge für die Lösung von Problemen bezüglich Prozessen und Kollaboration:

- Um die Aufmerksamkeit für einen Styleguide zu wecken, sollte dessen Einführung werbemässig geplant werden. Man sollte Styleguideposter aufhängen und eine „Party“ feiern, dabei sollten die Vorteile von Styleguides klar in Erscheinung treten.
- Das Management sollte Vorgaben erlassen, welche die Einhaltung von Konsistenz fordert.
- Es sollten Foren erstellt werden, um einen breiten Review eines Styleguides zu unterstützen.
- Es sollte ein einfacher Weg angeboten werden, Konflikte bezüglich der Inhalte eines Styleguides zu diskutieren. Dies kann beispielsweise ein Forum sein, wo Probleme für eine gewisse Zeit besprochen werden können. Danach wird vom Styleguide-Team eine Entscheidung gefällt.
- Es sollte eine Methode angeboten werden, die es erlaubt, Benutzer von Styleguides über Änderungen aufmerksam zu machen. Hilfreich wäre es, einen Styleguide online zugänglich zu machen, aber man benötigt dennoch einen Mechanismus zur Benachrichtigung.

Vorschläge von Johnson (2010)

LIT-JOHN

Johnson definiert fünf sogenannte „soziale Regeln“, die man bei der Erstellung, Änderung und Einführung von Styleguides befolgen sollte:

- Schliesse keine Personen von Entscheidungen aus, die sie betreffen.
- Führe keine neuen Regeln oder Konventionen ein, ohne einen guten Grund.
- Erwarte kein „Dankeschön“ für deine Arbeit.
- Erahne Einwände im Voraus.
- Höre genau zu, wenn andere Personen Bedenken äussern.

4.2 Recherchen in der Praxis

Ce chapitre décrit les résultats et analyses de la recherche empirique réalisée sur le terrain. L'analyse des données récoltées lors des interviews ont permis la distinction des utilisateurs d'une part et des processus d'autre part.

Dans la première partie l'étude des variables comportementales est exposée, décrivant le cheminement permettant de parvenir aux Personas définitives. Ensuite les Personas sont brièvement décrites. Les artefacts complets sont disponibles en annexe *Anhang B: Personas*.

La deuxième partie synthétise le contenu du document Business Modeling (*Anhang C: Business Use-Cases*). Les processus liés aux styleguides sont détaillés et expliqués. L'utilisation des styleguides au sein de projets fait l'objet d'un sous chapitre à part.

La dernière partie constitue un récapitulatif final des principales découvertes et points essentiels des recherches empiriques.

L'analyse des tâches a été réalisé dans le cadre des scénarios de contexte Ils décrivent l'utilisation future de l'application par les Personas. Les scénarios font partie intégrante des Personas, sous la forme d'un paragraphe „a day in life of“.

4.2.1 Analyse der Benutzer

Suivant l'hypothèse de base [HYP-1], il existe trois Personas pour l'application *Online-Styleguide: Owner, Writer* et *User*. Les interviews ont été planifiés et exécutés en fonction de cette constellation. Le chapitre 3.1.2 *Recherche über Benutzer und Aufgaben – Interviews* décrit en détail les types d'interviews ainsi que la répartition de ceux-ci par Persona.

Lors des interviews les aspects du contexte et des tâches du futur utilisateur ont été explorés et documentés, selon la catégorisation de Richter et al. (2007 S. 21). Les priorités ont été définies différemment pour certains aspects selon la Persona examinée. Le tableau illustre ces différences:

Aspects	Priorité pour le Owner	Priorité pour le Writer	Priorité pour le User
Distribution des rôles et communication	Haute	Haute	Moyenne
Stratégie d'action et approche	Moyenne	Moyenne	Haute
Artefacts	Basse	Moyenne	Haute
Influence culturelle et sociale	Basse	Basse	Basse
Environnement physique	Basse	Basse	Basse

Tabelle 14: Aspects des interviews et priorités par rapport aux Personas

La communication et les processus sont prioritaires avec les *Owners* et *Writers*, tandis que les documents et la manière de travailler étaient en centre de l'entretien pour les *Users*. L'environnement physique joue un rôle moindre étant donné le cadre de travail de bureau dans lequel se situait tous les intervenants.

4.2.1.1 Analyse der Personas

Le graphique ci-dessous répartit les interviews sur les variables comportementales. Le nom des interviews commencent par la Persona: "O" pour *Owner*, "W" pour *Writer*, "U" pour *User* suivit du numéro d'interviews. Dans les utilisateurs, les analystes sont en orange et les développeurs en bleu.

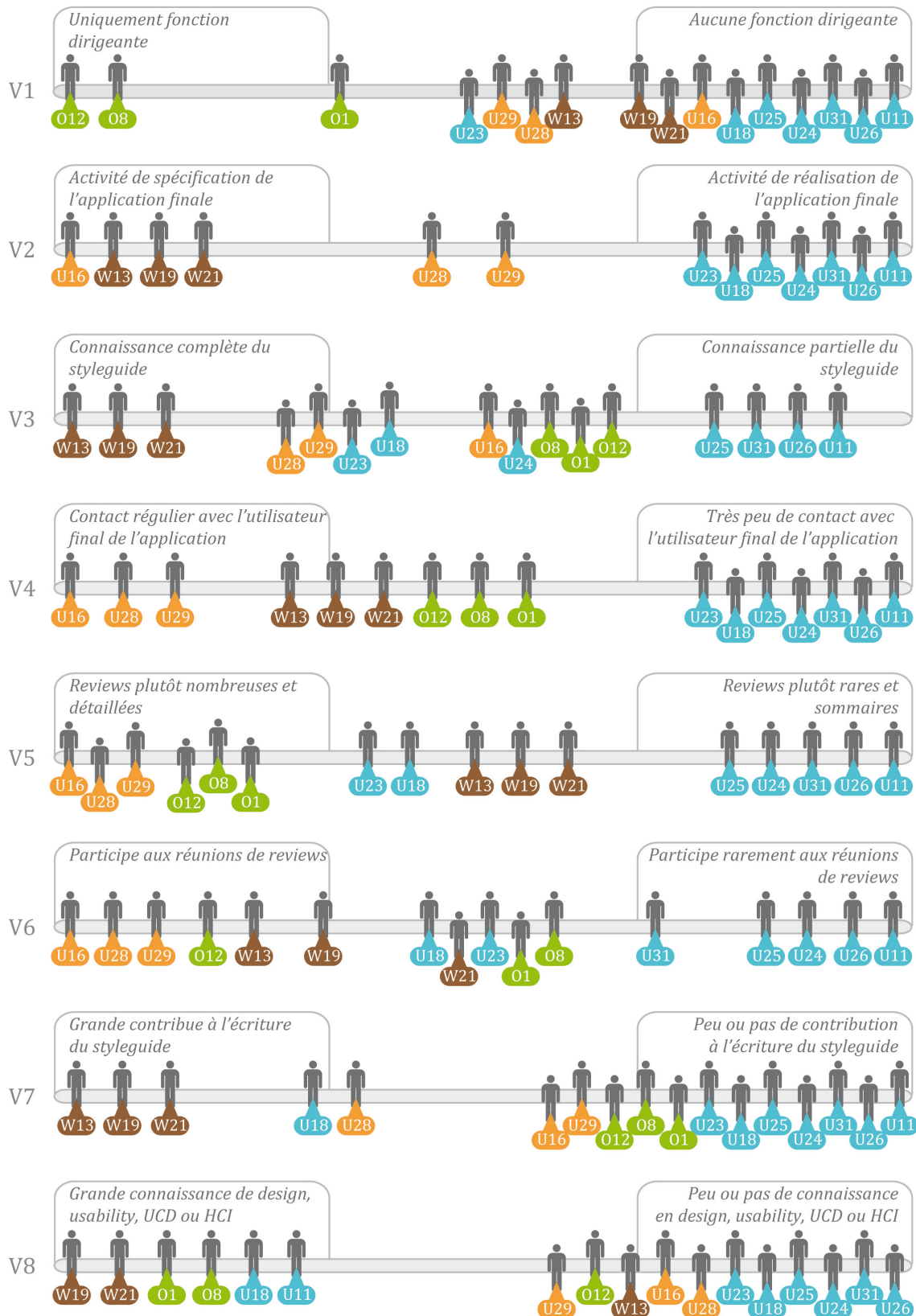


Abb. 14: Analyse des Personas (Behavior Variable Map)

Huit variables ont été retenues pour évaluer l'hypothèse sur les Personas. Ces variables sont utilisées et référencées dans l'analyse entre [crochets].

Variable	Description
V1	Part de management dans le travail de la personne: uniquement dirigeante, partiellement dirigeante ou aucune fonction dirigeante.
V2	Rôle joué par la personne dans le projet: soit dans la réalisation ou dans la spécification de l'application.
V3	Degré de connaissance du styleguide: connaissance globale ou fragmentaire (parties la concernant) du styleguide.
V4	Contact avec les utilisateurs finaux (lorsque c'est possible) ou avec leurs représentants (département de communication ou marketing).
V5	Etendue et fréquence des révisions réalisés: soit très régulières et détaillées, éventuellement volumineuses, ou révisions ou feedbacks plutôt rares et concis.
V6	Liée avec le variable V5, participation aux réunions de révision: D'un côté les personnes assistent aux réunions de révision, exposent leurs points de vue, de l'autre les personnes ne sont pas présentes ou pas invitées.
V7	Contribution des personnes pour la spécification et la rédaction du styleguide.
V8	Connaissances et compétences des personnes dans les domaines de l'utilisabilité, du design ou du design centré sur l'utilisateur.

Tabelle 15: Description des variables comportementales (Behavior Variables)

Trois Personas au lieu de *User*

Il apparaît lors de l'analyse que la persona *User* ne peut pas représenter tous les utilisateurs. Les développeurs ont montré des motivations, comportements et objectifs différents des analystes. Les variables [V2], [V4], [V5] et [V6] illustrent clairement ces divergences. Pour ces raisons le *User* a été scindé en deux Personas, *Dave Loper* et *Anna Lyst*.

Lors des interviews, il est apparu qu'une Persona supplémentaire pouvait être créée. L'utilisateur final de l'application peut dans certain cas être amené à consulter le styleguide (pour former les futurs utilisateurs par exemple). La Persona *Urs Superuser* a donc été créée, avec la mention provisoire, étant donné que nous ne l'avons pas vérifiée par des interviews.

Owner est conforme

Le *Owner* a pu être vérifié et est conforme aux attentes formulées dans l'hypothèse de base.

Pas assez d'informations sur le *Writer*

L'analyse du *Writer* n'a pu aboutir, il est donc provisoire. Trop de zones d'ombres persistent sur le rôle que les *Writers* jouent dans les révisions (variables [V5] et [V6]) ainsi qu'au niveau de leurs contacts avec les utilisateurs finaux [V4]. L'impact sur l'application du fait que le *Writer* est souvent externe à l'entreprise doit être éclairci. Le rôle joué par certains développeurs [U18] ou certains analystes [U28], pouvant être amenés à écrire ou compléter des parties de styleguides n'est pas clair non plus.

Priorisations

Anna Lyst est la Persona primaire des utilisateurs, vu l'importance qu'elle joue dans la création et dans la coordination du styleguide. *Dave Loper* est une Persona secondaire. Il est satisfait des fonctions de base prévues pour *Anna Lyst*. Les fonctions supplémentaires dont il a besoin, par ex. pour le code source, ne la gênent. *Urs Superuser* n'étant pas prioritaire et n'ayant pas de besoins supplémentaires, il est donc une persona complémentaire.

Ottmar Owner et *Hans Writer* sont tout deux Primaires. Ils ont deux rôles et besoins clairement différents de *Anna Lyst*.

Problèmes et cas spéciaux

La validité de l'interview [O1] peut être remise en question. Le déroulement de l'interview et le manque de clarté dans les déclarations de la personne peut avoir un impact sur les résultats (rôle passé ou présent, est-ce qu'elle parle pour les autres).

La pertinence de la variable [V4] doit être relativisé. Elle dépend largement de l'application développée (interne ou externe). Par exemple pour un site internet, il n'est pas possible de voir l'utilisateur final, il est donc représenté par le *Owner* ou le département de marketing. Au contraire d'une application développée pour un département interne. Ceci rend une comparaison entre les différents interviews difficile.

Résultats

L'analyse n'a en général pas été aisée. Cependant nous pouvons dire que l'hypothèse formulée [HYP-1] est en partie valide et vérifiée. Les Personas provisoires sont illustrées avec un dessin et non une photographie pour renforcer leur nature provisoire, comme le recommande Cooper et al (2007).

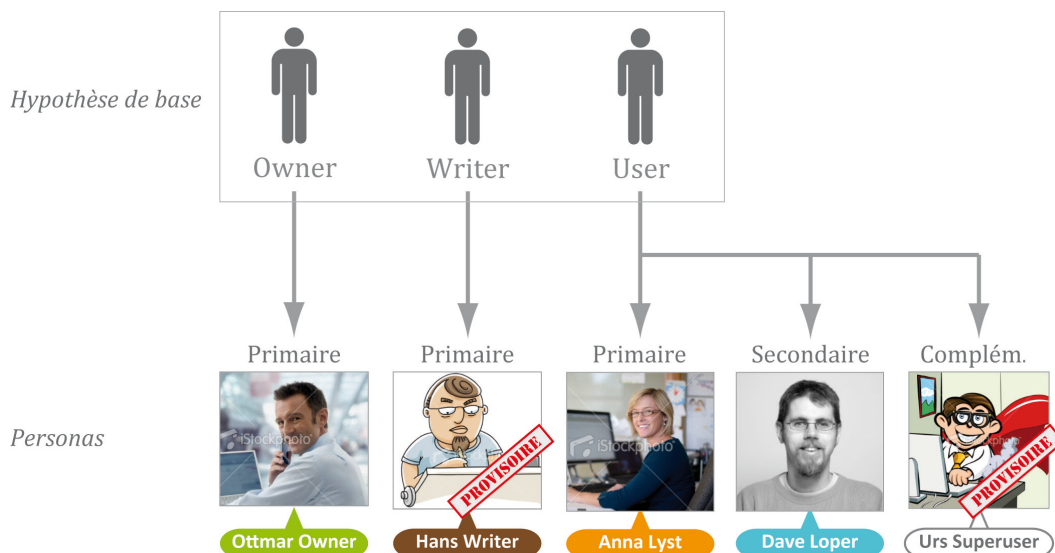


Abb. 15: Mapping des hypothèses et des résultats des Personas

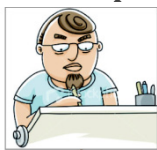
4.2.1.2 Personas in Kürze

Persona primaire *Ottmar Owner*



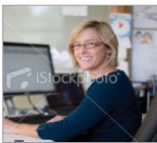
Ottmar est cadre de son entreprise, il gère des projets, s'occupe entre autres de travaux administratifs et du budget. Il coordonne par ex. les ressources externes à l'entreprise, les appels d'offres, etc. Il n'a que des connaissances partielles du styleguide. Il organise et exécute parfois lui-même les révisions du styleguide. Il a de bonnes connaissances du domaine HCI et voit parfaitement les avantages des styleguides. Il s'assure de garantir leur qualité ainsi que leur utilisation afin de livrer les meilleurs produits et si possible d'économiser de l'argent.

Persona primaire provisoire *Hans Writer*



Hans est un spécialiste dans son domaine. Il rédige les styleguides qui seront utilisés dans les projets de développement d'application. Il est souvent externe à l'entreprise et travaille de temps en temps chez les clients. Il utilise des outils pointus de design pour réaliser la partie visuelle du styleguide ou des prototypes. Il a suivant son domaine des connaissances de base d'HTML pour la réalisation de prototypes. Il connaît très bien les styleguides sur lesquels il travaille. Il n'a pas de fonction dirigeante.

Persona primaire *Anna Lyst*



Anna est la Persona primaire des utilisateurs. C'est elle qui rédige les spécifications de l'application et peut également être amenée à faire du prototypage. Elle donne des mandats aux designers. Elle assure un rôle de coordination entre les différents membres du projet (*H. Writer, O. Owner, Dave Loper, Urs Superuser*). En général, elle dispose de bonnes à très bonnes connaissances du styleguide. Elle n'a pas fait d'études spéciales dans le domaine HCI. Elle connaît le métier de l'entreprise qu'elle décrit. Elle peut enfin avoir des tâches de gestion de projet, mais celles-ci ne constituent pas son activité principale.

Persona secondaire *Dave Loper*



Dave réalise et teste les applications. Il trouve les styleguides pratiques et connaît les avantages de leurs utilisations. Parfois, il préférerait ne pas en avoir pour être plus libre. Plus que l'application de gestion des styleguides, c'est le contenu du styleguide qui joue un rôle pour lui. Il préfère une structure claire, des exemples avec des illustrations ainsi que des extraits de code. Il doit pouvoir accéder facilement à ce qu'il cherche. Avec l'expérience, il peut prendre parfois la responsabilité du groupe de développement. Il a relativement peu de contact avec l'utilisateur final.

Persona complémentaire provisoire *Urs Superuser*



Urs est un utilisateur expérimenté, il connaît très bien le métier de l'entreprise. Il a participé à la spécification de la nouvelle application et conseille régulièrement le projet sur ses préférences. Il n'accède au styleguide qu'en lecture. Il regarde à quoi ressemblera l'application et utilise le styleguide pour former les futurs utilisateurs.

4.2.2 Analyse der Prozesse

Nous abordons ici les processus et le contexte d'utilisation des styleguides. La description des processus à l'aide des Business Use-Cases (BUC) permet de modéliser et de présenter une vue d'ensemble des acteurs métiers et des processus d'entreprise. Ce chapitre explique les BUC décrits dans l'annexe *Anhang C: Business Use-Cases*. La seconde partie de ce chapitre concerne le cycle de vie des styleguides dans les processus de développement d'application.

4.2.2.1 Business Use-Cases

Dans les types de processus, on distingue habituellement :

- Les processus opérationnels (de réalisation)
- Les processus de support (de soutien ou ressources)
- Les processus de pilotage (de management ou décisionnels)

Les processus opérationnels représentent l'activité primaire de l'entreprise, ce qu'elle produit. Les processus de support mettent à disposition les ressources ou le matériel à l'interne. Les processus de pilotage permettent quand à eux de diriger l'entreprise (le pilotage stratégique et opérationnel) (Brandenburg, et al., 2003). La figure ci-dessous montre une vue d'ensemble des BUC liés aux styleguides. Les autres processus d'entreprise ne font pas partie du scope et ne sont pas décrits ici.

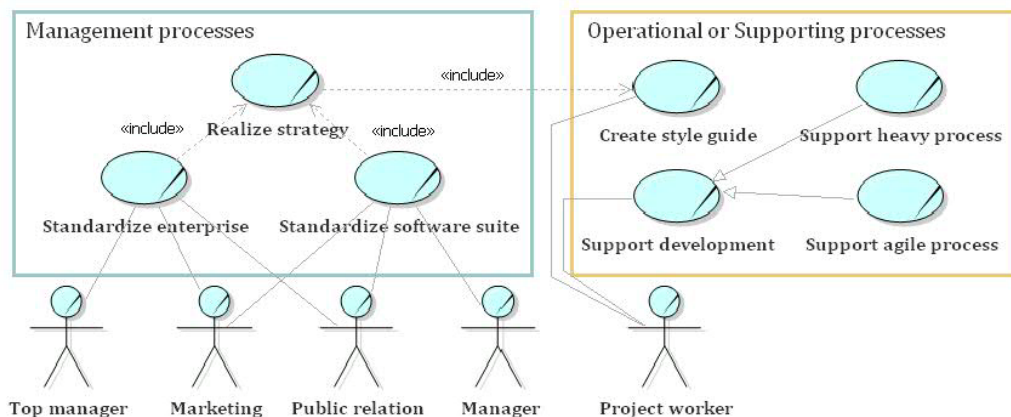


Abb. 16: Vue d'ensemble des Business Use-Cases

Les processus de pilotage

Le *Top manager* à travers les processus de pilotage (*Management processes*) définit les objectifs et buts stratégiques pour l'entreprise et prend les mesures pour y arriver. Ces processus ont pour but la standardisation des logiciels de l'entreprise à l'aide de styleguide(s). Les avantages de ceux-ci sont listés dans la *Tabelle 6: Vorteile von Styleguides* (Gale, 1996).

Les objectifs sont en premier lieu définis dans *Standardize enterprise*, s'ils s'appliquent à toutes les applications de l'entreprise. *Standardize software suite* réunit les objectifs du ou des styleguide(s) ne s'appliquant qu'à une partie des applications par ex. à une suite logicielle. Cette différenciation est due aux acteurs métiers qui peuvent décider de ces standardisations. Ces deux niveaux de processus ont été observés dans différentes grandes entreprises que nous avons étudiées. La mise en place de la stratégie *Realize strategy* correspond à la création du ou des styleguides par les processus opérationnels ou de supports. Une fois créés, les styleguides sont assignés aux projets de développement des applications. Pour finir les résultats sont vérifiés et les objectifs contrôlés.

Les processus de support et opérationnels

Support development et *Create style guide* peuvent être de type opérationnel ou de support suivant les activités de base l'entreprise. Si l'entreprise est prestataire de service informatique à des tiers, comme une agence de création de site web, ces processus sont opérationnels. Si les processus ne sont pas liés au métier de base, il s'agit de processus de support, par ex. la création du site internet d'une compagnie d'assurance par son service IT. Des styleguides peuvent être assigné aux projets (styleguide d'entreprise) mais également être créé pendant le BUC de support du développement (styleguide de projet). C'est le chef de projet ou le manager qui décide de sa création. Ce schéma est fréquent dans les petites et moyennes entreprises, mais aussi dans les grandes entreprises. Ces cas de figure ont été également observés dans les interviews.

Le support du développement

Dans *Support development*, les styleguides appuient et supportent le développement d'applications informatiques. Suivant la méthodologie, les processus peuvent varier considérablement. Les interviews ont confirmé les points relevés dans la littérature (4.1 *Recherchen in der Literatur*). *Support development* est donc spécialisé par deux BUC, l'un pour les processus lourds et l'autre pour ceux dit léger ou „Agile”.

La création du styleguide

Le déroulement de *Create style guide* dépend du type de styleguide, des acteurs impliqués, du commanditaire (processus de pilotage ou projet) et surtout de la méthodologie du projet. C'est pourquoi les sous processus de la création sont optionnels et sont reliés par des „extend” à *Create style guide*. Ils ne sont exécutés que si les conditions sont remplies.

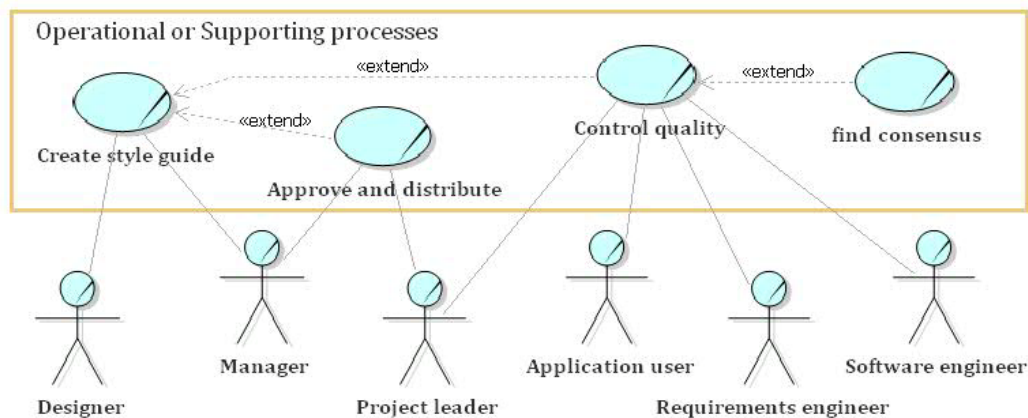


Abb. 17: Create styleguide Business Use-Case

Les contrôles de qualité (*Control quality*) sont effectués si le manager le décide, au rythme qu'il juge nécessaire. Tous les acteurs peuvent être impliqués dans les révisions. Ce processus peut inclure un cas spécial de révision *Find consensus*, pour trouver des solutions ou voter sur une variante de design. L'approbation et la distribution du styleguide (*Approve and distribute*) sont dépendantes de la méthodologie choisie et du Workflow défini. L'approbation et la distribution du styleguide (*Approve and distribute*) sont dépendantes de la méthodologie choisie et du Workflow défini.

4.2.2.2 Styleguides im Software-Entwicklungsprozess

Dans ce chapitre il est question de l'utilisation des styleguides dans les méthodologies de développement. Les informations proviennent des interviews et de la littérature (2.4.5 *Styleguides im Software-Entwicklungsprozess*). Les illustrations n'ont pas la prétention d'avoir une quelconque validité scientifique. Notons que le SG – de type projet – est dans les deux cas créé pendant le projet.

L'écriture du styleguide est symbolisée en rouge orange dans le graphique ci-dessous et représente la répartition de l'activité lors de la création du SG en fonction du temps et des méthodologies.

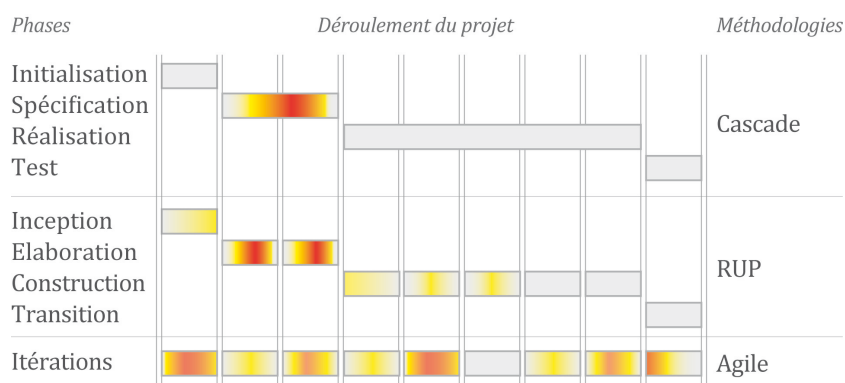


Abb. 18: Activité d'écriture des styleguides dans les méthodologies

La création dans Agile est répartie tout au long des itérations dès lors que le SG le plus sommaire a été créé. Comme vu dans la théorie (2.4.5 *Styleguides im Software-Entwicklungsprozess*), l'effet suiveur ou moteur du SG est visible. Parfois le SG n'est pas mis à jour tout de suite, il suit le développement. En théorie dans un modèle en cascade le SG est défini dans la phase de spécification et ne devrait plus être modifié par la suite. Ce point n'a pas pu être vérifié, du fait qu'aucun partenaire n'utilisait de modèle cascade pur. Les SG sont également créés de manière itérative comme dans l'exemple RUP ci-dessus. Ces points ont également été observés dans les interviews.

La lecture du styleguide est symbolisée en bleu dans le graphique ci-dessous et représente la répartition de l'activité de lecture du SG en fonction des méthodologies et du temps.

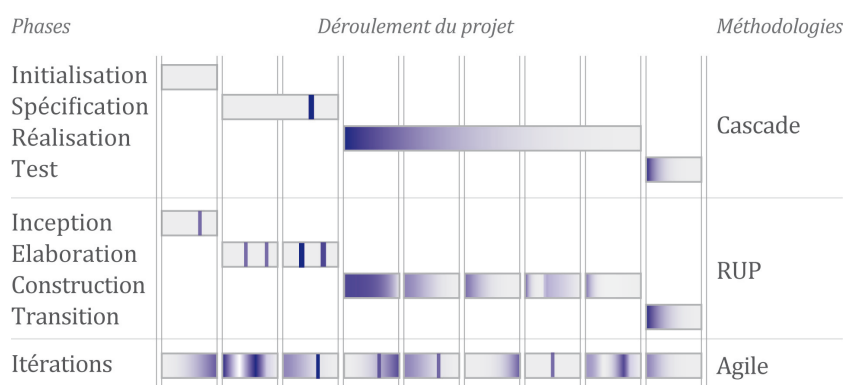


Abb. 19: Activité de lecture des styleguides dans les méthodologies

L'utilisation du styleguide est plus homogène dans la méthodologie Agile. Etant donné que le contenu change plus souvent, celui-ci est ainsi consulté plus fréquemment. Dans les méthodologies „lourdes“, le styleguide est lu en entier au début projet et sert ensuite de référence [U31, U25, U26], c'est pourquoi la barre se dégrade au fil du projet. Lors de la phase de test le styleguide est utilisé pour vérifier si l'application est conforme à la spécification.

4.2.3 Fazits der empirischen Analyse

Dans ce chapitre nous résumons les découvertes, points intéressants des interviews et de la recherche de terrain. Toutes les interviews et variables comportementales du graphique *Abb. 14: Analyses des Personas (Behavior Variable Map)* sont référencées entre [crochets]. Le styleguide est abrégé SG.

4.2.3.1 Sur les personnes, les rôles et tâches

Nous avons observé que la majorité des personnes interrogées jouent plusieurs rôles simultanément au sein de leur organisation ou projet. Ceci indépendamment de la taille de l'entreprise. Les noms des postes et des rôles changent beaucoup d'une entreprise à l'autre, tout comme les responsabilités et tâches assignées. Les fonctions dirigeantes sont aussi très partagées. Sur 16 interviews, 5 personnes interrogées ont en partie des tâches de gestion, toutes Personas confondues ([O1, U13, U23, U28, U29] variable [V1]).

Les personnes présentes depuis le début du projet ont de meilleures connaissances du SG [U18, U23, U24, U28, U29] au contraire des personnes arrivant en cours de projet [U16, U25, U26, U31]. La majorité des personnes interrogées n'ont pas de connaissances spéciales dans les domaines du design ou du design centré sur l'utilisateur (UCD) (variable [V8]).

Le „Top manager” est cité dans plusieurs interviews et apparaît dans les BUC. Il n'a pas été pris en compte dans l'analyse des utilisateurs (Personas), car il est en dehors du scope et difficile d'accès. C'est une non-Persona, c'est-à-dire une personne n'utilisant pas le système directement, mais qui doit être pris en compte dans le processus global de design de l'application (Cooper, et al., 2007).

4.2.3.2 Sur les entreprises et le contexte

Des structures organisationnelles distinctes sont parfois constituées spécialement comme dans les interviews [U8, U24, U21]. Leurs tâches sont uniquement de générer et de mettre à disposition les styleguides. Ces structures ont surtout été observées dans les grandes entreprises.

Beaucoup de différences ont été observées lors des interviews, en termes de plateformes (Web, Rich client, etc.), de technologie (J2EE, HTML, PHP, etc.) et d'applications (domaine bancaire, assurance, transport). Dans chaque cas le SG et les besoins étaient différents. En tant que développeur, [U31] est d'avis qu'il est très difficile de faire des SG détaillés pour une technologie, en raison du changement constant des Frameworks et des plateformes.

4.2.3.3 Sur les processus et les méthodologies

Chaque entreprise observée utilise et adapte les méthodologies à ses processus. Il n'y a pas de claire majorité dans les méthodologies utilisées, comme le montre la table ci-dessous:

Méthodologie	O1	O8	O12	U11	U13	U16	U23	U29	U31
RUP	x	x	x					x	Mix
Agile (incl. Scrum)	Scrum			Scrum		Agile			Scrum
Cascade (incl. Hermes)			x	Mix	x		Hermes		

Tabelle 16: Types de méthodologie utilisées (résumé de la recherche empirique)

Il n'est pas rare que les méthodologies soient mixées entre elles ou qu'un projet utilise une autre méthodologie que celle définie dans l'entreprise [U16]. Les entreprises en changent également [O1] ou en utilisent plusieurs à la fois [U16, U11 et U31].

Les processus de développement Agile utilisent en général le SG de manière plus régulière, autant pour la lecture que pour l'écriture (comme nous l'avons précisé dans le chapitre 4.2.2.2 *Styleguides im Software-Entwicklungsprozess*). Le SG est rédigé au fur et à mesure pendant les itérations ou „sprint“. Ceci est confirmé par les observations issues des interviews. Dans cette optique, [U11] ne s'intéresse qu'au changement du styleguide sur les parties qu'il l'intéresse.

Création du styleguide

Les SG sont en général créés par plusieurs personnes en parallèle [W13, W19]. Le SG est rédigé par le *Writer*, bien qu'il puisse être aidé par le *User* ([U18, U28] variable [V7]). De plus, les SG sont très souvent créés par des entreprises externes [U18, U16, O1, W21]. La réutilisation de partie de SG dans d'autres styleguides peut se produire, comme dans le cas de [U18].

Les révisions

Dans toutes les entreprises visitées les styleguides sont révisés. Le volume et l'envergure dépendent encore une fois des entreprises et des méthodologies choisies. Les développeurs ne sont pas toujours intégrés dans les processus de révisions ainsi que dans les réunions organisées à cet effet ([U11, U24, U25, U26, U31] variables [V5, V6]). Il n'est cependant pas toujours possible d'intégrer les développeurs comme l'explique [O8]. Si le développement est externalisé et qu'il y a un appel d'offre, le SG est déjà terminé quand l'entreprise est choisie. Un cas spécial des révisions est le vote d'une alternative de design. Il arrive que le designer ait plusieurs solutions à soumettre à l'équipe de projet. Les variantes sont évaluées par l'équipe, qui se met d'accord et en choisit une [U16].

La collaboration

Les moyens de communication sont principalement les courriels et la communication directe. La collaboration peut être plus compliquée dans le cas où certains membres du projet sont externes à l'entreprise et qu'ils travaillent en délégation ([U31], les *Writers* qui travaillent avec [U16] et [U18]). Le *Writer* du projet de [U16] travaille un jour par semaine sur place avec le groupe de projet. [U16] trouve cela bien et permet de régler beaucoup de problèmes directement et rapidement. Il dit perdre souvent du temps pour rassembler les feedbacks des révisions et ne sait pas parfois où en est le *Writer* dans son travail. [U18] ajoute que la communication avec le *Writer* n'est pas optimale.

La gestion des versions

Toutes les entreprises visitées gèrent les versions de leurs styleguides, autant les versions papiers qu'électroniques. Dans le projet de [U16], le SG est online et chaque itération génère une nouvelle page HTML portant le numéro de l'itération. Toutes les versions sont affichées en tant que lien à l'écran. Il est possible de changer de version d'un simple clic. [U16] trouve également pratique de voir l'historique des changements et la comparaison entre les versions.

4.2.3.4 Sur les styleguides

Tous les styleguides observés lors des interviews sont différents au niveau du fond et de la forme. Le contenu et le niveau d'abstraction varient beaucoup suivant les entreprises. Les raisons ont déjà été évoquées plus haut au chapitre 4.2.3.2 *Sur les entreprises et le contexte*. Les utilisateurs travaillent avec plusieurs SG en même temps. En général plusieurs SG sont assignés à un projet de développement.

Le contenu des styleguides, mise à part le texte, peut être constitué d'images, d'icônes, de graphiques et d'extraits de codes. Les exemples de code et autres Snippets sont très appréciés des développeurs [U25, U26, U31]. Les exemples et explications illustrées sont des préférences maintes fois citées.

Presque toutes les personnes interrogées disent imprimer le styleguide. [U31, U25] ne l'impriment que la première fois pour le lire en entier, car ils ont le sentiment que la lecture est plus agréable. Par la suite, pour prendre connaissance de petits articles le format PDF est consulté. Pour [U31], ce sont principalement pour des raisons écologiques. [U26] ne lit le styleguide qu'à l'écran. [U29] imprime le styleguide sous forme de booklet et le prend pour les discussions dans les réunions.

L'utilisation et la recherche dans les styleguides

L'utilisation et la création des SG sont directement dépendantes de la méthodologie du projet. Dans son projet agile [U16] travaille avec trois versions du SG en parallèle (l'itération précédente pour les testeurs, l'itération actuelle pour les développeurs, l'itération future pour lui). Il existe différentes stratégies d'utilisation du SG comme expliqué dans le chapitre 4.2.3.2 *Sur les entreprises et le contexte*. Dans la majorité des cas le SG est utilisé comme référence. Suivant la phase de projet l'utilisation du SG peut être plus ou moins importante [U31].

Les techniques de recherche sont très différentes selon les personnes interrogées, leur niveau de connaissance, le type et le format du SG. [U25] dit n'avoir besoin d'aucune recherche particulière. [U23] et [W21] connaissent tellement bien le styleguide qu'ils savent à peu près où se trouve ce qu'ils cherchent. Ils utilisent dans ce cas la table des matières pour aller au chapitre voulu. La table ci-dessous résume les méthodes de recherches principales citées dans les interviews:

Type de recherche	O1	U11	U16	U18	W21	U23	U24	U25	U28	U29	U31
Table des matières			x		x	x		x	x		
Textuelle, mots clefs	x	x					x			x	
Scroller											x

Tabelle 17: Types de recherche utilisées (résumé de la recherche empirique)

Les formats et les outils pour éditer les styleguides

L'édition ainsi que la transmission du SG sont également des préoccupations récurrentes. Le format de fichier PDF est privilégié dans beaucoup d'entreprises pour distribuer le SG [O1, U18, U23, U28, 31]. Pour le projet de [U16], l'entreprise externe du *Writer* a mis en place un SG online que [U16] trouve très pratique car il peut y accéder en tout temps et de partout. C'est pour lui un "luxe". Dans la même entreprise, le choix d'un styleguide online a été abandonné — dans le projet de [O12] — en raison de coûts trop importants, du manque de plateforme adéquate, du manque de connaissances dans le domaine et du format de livraison préalablement convenu avec le créateur du SG.

La table ci-dessous récapitule les différents formats rencontrés lors des interviews:

Format	O1	U11	U16	U18	W21	U23	U24	U28	U29	U31
PDF	x			x		x		x		x
WORD	x	x	x	x		x				
HTML			x	x					x	
Adobe					x					
POWER POINT							x			

Tabelle 18: Types de format des SG (résumé de la recherche empirique)

WORD reste un format très attractif, parce que facilement modifiable par n'importe qui [O1, U11]. Chez [U23] WORD reste utilisé chez le créateur du styleguide. D'autres outils sont utilisés pour créer le styleguide, particulièrement pour tout ce qui concerne le domaine du visuel, comprenant icônes, graphiques, interfaces (par exemple ILLUSTRATOR de la suite Adobe [W21]).

4.2.3.5 Table récapitulative

La table ci-dessous résume les points les plus importants des résultats la recherche empirique:

Code	Thème	Résumé	Section
PRAX-1	Rôles et tâches	Beaucoup de rôles et de tâches différentes dans les entreprises	4.2.3.1
PRAX-2	Connaissances du SG	Dépend du temps passé dans le projet, du moment d'arrivé	4.2.3.1
PRAX-3	Contexte (entreprise)	Bcp. de différents Frameworks, plateformes, technologies	4.2.3.2
PRAX-4	Méthodologie	Beaucoup de différences entre les entreprises, pas de standard	4.2.3.3
PRAX-5	Agile	Dans Agile, les SG sont utilisés de manière plus homogène	4.2.3.3
PRAX-6	Communications	Principalement les Emails et les réunions directes	4.2.3.3
PRAX-7	Outils de collaboration	Pour faire des révisions et voter pour des variantes de design	4.2.3.3
PRAX-8	Gestion des versions	Tous les SG ont une gestion des versions	4.2.3.3
PRAX-9	Participants aux révisions	Les développeurs ne sont pas toujours impliqués dans les révisions du SG, ça dépend de l'entreprise	4.2.3.3
PRAX-10	Réutilisation du SG	Des parties de SG sont réutilisées dans d'autres SG	4.2.3.3
PRAX-11	Collaborateurs externes	Difficulté parfois de communiquer et de se synchroniser	4.2.3.3
PRAX-12	Utilisation des SG	Les SG peuvent être utilisés dans et par plusieurs projets	4.2.3.3
PRAX-13	Avertissement	L'utilisateur veut être informé des changements du SG	4.2.3.3
PRAX-14	Création de SG	Le SG est créé par plusieurs personnes en parallèle, souvent par des collaborateurs externes	4.2.3.3
PRAX-15	Structure des SG	Beaucoup de différence entre les SG des entreprises	4.2.3.4
PRAX-16	Impression des SG	Les styleguides sont imprimés	4.2.3.4
PRAX-17	Moyens de recherche	Les utilisateurs utilisent plusieurs types de recherche	4.2.3.4
PRAX-18	Format des SG	Il existe plusieurs formats de SG	4.2.3.4
PRAX-19	Contenu des SG	Les SG contiennent des images, du texte et du code. Ce dernier est très apprécié des développeurs.	4.2.3.4
PRAX-20	Création de SG	Plusieurs outils sont nécessaires à la création du SG, en particulier pour ce qui visuel	4.2.3.4

Tabelle 19: Récapitulatif des découvertes de la recherche empirique

5 Die Zusammenfassung der Erkenntnisse – Aspekte, Features und Konkurrenzanalyse

Aus den Recherchen in der Literatur und der Praxis (4 *Die Suche nach Erkenntnissen – Recherchen*), haben wir viele Erkenntnisse gewonnen. Um anhand dieser ein Konzept für unser System abzuleiten, haben wir zunächst eine Reihe von *Aspekten* formuliert. Daraus haben wir anschliessend konkrete Anforderungen an das System abgeleitet und in Form von *Features* und *Requirements* festgehalten. Diese wiederum bilden die Basis für eine *Konkurrenzanalyse*.

Dieses Kapitel gliedert sich daher in drei Abschnitte.

Der erste Abschnitt beschreibt die *Aspekte* des Systems. *Aspekte*, oder wahlweise auch *Top-Level Features* genannt, sind die grundlegenden Eigenschaften, die das System erfüllen muss, damit es einen Nutzen bringt. In diesem Abschnitt wird für jeden Aspekt beschrieben, welche Ziele er verfolgt, welche Erkenntnisse aus der Literatur oder der Praxis dazu führen und welche Lösungsvorschläge dabei in Betracht gezogen werden. Bei den Begründungen für einen Aspekt wird auf die Erkenntnisse aus der Literatur (4.1 *Recherchen in der Literatur*) und die Fazits der Praxisrecherchen (Tabelle 19: *Récapitulatif des découvertes de la recherche empirique*) Bezug genommen. Dies geschieht anhand der in den jeweiligen Kapiteln, bzw. Tabellen erstellten Referenzen. Diese finden sich in der Beschreibung anhand [eckiger Klammern].

Der zweite Abschnitt behandelt die *Features*, die den jeweiligen Aspekten untergeordnet sind. Diese sind tabellarisch zusammengefasst (Tabelle 20: *Features des Systems*). Die *Requirements*, bzw. die weiteren detaillierten Anforderungen an das System, sind nicht Teil dieses Kapitels. Sie sind im *Anhang D: Features und Requirements* abgelegt.

Der dritte Abschnitt widmet sich der Konkurrenzanalyse. Dabei werden, basierend auf den *Features*, existierende Applikationen einem „Benchmarking“ unterzogen. Der Sinn dabei ist, Aufschluss darüber zu geben, in welchem Mass die untersuchten Applikationen den Anforderungen genügen.

Scope der Requirements

Es ist anzumerken, dass die im Artefakt *Features und Requirements* beschriebenen *Requirements* sich auf den Scope des Prototyps und entsprechend auf die Use-Cases *Visualize Content* und *Search Content* beziehen (Abb. 34: *Use-Case Model (User, Coordinator, Realizer)*). Sie sind ebenfalls auf die Personas *Anna Lyst* und *Dave Loper* (4.2.1.2 *Personas in Kürze*) beschränkt.

Stand der Features und Requirements

Der Stand der in diesem Kapitel beschriebenen *Features* und die im Artefakt *Features und Requirements* enthaltenen Daten entsprechen dem Stand vor der Durchführung der Usability-Walkthroughs. Die Ergebnisse aus den Usability-Walkthroughs sind entsprechend nicht in die Beschreibung eingeflossen.

5.1 Aspekte

5.1.1 Unterstützung der Kollaboration

Ziele Die Förderung der Kollaboration durch Einbezug betroffener Personen steigert die Akzeptanz von Styleguides und sorgt dafür, dass darin die besten Ideen und Vorschläge integriert werden. Dadurch lässt sich die Benutzbarkeit und die Umsetzbarkeit der Styleguides sicherstellen. Die Qualität von Styleguides und folglich von Produkten steigt. Durch die Vergabe von Aufträgen zur Erstellung von Styleguides und Durchführung von Reviews werden betroffene Personen in den Prozess einbezogen. Gleichzeitiges Editieren mehrerer Personen an einem Styleguide fördert ebenfalls die Kollaboration und erhöht die Effizienz, da nicht abwechselnd daran gearbeitet werden muss.

Grund Dieser Aspekt entstammt aus den Erkenntnissen der Problematik der Prozesse und der Kollaboration aus der Literatur (4.1.5 *Themenbereich: Prozesse und Kollaboration*). Dadurch sollen die Probleme der Kollaboration und Akzeptanz [LIT-PK-5], der Aktualisierungen und Änderungen [LIT-PK-6] und der Umsetzbarkeit [LIT-PK-7] gelöst werden. Aufgrund der Vorschläge in [LIT-QUES] und [LIT-GALE] soll ein Styleguide einem breiten Publikum zugänglich gemacht werden. Dadurch werden die Benutzer von Styleguides sensibilisiert und die Konsensbildung, durch Einflussnahme mittels Kommentaren, unterstützt.

Dieser Aspekt bildet auch die Basis für eine Lösung der Probleme aus den Themenbereichen des Inhaltes (4.1.3 *Themenbereich: Inhalt*) und der Struktur und Form (4.1.4 *Themenbereich: Struktur und Form*). Die Probleme des Fokus, Detaillierungsgrades und Kontextes [LIT-IN-1], der Inkonsistenz [LIT-IN-3] und der Sinnhaftigkeit und Zweckmässigkeit [LIT-IN-7] lassen sich durch Einbezug von Benutzern eines Styleguides, insbesondere der Entwickler, einfacher lösen. Die Unterstützung der Kollaboration ist auch für die Problematik der Struktur und Organisation [LIT-SF-4] entscheidend, da ohne Einbezug der Zielgruppe des Styleguides nicht garantiert ist, dass eine benutzbare Struktur (Kapitel und Inhalte) erstellt werden kann.

Auch in der Praxis wird teilweise kollaborativ gearbeitet. Allerdings werden Entwickler nicht immer in die Prozesse einbezogen [PRAX-9]. Der Informationsaustausch findet meist über E-Mails oder mündlich statt [PRAX-6]. Sehr oft haben Personen unterschiedliche Rollen und Aufträge. Diese bearbeiten sie entweder selbst oder vergeben sie an andere Personen [PRAX-1]. Bisweilen wird bei der Durchführung von Reviews über Varianten bestimmter Inhalte abgestimmt [PRAX-7]. Die Praxis zeigt auch, dass die Erstellung von Styleguides teilweise von mehreren Personen gleichzeitig durchgeführt wird [PRAX-14].

5.1.2 Wiederverwendbarkeit von Inhalten

Ziele Schnellere Erstellung von Styleguides durch Wiederverwendung existierender Inhalte.

Grund In der Literatur geht nur Gale (1996) in seinen Vorschlägen zu den Prozessen [LIT-GALE] darauf ein, dass ein Styleguide ebenfalls von „anderen Kommunikationsmitteln“ begleitet werden sollte. Darunter versteht Gale auch den Einsatz von direkt wiederverwendbarem Quellcode. Dieser könnte in verschiedene Styleguides integriert werden.

In der Praxis werden Inhalte von Styleguides für andere Styleguides wiederverwendet [PRAX-10]. Dieselben Informationen, beispielsweise über Farbgebung, werden für Styleguides verschiedener Projekte verwendet.

5.1.3 Inhalts- und Change-Management

Ziele	Flexibilität bei der Erstellung der Struktur und Organisation von Styleguides und Inhalten und dadurch deren optimale Anpassung an die Bedürfnisse der Benutzer. Durch die Unterstützung verschiedener Arten von Inhalten können Informationen besser vermittelt werden und, im Falle von Inhalten wie Quellcode, direkt für die Entwicklung verwendet werden. Dadurch ergibt sich ein Zeitersparnis bei der Suche nach Informationen und die Verwendung derer. Zusätzlich stellt eine Versionierung sicher, dass alte Styleguides und Inhalte, beispielsweise zu Vergleichszwecken, zur Verfügung stehen.
Grund	Styleguides sind typischerweise in verschiedene Themenbereiche unterteilt, welche die Struktur (z.B. Kapitel, Unterkapitel) bilden. Nach der Literatur werden Styleguides stets in einem bestimmten Kontext (<i>Tabelle 7: Klassifizierungen von Styleguides</i>) erstellt und Benutzer haben unterschiedliche Ansprüche an ihn [LIT-SF-4]. Daher ist diese Struktur abhängig vom Kontext und der Benutzer. Tatsächlich zeigt die Praxis, dass, aufgrund der verschiedenen Kontexte [PRAX-3], diese Strukturen sehr unterschiedlich sind [PRAX-15]. Daher muss das System die Erstellung flexibler Strukturen (Kapitel, Unterkapitel, etc.) ermöglichen. Auch die Art der Inhalte selbst ist in der Praxis unterschiedlich [PRAX-19]. Meist beschränkt sich die Art des Inhalts auf Text und Illustrationen. Bei „Online-Versionen“ von Styleguides wird teilweise auf Quellcode oder direkt auf, für die Entwicklung direkt verwendbare, Widgets referenziert. In der Literatur zeigt die Studie von Tetzlaff et al. (1991), dass die Verwendung von Beispielen und Illustrationen in Styleguides den stärksten Einfluss (in positivem Sinne) auf die Vermittlung von Informationen haben. Durch die Verwendung „interaktiver Beispiele“ (Flash, Applet, etc.) könnte dieser Einfluss noch verstärkt werden. Daher soll das System nicht nur Text und Bild, sondern darüber hinaus noch weitere Inhaltsarten unterstützen (Video, Audio, Quellcode und „interaktive Beispiele“). Styleguides existieren in der Praxis in verschiedenen Versionen [PRAX-8]. Durch eine Versionierung ergibt sich die Möglichkeit eines Vergleichs verschiedener Versionen. Sie stellt auch sicher, dass alte Versionen erhalten bleiben und dient darüber hinaus dem Aspekt der Wiederverwendbarkeit (<i>5.1.2 Wiederverwendbarkeit von Inhalten</i>). Die Literatur verweist auf die Problematiken des Index und der Querverweise [LIT-SF-1] und der Form [LIT-SF-3]. Grundsätzlich sollen sich Inhalte von Styleguides verlinken lassen, um schneller auf relevante oder thematisch verwandte Inhalte zugreifen zu können.

5.1.4 Import und Export von Inhalten

Ziele	Importieren von Dateien in den Styleguide, Exportieren und Drucken von Styleguides oder bestimmter Inhalte davon. Dadurch wird die Flexibilität des Systems gesteigert.
Grund	Die Praxis zeigt, dass Ersteller von Styleguides mit unterschiedlichen Werkzeugen arbeiten, um Illustrationen oder andere grafische Inhalte zu erstellen [PRAX-20]. Es hat sich ebenfalls gezeigt, dass Styleguides, aus verschiedenen Gründen, häufig ausgedruckt werden [PRAX-16]. Da Styleguides sehr umfangreich sein können [LIT-SF-2], sollten auch nur bestimmte Teile davon ausgedruckt werden können.

5.1.5 Benachrichtigung der Benutzer

Ziele Benachrichtigung bei Aufträgen oder Änderungen an Styleguides. Dadurch steigert sich bei den Benutzern das Bewusstsein, Teil eines Teams zu sein und sie werden dadurch besser in die Prozesse integriert. Eine Benachrichtigung sorgt zudem dafür, dass zu erledigende Arbeiten nicht vergessen gehen.

Grund Dieser Aspekt ist eng mit dem Aspekt der Unterstützung der Kollaboration (5.1.1 *Unterstützung der Kollaboration*) verflochten. Die Benachrichtigung der Benutzer bezieht sich insbesondere auf die Problematik der Aktualisierungen und Änderungen [LIT-PK-6]. Bei Änderungen an Inhalten oder bei der Vergabe von Aufträgen an Personen, müssen sie benachrichtigt werden. Dies besagt auch ein Punkt der Lösungsvorschläge in [LIT-WILS].

In der Praxis wollen Personen über Änderungen, die sie betreffen, informiert werden [PRAX-13]. Dies ist speziell in agilen Entwicklungsprozessen wichtig, da sich Styleguides in einem solchen Umfeld viel häufiger ändern [PRAX-5]. Damit die Benutzer jedoch nicht von einer Flut von für sie irrelevanten Benachrichtigungen überrollt werden, sollen individuelle Benachrichtigungsoptionen möglich sein.

5.1.6 Flexibles Workflowmanagement

Ziele Adaption an Firmenprozesse. Das System ist dadurch flexibel und lässt sich in bestehende Firmen- und Entwicklungsprozesse besser integrieren.

Grund In der Praxis sind Styleguides verschiedenen Firmen- und Entwicklungsprozessen unterworfen [PRAX-4], und unterschiedliche Firmen befolgen unterschiedliche Prozesse. Erstellung, Review, Abnahme und Publikation von Styleguides sind von diversen Faktoren abhängig. Diese sind die Grösse des Unternehmens, die Art und der Umfang der Änderungen an einem Styleguide oder die Anzahl der dadurch betroffenen Projekte und Produkte.

Die Literatur verweist darauf, dass bei der Erstellung von Styleguides oft der Software-Entwicklungsprozess nicht berücksichtigt wird [LIT-PK-2]. Dadurch werden Styleguides unflexibel und sind nicht in die Entwicklung integrierbar.

5.1.7 Sicherheit, Projekt- und Benutzerverwaltung

Ziele Sicherstellung der Sicherheit des Systems und Vermeidung unbefugter Manipulation mittels einer Rollenverwaltung für Benutzer des Systems. Durch die Zuweisung von Styleguides zu Projekten und Benutzern sind die Verantwortlichkeiten klar gegeben.

Grund Styleguides werden häufig von externen Firmen erstellt [PRAX-11]. Die Sicherheit des Systems muss auch für die Benutzung von ausserhalb des Unternehmens gewährleistet sein.

In der Praxis haben Personen, je nach Unternehmen, unterschiedliche Rollen und Aufgaben. Nicht alle Personen erstellen Styleguides, ändern diese, führen Reviews durch oder publizieren sie [PRAX-1]. Je nach Rolle nehmen sie unterschiedliche Aufgaben wahr, bzw. „dürfen“ unterschiedliche Tätigkeiten durchführen.

Weiter werden in der Praxis mehrere Styleguides in verschiedenen Projekten verwendet [PRAX-12]. Da typischerweise an Projekten mehrere Personen arbeiten, müssen für Projekte, mehrere Styleguides und mehrere Benutzer zugewiesen werden können. Die Verwaltung von Projekten, Styleguides und Benutzern ist die Voraussetzung dafür, dass Aufträge an Personen vergeben werden können und diese dann bei Änderungen an Styleguides benachrichtigt werden. Diese Zuweisung löst daher auch die Problematik der Verantwortlichkeiten [LIT-PK-4].

5.1.8 Visualisierung und Suche

Ziele	Anzeige aller relevanten Informationen und schnelles Auffinden benötigter Informationen durch die Unterstützung verschiedener Möglichkeiten zur Suche und Navigation. Die Effektivität und Effizienz von Systembenutzern wird dadurch gesteigert.
Grund	Die Visualisierung ist ein vielschichtiger Aspekt, auf den viele andere Aspekte einen Einfluss haben. Die Visualisierung betrifft die Darstellung der Beziehungen zwischen Projekten und Styleguides (5.1.7 <i>Sicherheit, Projekt- und Benutzerverwaltung</i>), der Kollaborationsaspekte (5.1.1 <i>Unterstützung der Kollaboration</i>), der Benachrichtigungen an die Benutzer (5.1.5 <i>Benachrichtigung der Benutzer</i>) und schlussendlich des Styleguides und dessen Inhalt selbst (5.1.3 <i>Inhalts- und Change-Management</i>). Die Literatur besagt, dass es verschiedene Arten der Benutzung gibt. Quesenbery (2001a) formuliert dies so: „[...] <i>use of the style guide shifts during the project. It starts out supplying a conceptual overview, but quickly moves into a role as a detailed reference guide.</i>“. Die Benutzung ändert sich demnach in Abhängigkeit der Projektphase [LIT-SF-4]. Diese Aussage bestätigt sich in der Praxis. Der Zugriff auf Informationen innerhalb eines Styleguides ist sehr unterschiedlich. Abhängig ist dies von der Form des Styleguides (z.B. gedruckt oder als „PDF“ verfügbar) [PRAX-18] und wie gut ein Benutzer den Styleguide kennt [PRAX-2]. Die dabei verwendeten Suchstrategien hängen zusätzlich von der konkreten Fragestellung zu einem spezifischen Problem ab [PRAX-17]. Die Problematik des Index und der Querverweise [LIT-SF-1] kann dazu führen, dass auf relevante Inhalte nur schwer zugegriffen werden kann, da man sie nicht oder nur mühsam findet. Es sollen daher unterschiedliche Einstiegspunkte, Such- und Navigationshilfen angeboten werden, um auf die gewünschten Inhalte zugreifen zu können. Durch diese Unterstützung wird auch die Problematik der Struktur und Organisation [LIT-SF-4], bei der Benutzer verschiedene Ansprüche an die Struktur von Styleguides haben, entschärft.

5.1.9 Interoperabilität mit externen Applikationen

Ziele	Konnektivität und Interoperabilität. Das System fügt sich nahtlos in das Umfeld bereits existierender Firmensoftware ein und ist dadurch flexibler.
Grund	In der Praxis werden verschiedenste Werkzeuge eingesetzt; von E-Mail und andere Kommunikationsmittel, über diverse Entwicklungsumgebungen, bis hin zu grafischen Applikationen zur Text- und Bildbearbeitung. Da ein System nicht sämtliche Bedürfnisse abdecken kann, sollte eine Anbindung an existierende Applikationen im Firmenumfeld unterstützt werden. Beispielsweise soll die Bearbeitung von bestimmten Inhalten eines Styleguides über andere Werkzeuge erfolgen, als diejenige, die durch das System selbst angeboten werden. Insbesondere für einen Ersteller von Styleguides ist es mitunter wichtig, dass er ein Werkzeug einsetzen kann das er gut kennt [PRAX-20].

5.2 Features

Aus den Aspekten haben wir eine Reihe von Features abgeleitet, welche konkretere Anforderungen an das System repräsentieren. Jedes Feature ist mit einer Referenz versehen, auf die in den Kapiteln 6 *Das Giessen des Fundamentes – Spezifikationen* und 7 *Vom Fundament zur Lösung – Konzept und Prototyp* Bezug genommen wird.

Quellen und Gewichtung

Für jedes Feature wird aufgeführt, welchen Quellen es entstammt. Quellen können die Literatur oder die Praxis sein. Quellen der Praxis umfassen dabei die Ergebnisse aus den Interviews und die Analyse von erhaltenen Styleguides.

Zusätzlich wird für die jeweilige Quelle eine Gewichtung festgelegt. Der Sinn dabei ist die Stärke der Nachweise aufzuzeigen, auf die sich ein Feature stützt. Dazu wird folgende Skala verwendet:

- + Es gibt nur Indizien und/oder wenige konkrete Nachweise aus der jeweiligen Quelle.
- ++ Es gibt mehrere konkrete Aussagen oder Nachweise aus der jeweiligen Quelle.
- +++ Es gibt viele konkrete Aussagen oder Nachweise aus der jeweiligen Quelle.

Falls ein Feature keine Gewichtung aufweist, handelt es sich um eine Idee. Ideen sind weiterführende Überlegungen, die auf eigenen Erfahrungen und Einschätzungen basieren. Sie sind aber weder durch die Literatur, noch durch die Praxis konkret belegt.

Es ist anzumerken, dass die Gewichtung **ausschliesslich** auf der Basis der untersuchten Literatur und den empirischen Recherchen basiert.

Referenztafel der Features

Referenz	Feature	Literatur	Praxis
Unterstützung der Kollaboration			
FEAT-KOL-1	Styleguides oder bestimmte Inhalte davon können von Benutzern mit Kommentaren und/oder Wahlstimmen versehen werden.	+++	++
FEAT-KOL-2	Für die Erstellung, Reviews oder Abnahmen von Styleguides können Aufträge vergeben werden.		++
FEAT-KOL-3	Das System muss ein kollaboratives Editieren an Styleguides unterstützen.		+
Inhalts- und Change-Management			
FEAT-INCH-1	Das System muss die Erstellung von Styleguides mit unterschiedlicher Struktur unterstützen.	++	+++
FEAT-INCH-2	Das System muss verschiedene Arten von Inhalt unterstützen (Text, Bild, Video, Audio, Code, interaktiver Inhalt).	+	++
FEAT-INCH-3	Styleguides und Inhalte müssen versioniert werden können.		+++
FEAT-INCH-4	Inhalte von Styleguides müssen zueinander verlinkt werden können.	++	
Import und Export von Inhalten			
FEAT-IMEX-1	Das System ermöglicht den Import von Dateien in die Struktur des Styleguides.		+
FEAT-IMEX-2	Das System ermöglicht das Exportieren oder Drucken ganzer Styleguides oder bestimmter Inhalte.		+++

Referenz	Feature	Literatur	Praxis
Wiederverwendbarkeit von Inhalten			
FEAT-WV-1	Das System erlaubt die Wiederverwendung von Inhalten für mehrere Styleguides.	+	+
Benachrichtigung der Benutzer			
FEAT-BEN-1	Bei Änderungen am Styleguide, müssen Benutzer informiert werden.	+	++
FEAT-BEN-2	Benutzer müssen über zu bearbeitende Aufträge informiert werden.		+++
FEAT-BEN-3	Das System unterstützt individuelle Benachrichtigungsoptionen.		
Flexibles Workflowmanagement			
FEAT-FLEX-1	Das System unterstützt Mechanismen zur flexiblen Gestaltung von Prozessabläufen (Erstellung, Review, Abnahme, Publikation).		+++
Sicherheit, Projekt- und Benutzerverwaltung			
FEAT-SPR-1	Das System muss eine Authentifizierung von Benutzern unterstützen.		
FEAT-SPR-2	Das System muss eine Autorisierung von Benutzern unterstützen.		+
FEAT-SPR-3	Das System ermöglicht die Verwaltung von Styleguides in Bezug auf Projekte, Benutzer und Rollen.		++
Visualisierung und Suche			
FEAT-VIS-1	Das System zeigt die Beziehungen von Projekten und Styleguides an.		++
FEAT-VIS-2	Das System zeigt alle relevanten Informationen zum Styleguide an, Inhalt und Metadaten.	++	++
FEAT-VIS-3	Das System zeigt alle relevanten Informationen zu den Prozessen an (Aufträge und Änderungen)		+
FEAT-VIS-4	Das System bietet verschiedene Navigations- und Zugriffsmöglichkeiten auf Styleguides und deren Inhalte (Inhaltsverzeichnis, Index, Bookmarks und Tags)	+++	++
FEAT-VIS-5	Das System unterstützt eine Volltextsuche über den Inhalt von Styleguides.		++
FEAT-VIS-6	Das System bietet die Filterung von Suchresultaten anhand verschiedener Kriterien.		
Interoperabilität mit externen Applikationen			
FEAT-IO-1	Das System erlaubt eine Anbindung externer Applikationen wie z.B. Skype und E-Mail.		

Tabelle 20: Features des Systems

5.3 Konkurrenzanalyse

L'analyse concurrentielle consiste en un benchmark des différents produits existant sur le marché. Le benchmark peut être défini comme suit: „*Le benchmark est dans un contexte marketing une démarche d'observation et d'analyse des performances atteintes et des pratiques utilisées par la concurrence [...] dans un contexte technique le benchmark est surtout orienter vers la comparaison de performances, comme par exemple dans l'informatique.*“ (Bathelot, 2010). La technique a été adaptée et centrée sur les fonctionnalités que proposent les concurrents par rapport au cahier des charges (Features et Requirements disponible dans *Anhang D: Features und Requirements*). La méthode d'évaluation est décrite dans le chapitre 3.1.9 *Evaluation bestehender Lösungen – Konkurrenzanalyse*, l'analyse complète est disponible en *Anhang E: Konkurrenzanalyse*.

Les concurrents

Selon Arnoult (2006) il existe deux catégories de concurrents. D'une part les **concurrents directs**, qui offrent des produits ou service dans le même segment et même fourchette de prix. D'autre part les **concurrents indirects ou produits de substitution**, qui fournissent des produits ou services dans un autre segment ou une autre fourchette de prix. Les produits de substitution peuvent être défini ainsi: „*Produit pouvant être acheté et consommé au lieu et place d'un autre, [...] sans que la satisfaction des besoins du consommateur ne soit fondamentalement altérée.*“ (Lehu, 2004).

Sept produits susceptibles de remplir le cahier des charges ont été sélectionnés et évalués. Les produits sont d'abord listés dans la table ci-dessous, puis décrit brièvement:

Catégorie	Type	Produits
Concurrents directs	Application dédiée	QUINCE PRO
	Medias actuels / traditionnels	WORD, PDF, version papier
Concurrents indirects	Produit de substitution, CMS	TYPO3
	Produit de substitution, outils collaboratifs (WIKI)	CONFLUENCE, MEDIAWIKI

Tabelle 21: Liste des produits concurrents du Benchmarking

QUINCE PRO

„*Quince Pro™ is a private, secure and organized way to collaborate, communicate and cultivate private UX design libraries to ensure consistent user experiences across teams, departments and companies.*“ (Infragistics, 2010). Cette application est disponible depuis mars 2010. Il y a peu d'information pour l'instant sur la popularité et le succès commercial de cette application. Néanmoins, le design assez lourd et le choix de Silverlight comme Framework de base ne semble pas faire l'unanimité des utilisateurs, comme le montre les commentaires laissés sur le blog de Fadeyev (2009).

TYPO3

TYPO3 est un CMS Open Source pour créer et gérer des sites web. Il a une très bonne compatibilité avec les bases de données et les différents navigateurs internet. Très populaire selon l'organisation qui le gère, selon eux plus de 300000 sites web seraient réalisés avec ce CMS (TYPO3, 2010). Il est simple d'utilisation, mais peut „frustrer“ les utilisateurs débutants (Free Host Review, 2010).

Médias et techniques actuels

L'avantage des médias actuels tel que des fichiers WORD de Microsoft ou PDF d'Adobe, c'est leur interopérabilité. La plupart des ordinateurs peuvent ouvrir et lire ces fichiers. WORD et ACROBAT

intègrent des outils de collaboration comme le suivi des modifications. Par contre la gestion des versions, la réutilisabilité du contenu, le multimédia ne sont pas ou peu supportés. Les possibilités de recherche sont également limitées aux tables des matières, indexes et la recherche textuelle lorsque le document est ouvert en format électronique.

CONFLUENCE

CONFLUENCE est un outil collaboratif de type WIKI, utilisé pour gérer le savoir, pour l'intranet ou la gestion de la documentation de l'entreprise. Il intègre Microsoft SHAREPOINT ainsi que la suite Office de Microsoft. Il intègre un véritable éditeur WYSIWYG au contraire d'un WIKI standard. Il existe différents plugins disponibles. Il permet la gestion avancée des permissions et des utilisateurs. Le support et l'accès au code source est payant (Chaney, 2007). Selon Atlassian (2010), 9300 organisations de 108 pays utilisent déjà leur produit.

MEDIAWIKI

„[...] is a free software wiki package written in PHP, originally for use on Wikipedia.“ (MediaWiki.org, 2007). MEDIAWIKI est Open Source et libre d'accès, sous licence GPL. Les avantages de WIKI sont un design simple et connu, des fonctionnalités de base tel que la gestion des pages, des images, des utilisateurs. La gestion des instances de MEDIAWIKI est par contre moins aisée, les updates et la configuration nécessitent de bonnes connaissances en PHP (Chaney, 2007).

Les résultats de l'analyse

Les résultats sont assez serrés entre les outils collaboratifs, TYPO3 et QUINCE PRO:

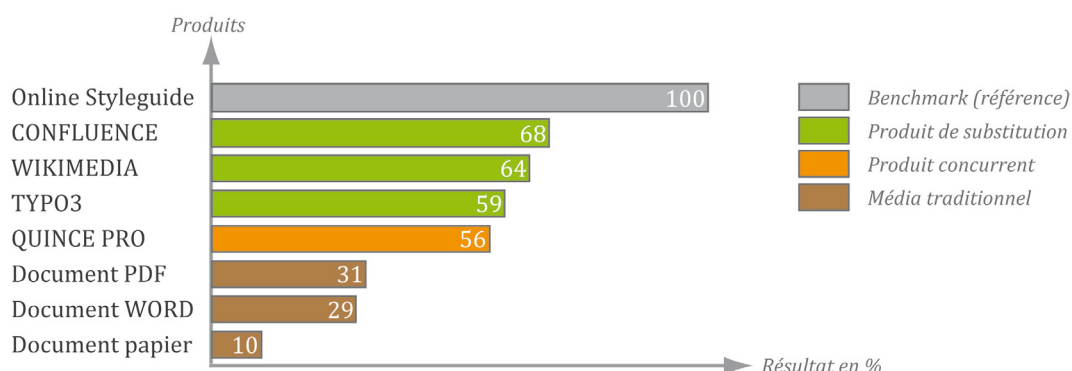


Abb. 20: Résultats de l'analyse concurrentielle (Benchmarking)

Beaucoup de fonctions sont disponibles dans les produits analysés (hormis les médias actuels), par ex. laisser des commentaires, des recherches performantes, la gestion des versions. CONFLUENCE semble mieux prendre en charge le multimédia et le code source, tandis que la gestion du Workflow est plus poussée dans TYPO3. Les médias traditionnels montrent sans surprise de grosses lacunes, entre autre dans la gestion (Workflow, utilisateurs, permissions) et le multimédia. Il est de plus difficile de comparer les prix des produits en raison des différents modèles de vente (installation locale ou Cloud-Computing). MEDIAWIKI et TYPO3 sont sous licence GPL et donc ne coûte rien. Pour pouvoir les comparer à QUINCE PRO, il faudrait pouvoir déterminer le prix de l'hébergement par utilisateur, ce qui sort du cadre de ce travail.

En résumé aucun outil ne remplit totalement les critères et le cahier des charges, même QUINCE PRO. Les produits de substitution ont chacun leurs points forts et leurs points faibles, dépendant du domaine et des objectifs de l'application. À noter: QUINCE PRO ne permet aucune installation locale.

6 Das Giessen des Fundamentes – Spezifikationen

Dieses Kapitel befasst sich mit der detaillierteren Spezifikation des Systems. Basis der Spezifikationen bilden dabei die Features (*Tabelle 20: Features des Systems*, bzw. *Anhang D: Features und Requirements*) und die Personas (*Anhang B: Personas*).

Aufgrund dieser Ergebnisse haben wir weitere Modelle erstellt, die in jeweils nachfolgenden Kapiteln beschrieben werden:

- **Concept Model**
Gibt einen Überblick über die Konzeption des Systems und dessen Zusammenhänge anhand verschiedener Konzepte und Beziehungen.
- **Use-Case Spezifikation**
Spezifiziert die Akteure und deren Anwendungsfälle (*Use-Cases*).
- **Strukturdiagramm**
Spezifiziert die Struktur und den Aufbau eines Styleguides.
- **Zustandsdiagramm**
spezifiziert die verschiedenen Zustände, die ein Styleguide im Verlaufe seines Lebens einnehmen kann.

Mit Ausnahme des Concept Models, sind die in diesem Kapitel beschriebenen Modelle und Diagramme jeweils eine Zusammenfassung der obengenannten Themen. Die detaillierten Spezifikationen sind in entsprechenden Anhängen abgelegt.

Referenzierung auf Features

Verschiedene nachfolgende Kapitel nehmen entweder Bezug auf die spezifizierten Features (*Tabelle 20: Features des Systems*) oder die Fazits aus der Praxis (*Tabelle 19: Récapitulatif des découvertes de la recherche empirique*). Der Bezug findet dabei anhand [eckiger Klammern] statt.

6.1 Concept Model

Der Zweck des Concept Models ist es, eine Übersicht über das System und das Umfeld, in dem es eingesetzt werden soll, zu geben. Es dient der Veranschaulichung der wichtigsten Konzepte und deren Beziehungen zueinander, welche die Basis für unser System bilden. Die Konzepte stammen dabei aus den Recherchen aus Literatur und Praxis, den Personas, sowie den Features und Requirements an das System. Eine Einführung in die Thematik von Concept Models findet sich unter *Anhang M: Hintergrund zu Concept Models*.

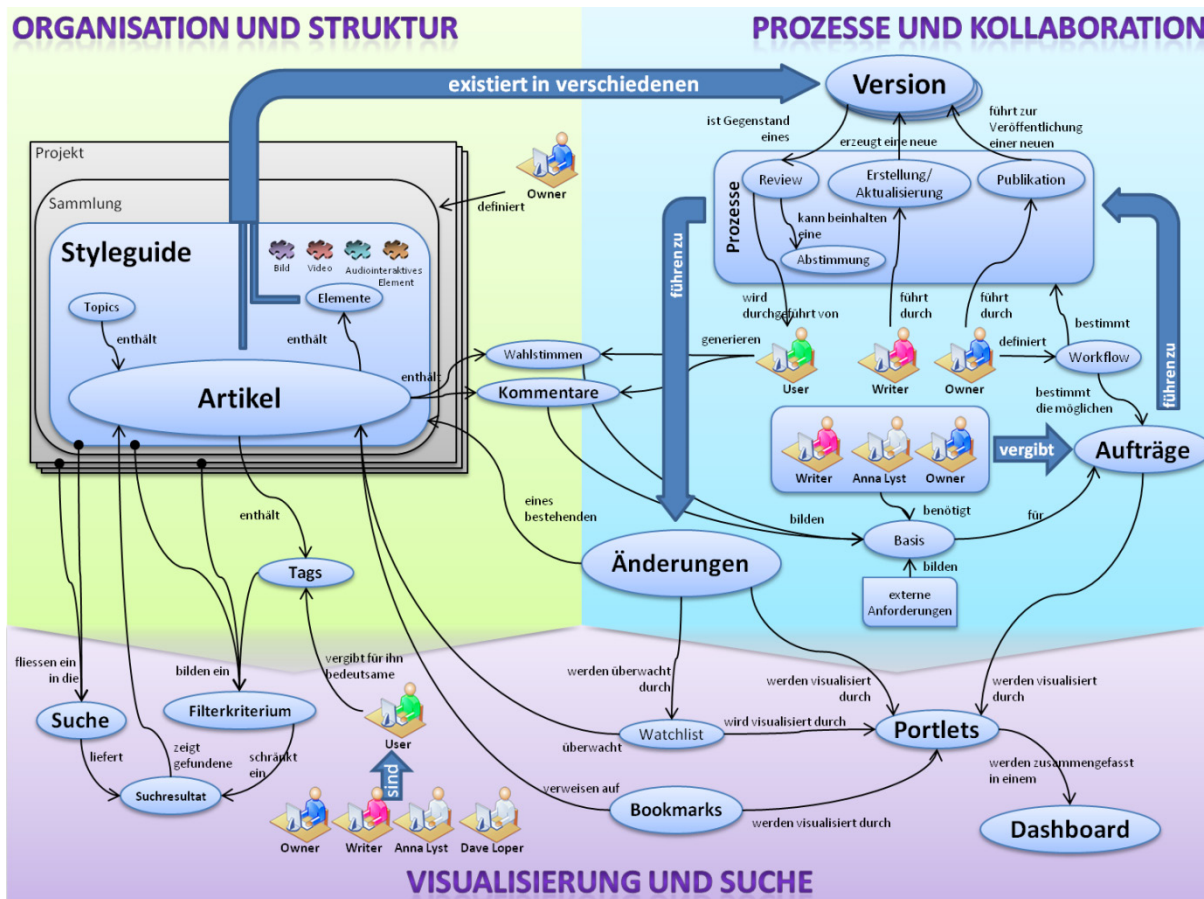


Abb. 21: Concept Model des Systems

Das zu konzipierende System umfasst viele verschiedene Konzepte, die von jeweils einer anderen „Qualität“ sind. Beispielsweise existieren Konzepte, welche die Prozesse und Kollaboration betreffen. Diese widerspiegeln sozusagen die „Dynamik“ des Systems. Es existieren aber auch Konzepte, welche die „Statik“ des Systems widergeben. Diese betreffen insbesondere die Organisation und die Struktur eines Styleguides. Weiterhin existieren Konzepte, welche das User-Interface direkt betreffen. Um die verschiedenen Konzepte in einem Diagramm dennoch strukturiert darzustellen, wird das Concept Model in drei „Konzeptbereiche“ aufgeteilt:

- **Konzepte der Organisation und Struktur:** Diese Konzepte widerspiegeln die Struktur und Organisation, der im System verwalteten Styleguides.
- **Konzepte der Prozesse und Kollaboration:** Hierbei handelt es sich um Konzepte, welche die Prozesse und die Kollaboration verschiedener Benutzer des Systems widerspiegeln.
- **Konzepte der Visualisierung und Suche:** Umfasst Konzepte, welche direkt das User-Interface des Systems betreffen.

6.1.1 Organisation und Struktur

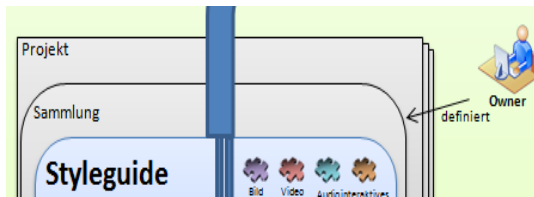


Abb. 22: Concept Model - Organisation

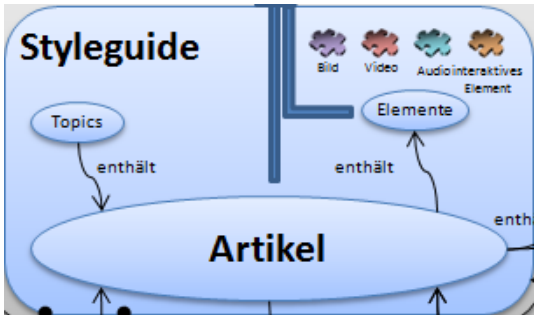


Abb. 23: Concept Model - Struktur eines Styleguide

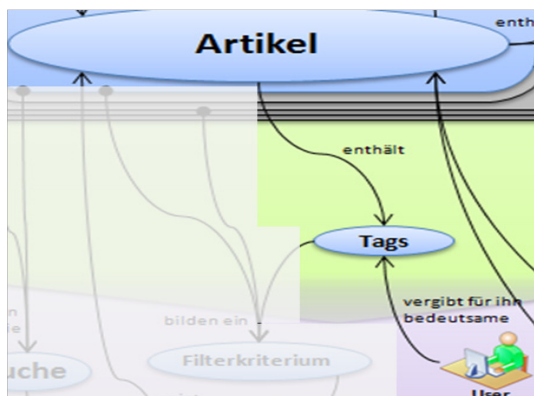


Abb. 24: Concept Model - Artikel und Tags



Abb. 25: Concept Model - Artikel, Kommentare und Wahlstimmen

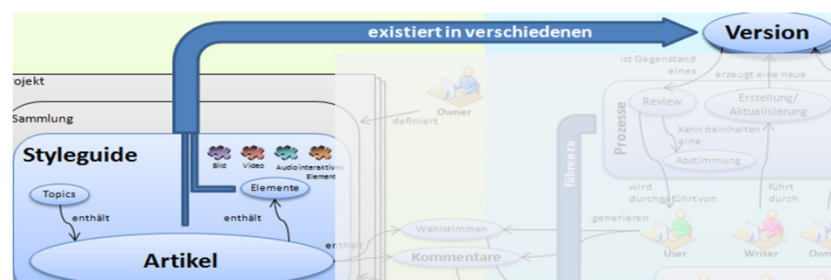


Abb. 26: Concept Model - Styleguide, Artikel, Elemente und Versionen

Ein **Projekt**, das in einem Unternehmen durchgeführt wird, beinhaltet eine **Sammlung** von **Styleguides**, die für das Projekt zur Anwendung gelangen sollen [FEAT-SPR-3]. Es ist dabei die Aufgabe von Ottmar Owner, diese Sammlung zu definieren.

Die Struktur eines Styleguides ist dreistufig [FEAT-INCH-1]. **Topics** sind analog zu Kapiteln in einem Dokument. Sie gliedern einen Styleguide in Themenbereiche (z.B. Layout, Widgets, Farben). **Topics** enthalten **Artikel**. **Artikel** enthalten die eigentlichen Informationen zu einem bestimmten Thema. **Artikel** können **Elemente** enthalten. Bei **Elementen** kann es sich um Bilder, Videos, etc. handeln [FEAT-INCH-2]. Die Struktur eines Styleguides wird in Kapitel 6.3 *Strukturdiagramm* ausführlicher beschrieben.

Artikel können auch **Tags** enthalten [FEAT-VIS-6]. **Tags** werden von **Usern** für **Artikel** vergeben. Sie gelten jeweils nur für den User selbst und sind anderen **Usern** nicht zugänglich. **Tags** dienen als ein Filterkriterium zur Einschränkung der Suchresultate. Ein Styleguide enthält noch weitere Metadaten, die jedoch nicht als eigene Konzepte im Concept Model erfasst sind. Eine ausführlichere Beschreibung dieser Metadaten ist im *Anhang G: Strukturdiagramm* enthalten.

Ein **Artikel** kann **Kommentare** und **Wahlstimmen** enthalten [FEAT-KOL-1]. **Kommentare** werden von **Usern**

während eines Reviews oder nach der Publikation eines Styleguides erfasst. **Wahlstimmen** werden nur während eines Reviews abgegeben. Dabei wird über Varianten eines Artikels abgestimmt.

Styleguides existieren in mehreren **Versionen** [FEAT-INCH-3]. Diese Versionen sind ein Produkt des Erstellungs- oder Änderungsprozesses. In Styleguides enthaltene **Artikel** und **Elemente** können ebenfalls in unterschiedlichen Versionen existieren.

6.1.2 Prozesse und Kollaboration

Als Startpunkt für die Beschreibung dieser Konzepte und Beziehungen dient die Auftragsvergabe.

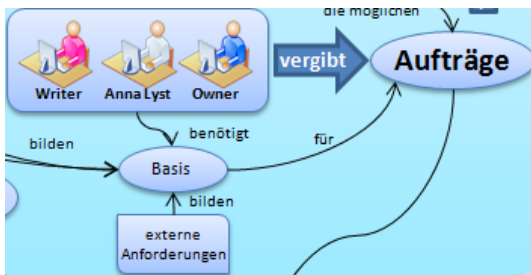


Abb. 27: Concept Model - Aufträge

Ottmar Owner, Hans Writer und Anna Lyst vergeben Aufträge an andere Personen [FEAT-KOL-2]. Aufträge können dabei die Erstellung oder die Änderung eines Styleguides oder die Einladung zur Teilnahme an Reviews sein. Basis für Aufträge bilden bestimmte Gründe. Diese können Kommentare oder Wahlstimmen (aus Reviews) oder externe Anforderungen sein (z.B. Auftrag der Firmenleitung zur Einführung von Styleguides).

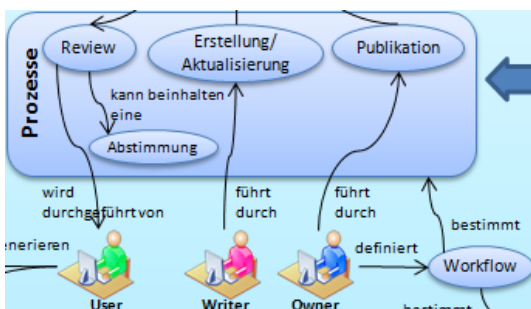


Abb. 28: Concept Model - Prozesse

Aufträge führen zur Auslösung eines Prozesses. Dabei kann es sich um einen Review, eine Aktualisierung (oder Erstellung) oder eine Publikation (Veröffentlichung) von Styleguides handeln. Teil eines Reviews kann auch eine Abstimmung sein. Dabei wird über Varianten desselben Artikels abgestimmt (z.B. verschiedene Farben). Die Reviewer vergeben dabei Wahlstimmen. Erstellungen und Aktualisierungen von Styleguides werden typischerweise von Hans Writer durchgeführt. Nach der Aktualisierung durch Hans Writer, und erfolgtem Review durch die User, publiziert Ottmar Owner den Styleguide. Der Workflow wird dabei durch Ottmar Owner definiert [FEAT-FLEX-1]. Ein Workflow definiert, welche Phasen ein Styleguide durchlaufen muss. Es kann sein, dass nicht jede Änderung einen Review erfordert (z.B. kleine Änderungen). Diese Thematik ist in Kapitel 6.4 Zustandsdiagramm genauer beschrieben. Ein Workflow definiert ebenfalls, welche möglichen Aufträge für das System in Frage kommen. Es bestünde die Möglichkeit, den Workflow so zu definieren, dass keine Reviews stattfinden müssen. Das System könnte beispielsweise neue oder geänderte Styleguides automatisch publizieren.



Abb. 29: Concept Model - Version und Prozesse

Dreh- und Angelpunkt von Prozessen bilden jeweils die Versionen von Styleguides [FEAT-INCH-3]. Die Erstellung oder Änderung eines Styleguides führt zu einer neuen Version. Diese Version bildet dann die Basis für einen Review oder eine Publikation. Prozesse führen zu Änderungen. Darunter fallen nicht nur Änderungen bezüglich des Inhalts von Styleguides, sondern auch Änderungen an deren Metadaten, beispielsweise neue Kommentare oder Wahlstimmen für verschiedene Varianten von Artikeln. Änderungen können aber auch den Zustand des Systems selbst betreffen (z.B. neue Aufträge). Änderungen sind die Grundlage für eine Benachrichtigung an die betroffenen User [FEAT-BEN-1, FEAT-BEN-2].

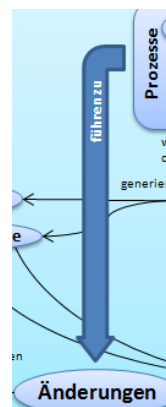


Abb. 30: Concept Model - Änderungen

6.1.3 Visualisierung und Suche

Die Konzepte der Visualisierung und Suche umfassen Konzepte, die in Zusammenhang mit der Anzeige und der Suche von Informationen stehen. Diese Konzepte betreffen das User-Interface selbst. In diesem Kapitel werden nur die wichtigsten Konzepte beschrieben, die ausführliche Beschreibung und die Umsetzung dieser Konzepte im Prototyp sind im Kapitel 7 *Vom Fundament zur Lösung – Konzept und Prototyp* enthalten.

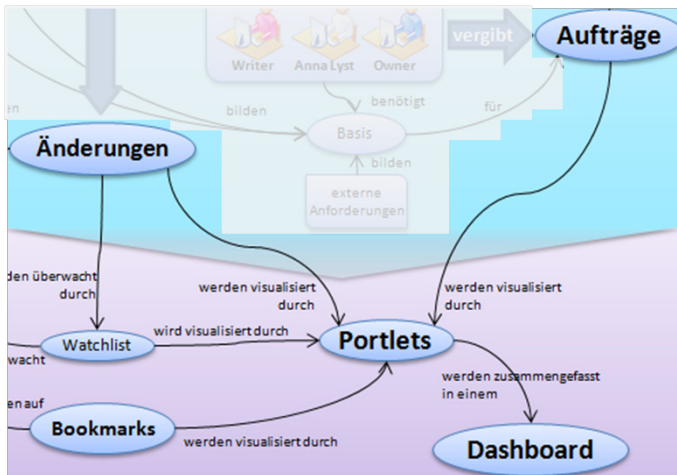


Abb. 31: Concept Model - Portlets und Dashboard

Ein Portlet visualisiert auch **Bookmarks**. Dabei handelt es sich um direkte Links zu Artikeln und dienen somit dem schnellen Zugriff darauf [FEAT-VIS-4]. Ein **Dashboard** vereinigt alle Portlets in einer übersichtlichen Art und Weise.

Ein zentrales Konzept sind die **Portlets**. Portlets visualisieren Informationen zu verschiedenen Aspekten. Beispielsweise werden **Aufträge**, die vergeben werden oder **Änderungen** an Artikeln in Portlets visualisiert [FEAT-VIS-3]. Portlets dienen aber auch als weitere Einstiegspunkte und Navigationsmöglichkeiten, um auf bestimmte Inhalte eines Styleguides zuzugreifen [FEAT-VIS-4]. Beispielsweise visualisiert ein Portlet eine **Watchlist**. Ihre Aufgabe ist es, **Änderungen** an bestimmten Artikeln zu überwachen. Ein

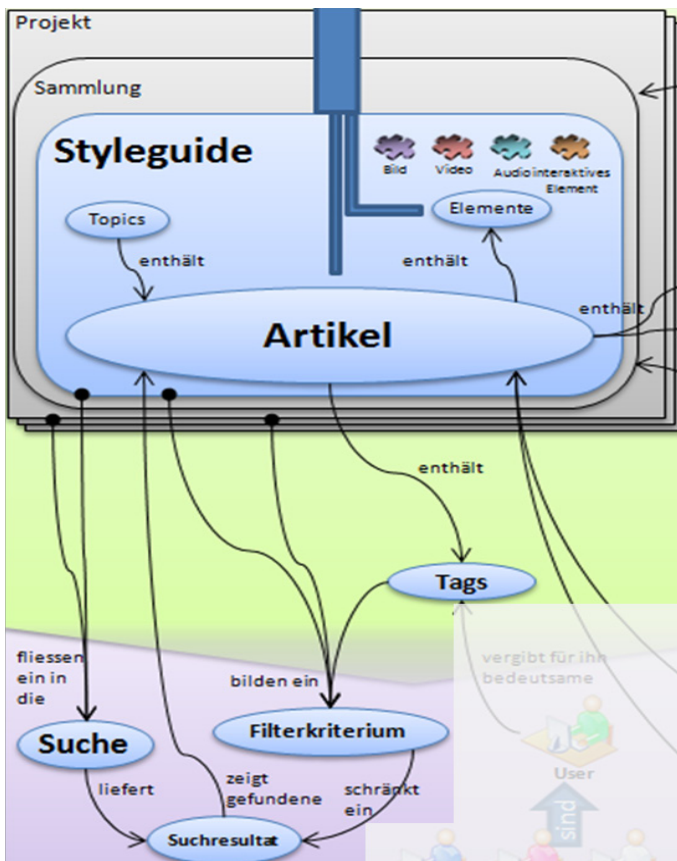


Abb. 32: Concept Model - Suchen und Filtern

Für die **Suche** nach Informationen werden die Inhalte von Styleguides aller Projekte, für die ein User zugewiesen wurde, und entsprechend alle Sammlungen berücksichtigt [FEAT-VIS-5]. Bei der Suche handelt es sich um eine „Volltextsuche“. Das Kriterium der Suche ist also ein textueller Begriff. Als **Suchresultat** liefert die Suche sämtliche Artikel, für welche die Suche erfolgreich war. Das Suchresultat kann anhand eines **Filterkriteriums** eingeschränkt werden [FEAT-VIS-6]. Dabei kann es sich um ein bestimmtes Projekt, ein bestimmter Styleguide, ein Tag oder eine Kombination von Allem sein.

6.2 Use-Case Spezifikation

Les System Use-Cases (SUC) sont la base de la méthodologie RUP. Ce chapitre présente un résumé du document de spécification des SUC disponible en *Anhang I: Use-Case Spezifikationen* ainsi que les commentaires sur la modélisation. Seul les SUC de *User*, *Coordinator* et *Realizer* sont spécifiés et listés dans ce travail. Les SUC des autres acteurs sont en dehors du scope et n'ont pas été réalisés.

6.2.1 Aktoren

Définir les acteurs systèmes n'est pas une tâche facile. Comme l'écrit Bittner et al. (2002) „*Don't Confuse Actors with Organizational roles or job titles*“. Il ne faut surtout pas les mélanger avec les acteurs métiers ou les Personas. C'est pourquoi les noms des acteurs systèmes ont été définis en fonction du type de travail à effectuer sur le système. Les acteurs du système généralisent *User*:

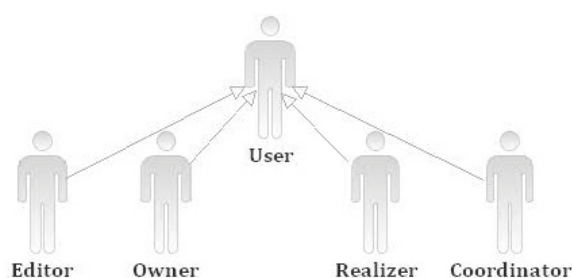


Abb. 33: System Actors

Les acteurs systèmes sont listés ci-dessous, avec leurs tâches et les acteurs métiers correspondant.

Actors	Acteurs métiers	Tâches
User	Public Relation, utilisateur, etc.	Consulte et imprime les styleguides (SG)
Editor	Designer, Interaction Designer	Ecrit et réalise les styleguides
Owner	Chef de projet, manager	Gère les projets, les styleguides et les révisions
Realizer	Développeur	Fait des révisions (en plus de User)
Coordinator	Req. Engineer, Business Analyst	En plus, coordonne la création du SG, organise les révisions

Tabelle 22: Description des System Actors

6.2.2 Berechtigungen

Le système définit par défaut les permissions et fonctions à disposition pour chaque acteur. Cependant les utilisateurs peuvent avoir plusieurs rôles et tâches au sein de leur entreprise [PRAX-1]. Le système gère donc les droits de manière flexible. Les permissions par défaut sont marquées par un *x*. Les croix entre parenthèses indiquent une option, une permission que l'utilisateur peut recevoir.

Actors	Lire	Com- menter	Editer	Mandater une tâche	Mandater une révision	Faire une révision	Approuver un SG	Publier un SG
User	x	x						
Editor	x	x	x			(x)		(x)
Owner	x	x		x	x	x	x	x
Realizer	x	x	(x)			x		
Coordinator	x	x	(x)	x	x	x		(x)

Tabelle 23: Gestion des permissions

Cette table permet également d'avoir un aperçu de quelques fonctions de *Owner* et *Editor*.

6.2.3 Use-Case Model

L'illustration ci-dessous montre les Use-Cases des acteurs *User*, *Coordinator* et *Realizer*.

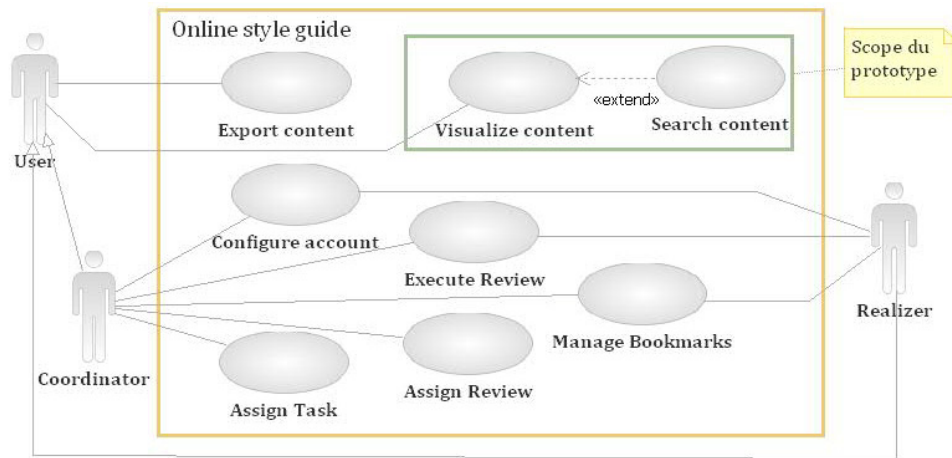


Abb. 34: Use-Case Model (*User*, *Coordinator*, *Realizer*)

Les UC de base de User

Visualize content n'est pas écrit du même point de vue que les autres SUC. Il montre une action du système et non un cas d'utilisation de celui-ci. L'équivalent du point de vue utilisateur serait par ex. *Read article*. La première formulation permet une description plus aisée de ce que fait le système pour afficher le SG à l'écran. Pour des raisons de lisibilité du Use-Case Model l'ajout de commentaires, de tags ou de favoris sont modélisés comme des variantes de *Visualize content*. Ces fonctions sont exécutées pendant la visualisation des articles du SG. Elles pourraient être modélisées sous forme de cas d'utilisation séparés qui étendent *Visualize content*.

Search content étend *Visualize content* parce que la recherche est optionnelle. L'utilisateur peut déjà savoir où se trouve l'information. Le but de l'utilisateur est de consulter le styleguide, d'en lire le contenu. La recherche n'est qu'un moyen d'y arriver et n'a pas de sens en tant que SUC autonome. *User* peut également exporter avec *Export content* le styleguide ou une partie de celui-ci en format PDF ou impression papier. Il serait également envisageable que ce SUC étende *Visualize content*.

Les UC de collaboration

Coordinator a des tâches de coordination au sein du projet. Il dispose pour ça des UC *Assign Review* pour organiser des révisions et *Assign task* pour assigner des tâches, par exemple une extension du styleguide à *Editor*. Tous les acteurs sauf *User* peuvent exécuter des révisions grâce à *Execute review* et peuvent également voter sur des variantes de design que *Editor* propose. Cette dernière fonctionnalité est directement intégrée dans *Execute review*, vu qu'il s'agit d'un cas spécial d'une révision.

Les autres UC

Coordinator et *Realizer* peuvent gérer leurs favoris avec *Manage bookmarks* et configurer leur compte et leurs préférences d'affichage avec *Configure account*. *User* peut être comparé à un invité, il n'a pas de compte à lui, donc ne doit pas le configurer, accède au SG en lecture et ne fait pas de révision. Dans une première itération ce niveau de détail suffit aux discussions et aux premiers prototypages.

6.3 Strukturdiagramm

Das Strukturdiagramm zeigt die Struktur, bzw. den Aufbau eines Styleguides im System in Form eines UML-Klassendiagrammes. Die Basis für die Struktur des Styleguides bilden die Features (Tabelle 20: *Features des Systems*) aus den Aspekten des Inhalts- und Change-Managements [FEAT-INCH] und der Wiederverwendung [FEAT-WV]. Dieses Kapitel ist eine Zusammenfassung der im Anhang G: *Strukturdiagramm* enthaltenen Informationen.

6.3.1 Struktur eines Styleguides

Um die Anforderungen bezüglich einer flexiblen Struktur, der Wiederverwendbarkeit und der Versionierung zu erfüllen, wird ein modularer Ansatz zur Strukturierung eines Styleguides verfolgt.

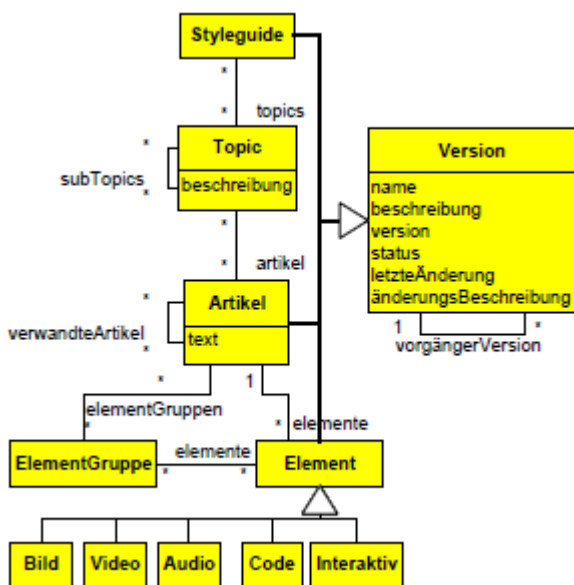


Abb. 35: Struktur eines Styleguide

Ein Styleguide ist typischerweise in bestimmte Themenbereiche gegliedert (Layout, Farben, etc.) Diese Gliederung findet mittels **Topics** statt.

Die eigentlichen Informationen zu einem Thema sind in **Artikeln** enthalten. Um Bezug zu thematisch verwandten Artikeln zu nehmen, können Artikel über *verwandteArtikel* miteinander verlinkt werden.

Das Gros der in Styleguides enthaltenen Informationen bildet praktisch immer Text. Daher enthalten Artikel direkt schon Text. Gute Styleguides vermitteln Informationen aber nicht nur durch Text, sondern mittels einer breiten Palette zusätzlicher **Elemente** (Bilder, Code, etc.).

6.3.2 Wiederverwendbarkeit, Versionierung und Metadaten

Der modulare Ansatz, bzw. die Aufteilung eines Styleguides in Topics, Artikel und Elemente in eigenen Klassen, ermöglicht eine Entkoppelung von Inhalt und Struktur. Dadurch lassen sich Artikel und Elemente für andere (oder denselben) Styleguides wiederverwenden. Ebenfalls können dadurch mehrere Personen verschiedene Teile eines Styleguides „gleichzeitig“ bearbeiten.

Damit Änderungen von Styleguides, Artikel und Elementen verfolgt und verglichen werden können, sind sie einer Versionierung unterworfen (*Version*). Damit auf die gesamte Änderungsgeschichte von Inhalten zugegriffen werden kann, sind die Versionen miteinander verknüpft (*vorgängerVersion*). Von einer Versionierung explizit ausgenommen sind Topics. Dies hat zwei Gründe. Einerseits sind die potentiell wiederverwendbaren Informationen nur in Artikeln und Elementen enthalten, andererseits würde die Businesslogik für die Versionierung unnötig komplex.

Um Informationen, beispielsweise über Status, Datum einer Änderung oder den Namen des Verfassers eines bestimmten Inhaltes, festzuhalten, enthalten Styleguides, Artikel und Elemente Metadaten. Als Metadaten enthält ein Artikel zusätzlich noch Kommentare, Wahlstimmen und Tags.

6.4 Zustandsdiagramm

Das Zustandsdiagramm zeigt die verschiedenen System-Zustände, die ein Styleguide im Verlaufe seines Lebens einnehmen kann. Es beschreibt die Zustandstransitionen, deren Auslöser und welche Benutzerrechte dafür notwendig sind. Die Notation des Zustandsdiagrammes ist dabei an UML angelehnt. Die Basis für das Diagramm bilden die Features bezüglich der flexiblen Gestaltung von Prozessabläufen [FEAT-FLEX-1] und der Verwaltung von Benutzerrollen [FEAT-SPR-3]. Dieses Kapitel ist eine Zusammenfassung der im *Anhang H: Zustandsdiagramm* enthaltenen Informationen.

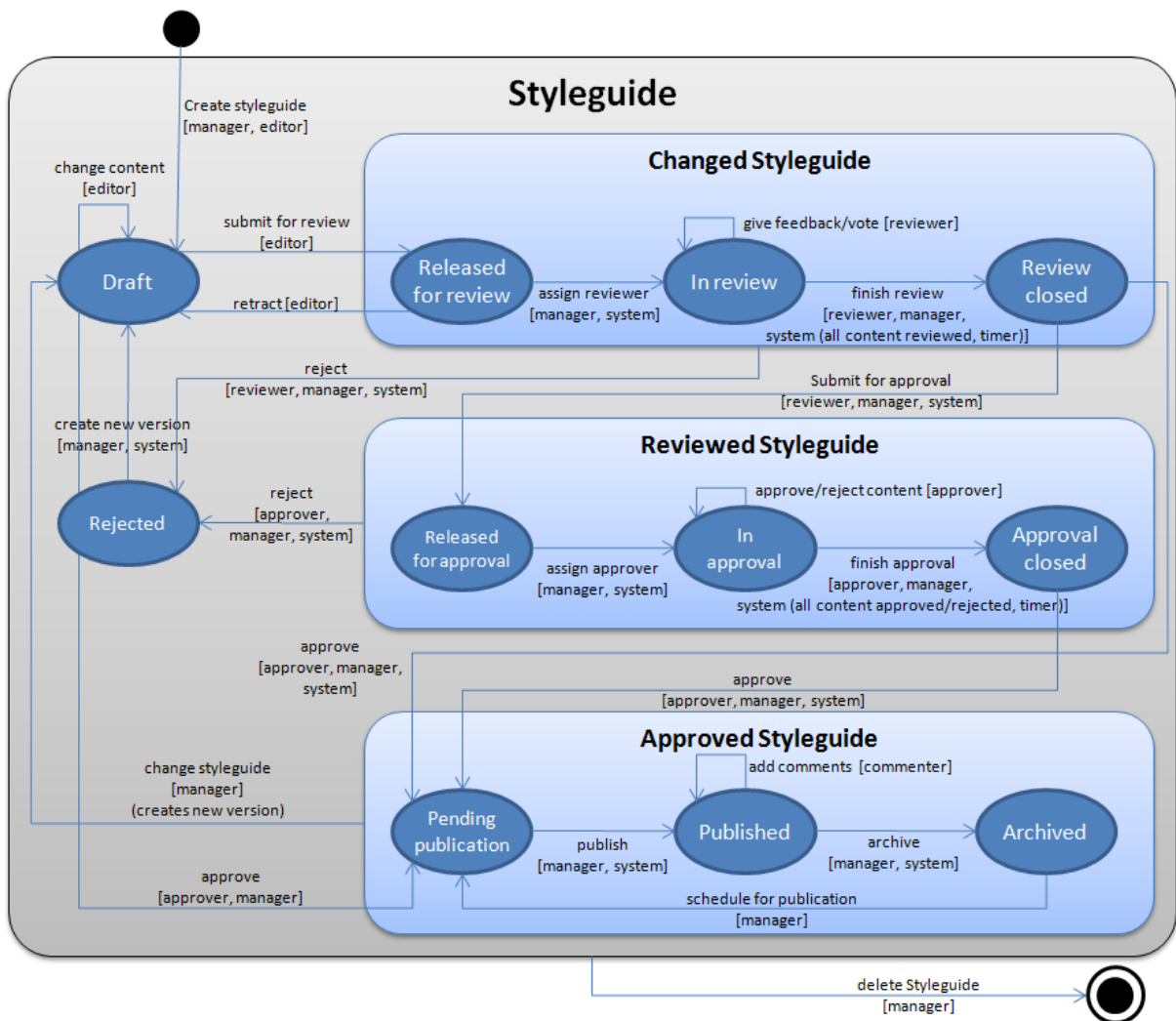


Abb. 36: Zustandsdiagramm eines Styleguides

6.4.1 Flexibilität

Die abgebildeten Zustände basieren auf den empirischen Recherchen über die jeweiligen Prozesse. Sie sind aufgrund der „komplexesten“ untersuchten Fälle definiert. Insbesondere für kleinere Unternehmen werden jedoch viele dieser Zustände nicht benötigt, da ein Styleguide nicht derart komplizierte Phasen durchläuft, wie jene für ein Grossunternehmen. Diesem Umstand wird durch die Rolle *System* Rechnung getragen, welche es erlaubt, verschiedene Zustände automatisiert, das heisst, ohne Einflussnahme seitens realer Personen zu durchlaufen. Damit ist es beispielsweise möglich, die komplette Phase der Abnahme (*Reviewed Styleguide*) zu umgehen.

6.4.2 Zustände und Transitionen

Ein Styleguide kann elf konkrete Zustände annehmen. Jedoch sind zum besseren Überblick und zur Vereinfachung der Transitionen einige Zustände in übergeordneten Zuständen (*super-states*) zusammengefasst. Die Transitionen, bzw. die Aktionen, die zu einer Zustandsänderung führen, entstammen dabei einerseits aus den Recherchen in der Praxis über die verschiedenen Styleguide-Prozesse und andererseits aus Studien diverser CMS Systeme (Vladimirskiy, 2009) und (Ng, 2004).

6.4.3 Rollen und Berechtigungen

Um bestimmte Aktionen, die zu einer Zustandsänderung führen, vornehmen zu können, müssen die Akteure über bestimmte Rechte verfügen (6.2.2 *Berechtigungen*). Dieser Zusammenhang wird im Diagramm als *Rolle* bezeichnet (*manager, reviewer, etc.*). Mit anderen Worten: eine *Rolle* im Zustandsdiagramm entspricht einem Akteur, der über die notwendigen Rechte verfügt. Im Diagramm sind die Rollen in [eckigen Klammern] aufgeführt.

7 Vom Fundament zur Lösung – Konzept und Prototyp

In diesem Kapitel wird das Konzept erläutert und die Umsetzung des Konzepts zu einem ersten konzeptionellen Prototyp auf Basis von Wireframes. Ausgangslage bilden die Features und Requirements (*Anhang D: Features und Requirements*), das Concept Model (6.1 *Concept Model*), die Use-Cases *Visualize Content* und *Search Content* (6.2.3 *Use-Case Model*) und die Szenarios der Personas *Anna Lyst* und *Dave Loper* (*Anhang B: Personas*).

Im ersten Teil wird das Konzept formuliert.

Der zweite Teil geht auf den Prototyp ein und beschreibt dessen Aufbau und Bedienkonzepte. Im Weiteren wird eine kurze Zusammenfassung über das Farbkonzept, die Symbol und Icons und über den Einsatz von typografischen Elementen aufgezeigt. Danach werden die im Prototyp verwendeten Patterns aufgelistet.

7.1 Konzept „Online-Styleguide“

Das Konzept „Online-Styleguide“ verfolgt verschiedene Ziele:

- **Vereinfachung, Vereinheitlichung und Beschleunigung** der Prozesse zur Erstellung, Verwaltung und Verwendung projektspezifischer Styleguides eines Unternehmens.
- **Flexibilität** durch Anpassung an die vorhandenen Firmen- und Entwicklungsprozesse.
- **Verfügbarkeit und Zugänglichkeit** von projektspezifischen Styleguides durch ein webbasierendes System.
- **Sicherheit und klare Verantwortlichkeiten** durch Verwendung von Verschlüsselungs- und Autorisierungstechniken und Rollenzuweisungen von Benutzern in Bezug auf Styleguides.
- **Qualitätssteigerung der Styleguides** durch verbesserte Kommunikation und Einbezug verschiedener Benutzergruppen mittels Kollaboration und als Folge daraus eine bessere Qualität und durchgehende Konsistenz von Produkten in Folge gesteigerter Akzeptanz und gegenseitigem Verständnis der Benutzergruppen von Styleguides.

Die **Vereinfachung und Vereinheitlichung** der Prozesse wird durch ein zentrales System für die Erstellung, Benutzung und Verwaltung von projektspezifischen Styleguides erreicht. Eine Vereinfachung wird auch durch die Unterstützung einer Auftragsvergabe an Benutzer erreicht. Für die **Beschleunigung** der Prozesse werden zwei Arten unterschieden:

1. **Der beschleunigte Prozess der Erstellung von Styleguides** durch die Wiederverwendung bereits existierender Inhalte, der Import- und Export-Funktionalität, der Kommentar- und Abstimmungs-Funktionalität und dem gleichzeitigen Editieren an verschiedenen Teilen.
2. **Der beschleunigte Prozess der Entwicklung des User-Interfaces**, durch klare Präsentation der Informationen (auch in mehreren Sprachen), der Zugriff darauf mittels mehrerer Einstiegspunkte (Inhaltsverzeichnis, Index, Aufträge, etc.), der Unterstützung der „Verlinkbarkeit“ von Inhalten, der Möglichkeit der Suche und des Filterns, und das Anbieten direkt wiederverwendbarer Inhalte (Quellcode, Widgets, etc.).

Die **Flexibilität** wird durch ein konfigurierbares Workflowmanagement erreicht, das eine Adaption an bestehende Firmen- und Entwicklungsprozesse erlaubt.

Die **Verfügbarkeit und Zugänglichkeit** wird dadurch erreicht, dass auf Styleguides, mittels einer Web-Applikation in einem gängigen Web-Browser, online zugegriffen werden kann.

Mit einer Benutzer- und Rechteverwaltung wird die **Sicherheit und Verantwortlichkeit** für projektspezifische Styleguides innerhalb des Systems erreicht.

Die **Qualitätssteigerung der Styleguides** wird durch kollaborative Funktionen wie Kommentarerfassung und Abstimmungen zu Styleguides oder bestimmter Inhalte erreicht.

7.2 Konzeptioneller Prototyp

Der Prototyp ist die Umsetzung des User-Interfaces des eingangs beschriebenen Konzeptes. Die Basis des konzeptionellen Prototyps bilden dabei Wireframes. Der Fokus des Prototyps liegt auf den Personas *Anna Lyst* und *Dave Loper* und den Use-Cases *Visualize Content* und *Search Content*. Es werden die dafür verantwortlichen Systemansichten *Dashboard*, *Styleguide* und *Liste der Suchresultate* abgebildet und erklärt. Die zentralen Bedienkonzepte für besagte Use-Cases werden in den jeweiligen Systemansichten erklärt.

Zunächst wird die Basis des konzeptionellen Prototyps erläutert und, darauf aufbauend, die verschiedenen Systemansichten.

7.2.1 Grundlayout

Das Grundlayout des Prototyps besteht aus den Bereichen *Header*, *Control* und *Content*. Die Aufteilung des Layouts folgt dabei den Empfehlungen über die Seitenstruktur von Lynch et al. (2008).

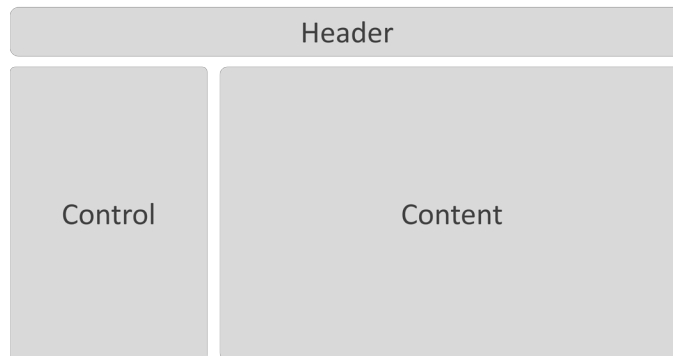


Abb. 37: Das Grundlayout zur Systemansicht „Styleguide“ und „Liste der Suchresultate“

Der **Header**-Bereich ist statisch. Er ist immer präsent und enthält das Logo und den Titel der Applikation. Er ist in weitere Unterbereiche aufgeteilt:

- **Sprachwahl** - In diesem Bereich wird die Sprachwahl angezeigt.
- **Benutzerinformation** - Hier wird der aktuell eingeloggte Benutzer angezeigt, zusätzlich gibt es eine „Hilfe“-Funktion und ein Benutzerlogout.
- **Suchfeld** - In diesem Feld ist die Suchmaske integriert.

Im **Control**-Bereich werden wahlweise die Auswahlmenüs (Projekt- und Styleguideauswahl, Inhaltsverzeichnis oder Index) für die Styleguides oder die Suchfilter angezeigt. Aufgrund der Empfehlung über die Platzierung von Hauptnavigationen in Kapitel 7.5 der *Research-Based Web Design & Usability Guidelines* (Usability.gov, 2003), werden die Auswahlmenüs in der linken Spalte angeordnet. Dies unterstützt den Aufbau einer Navigation mit Baumstruktur.

Im **Content**-Bereich wird wahlweise der Inhalt des Styleguide oder die Liste der Suchresultate dargestellt.

Dieses Grundlayout bildet die Struktur für die beiden Systemansichten *Styleguide* und *Liste der Suchresultate*.

Layout für das Dashboard

Für die Systemansicht *Dashboard* wird ein spezielles Layout verwendet, das eine reduzierte Version des Grundlayouts darstellt und nur aus den Bereichen **Header** und **Content** besteht. Der **Header**-Bereich erfüllt dabei die gleiche Funktion wie beim Grundlayout. Im **Content**-Bereich werden die *Portlets* dargestellt (7.2.3.7 *Das Portlet-Konzept*).

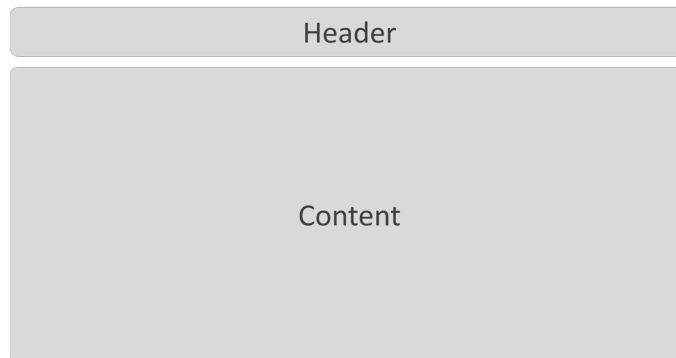


Abb. 38: Das Speziallayout für die Systemansicht „Dashboard“

7.2.2 Header

Im Header befinden sich das Logo, der Titel der Applikation und ein Bereich, der die Suchmaske, die Sprachwahl, die Benutzerinformation, die Hilfe und der Benutzerlogout enthält.



Abb. 39: Der Header-Bereich

Die Suchmaske wird im rechten Bereich des Headers platziert, da, laut einer Benutzerstudie (Shaikh, et al., 2006), Benutzer mehrheitlich einen Suchmechanismus im rechten oberen Bereich erwarten. Das Suchfeld wird prominent dargestellt, damit es vom Benutzer schnell als Solches erkannt wird.

Die Positionierung der Komponenten Sprachwahl, Benutzerinfo, Hilfe und Logout folgt der Annahme, dass diese Elemente eine eher untergeordnete Rolle in der Bedienung spielen und deshalb weniger prominent gestaltet werden. Um die Zusammengehörigkeit der Komponenten zu vermitteln werden sie in eine Gruppe zusammengefasst und auf der rechten Seite des Headers platziert.

7.2.2.1 Das Konzept der Suche

Damit Informationen in den Styleguides effektiv und effizient gefunden werden bietet sich das Konzept einer Suchfunktion an. Ausgehend von dem Feature [FEAT-VIS-5] bietet das System dem Benutzer eine Volltextsuche mit der sich die Inhalte sämtlicher im System hinterlegten Styleguides durchsuchen lassen. Die Such-Funktion ist dabei den heutigen Standards, wie sie der Benutzer von anderen populären Suchkonzepten kennt, nachempfunden. So unterstützt die Suchfunktion auch den Service der „Autovervollständigung“ (wie z.B. bei Google). Dies ermöglicht dem Benutzer seine Suchanfrage zu verfeinern, indem adäquate Suchvorschläge angeboten werden. Dies führt zu einer Effizienzsteigerung bei der Suche.

7.2.2.2 Das Konzept mehrerer Sprachen

Grundlage für dieses Konzept bildet das Non-Functional-Requirement [NFR-1]. Ein Projektteam kann aus verschiedensprachigen Personen zusammengesetzt sein. Daher kann die Bedienung und mehrsprachig erfasste Information in verschiedenen Sprachen angezeigt werden.

7.2.3 Ansichten Styleguide, Suchresultate und Dashboard

Styleguideansicht

In der Styleguideansicht werden die Inhalte des Styleguides angezeigt. Der Aufbau dieser Ansicht ist so konzipiert, dass der Benutzer eine ihm vertraute Struktur wiedererkennt. Dies hilft ihm (als Metapher), seine Erfahrungen mit ähnlichen Dokumenten aus dem Alltag anzuwenden.

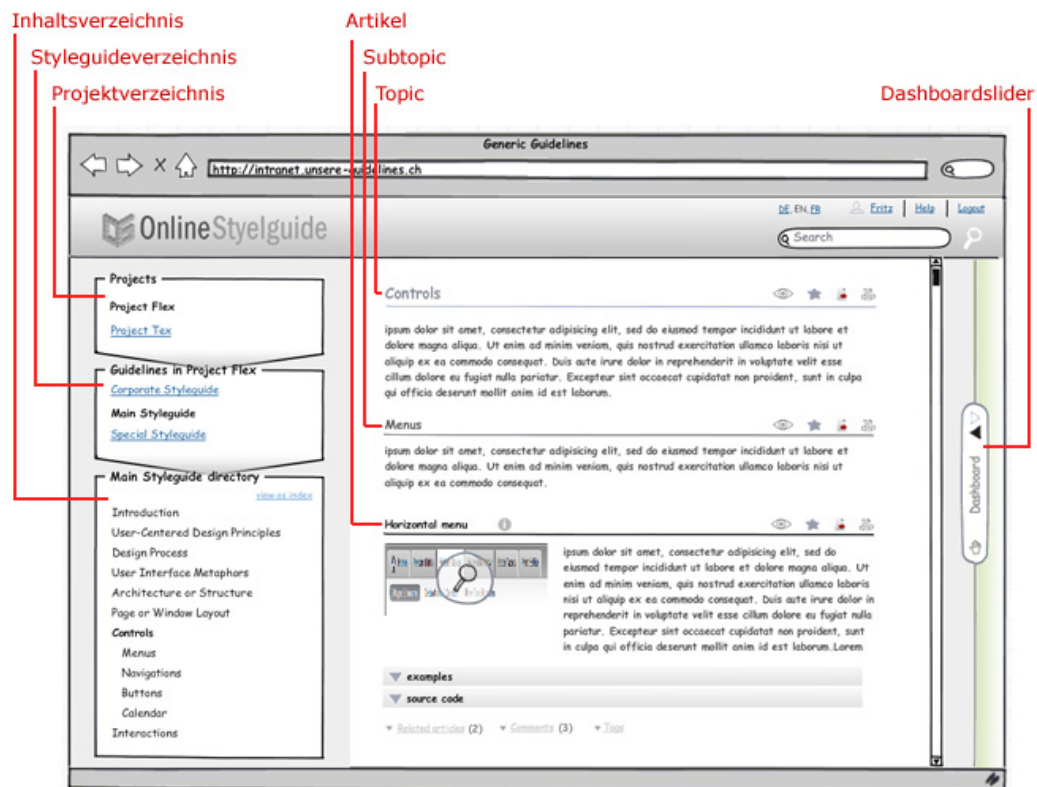


Abb. 40: Die Styleguideansicht

Die Styleguideansicht ist in einen **Control**- und einem **Content**-Bereich aufgeteilt. Grundlage für diese Anordnung bilden die Features [FEAT-VIS-1] und [FEAT-VIS-2].

Im **Control**-Bereich werden die folgenden drei Komponenten dargestellt:

- **Projektverzeichnis** – Bildet die erste Stufe der Navigation. Es wird eine Liste aller Projekte angezeigt, für die der Benutzer zugewiesen ist.
- **Styleguideverzeichnis** – Bildet die zweite Stufe der Navigation. Hier wird eine Liste aller Styleguides angezeigt, die dem Projekt zugewiesen sind.
- **Inhaltsverzeichnis/Index** – Zeigt das Inhaltsverzeichnis eines selektierten Styleguides als navigierbare Baumstruktur. Alternativ dazu, kann hier zum Index gewechselt werden.

Im **Content**-Bereich wird für eine Selektion im Inhaltsverzeichnis oder Index, der entsprechende Inhalt angezeigt. Der Dashboardslider (7.2.3.6 Das Dashboard-Konzept) ist rechts davon verfügbar.

7.2.3.1 Das Konzept zur Anzeige der Inhalte

Die eigentlichen Inhalte der Styleguides, bzw. die Topics, Artikel und Elemente, werden im **Content-Bereich** dargestellt. Die Darstellung der Inhalte folgt Mustern, wie sie von ähnlichen Dokumenten her (z.B. Word-Dokumenten) bekannt und gewohnt sind. Aufgrund des Non-Functional-Requirement [NFR-3], sind die Inhalte hierarchisch gegliedert und ermöglichen somit den Aufbau einer klaren Struktur. Dies hilft dem Benutzer Informationen effizient zu finden.

Um den Inhalt eines Styleguides anzuzeigen, klickt der Benutzer im Inhaltsverzeichnis auf ein entsprechendes Topic. Das Verzeichnis erweitert sich und es werden, falls vorhanden, weitere Subtopics oder direkt der Titel des Artikels angezeigt. Gleichzeitig werden im **Content-Bereich** die Inhalte des gewählten Topics mit allen Subtopics und Artikeln dargestellt.

Enthält ein Artikel ein Bild, wird es als *Thumbnail* innerhalb des Artikels dargestellt. Wenn darauf geklickt wird, öffnet sich die Grossansicht des Bildes in einer *Lightbox* (7.2.3.2 Das Lightbox-Konzept).

7.2.3.2 Das Lightbox-Konzept

Um eine einfache Übersicht der Informationen eines Styleguides zu wahren, werden erweiterte Informationen nicht unmittelbar im **Content-Bereich**, sondern mittels des Konzepts „Lightbox“ dargestellt. In Anlehnung an die *Lightbox JS* von Dhakar (2007), handelt es sich dabei um eine Art Pop-Up-Window, in dem Komponenten (Bilder, Hypertext, etc.) vor einem abgedunkelten Hintergrund dargestellt werden können.

7.2.3.3 Das Konzept zur Anzeige der Metadaten

Für jeden Artikel und für jedes Element können die Metadaten separat dargestellt werden. Dies soll dem Benutzer einerseits eine Übersicht bieten und andererseits helfen nachzuvollziehen, warum und in welchem Umfang Änderungen vorgenommen wurden.

Damit die Metadaten nicht mit den eigentlichen Informationen des Styleguides konkurrieren, sind sie nur über einen gesonderten Link abrufbar. Bei den Artikeln ist der Aufruf für die Metadaten jeweils rechts neben deren Titel in Form eines „Info“-Links platziert. Die Metadaten eines Artikels oder Elementes werden dann in einer Grossansicht dargestellt. Grundlage für dieses Konzept bildet das Feature [FEAT-VIS-2] und das Functional-Requirement [FR-1].

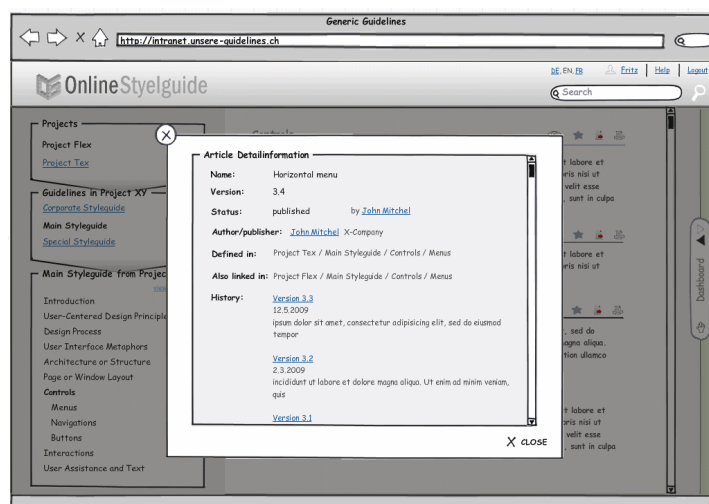


Abb. 41: Metadaten in einer Lightbox

7.2.3.4 Das Index-Konzept

Der Index bietet eine alphabetische Auflistung aller innerhalb eines Styleguides definierten Stichwörter (analog zu einem Stichwortverzeichnis in einem Sachbuch). Der Zweck des Index besteht darin, dem Benutzer die Inhalte eines Styleguides schnell zu erschliessen. Im Prototyp wird der Index wahlweise anstelle des Inhaltsverzeichnisses angezeigt. Grundlage für dieses Konzept bildet das Feature [FEAT-VIS-4] und das Functional-Requirement [FR-4].

Die Ansicht der Suchresultate

Wird eine Suchanfrage (7.2.2.1 *Das Konzept der Suche*) ausgelöst, werden die Ergebnisse der Suche in der Suchresultateansicht dargestellt. Die Anzeige der Suchresultate folgt dabei den Darstellungsformen wie sie von anderen populären Suchmaschinen bekannt sind.

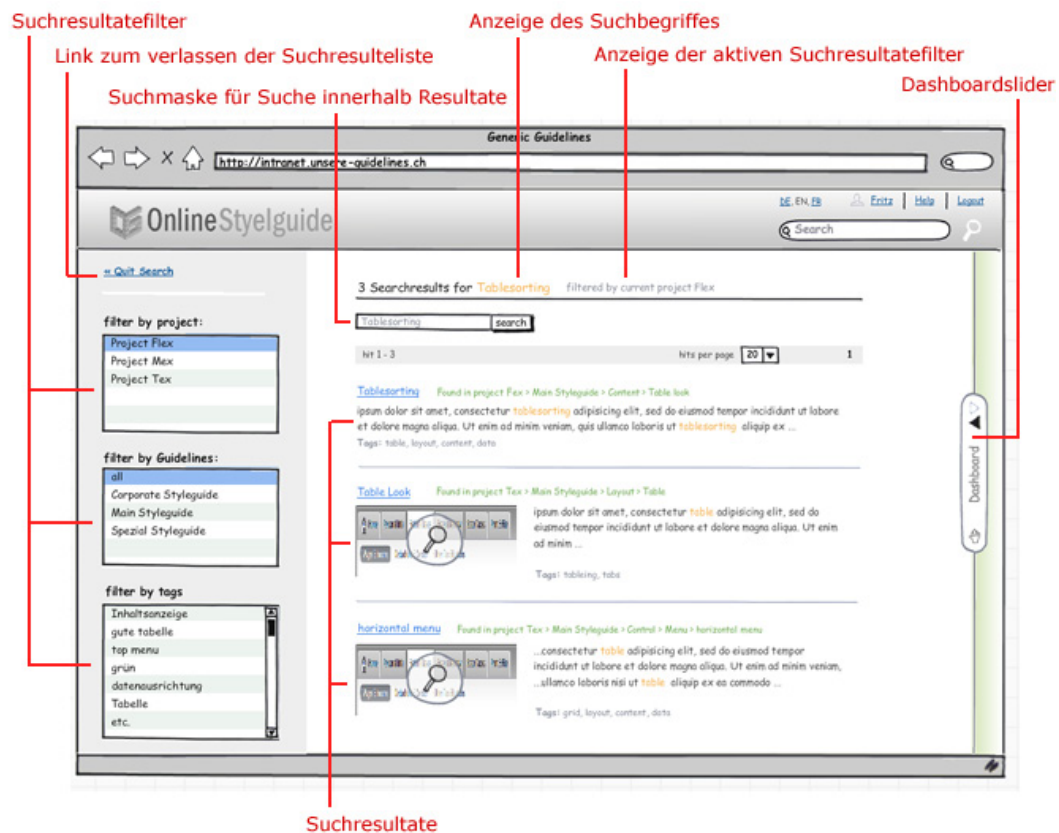


Abb. 42: Liste der Suchresultate

Die Suchresultateansicht ist, analog zur Styleguideansicht (7.2.3 *Ansichten Styleguide, Suchresultate und Dashboard*), in zwei Bereiche, dem **Control**- und dem **Content**-Bereich, aufgeteilt. Im Bereich **Control** werden die Suchresultatfilter (7.2.3.5 *Das Filter-Konzept*) dargestellt und im **Content**-Bereich werden die eigentlichen Suchresultate aufgelistet.

7.2.3.5 Das Filter-Konzept

Grundlage für dieses Konzept bildet das Feature [FEAT-VIS-6]. Eine Suche kann potentiell sehr viele Resultate liefern. Um die Anzahl dieser Resultate weiter einzugrenzen, können drei verschiedene Filter darauf angewendet werden. Der **Projektfilter** (Abb. 43) ermöglicht die Filtrierung der Suchresultate von allen oder bestimmten Projekten. Um Suchresultate nach bestimmten Styleguides zu filtern, wird der **Styleguidefilter** (Abb. 44) verwendet. Weiter lassen sich die Resultate nach Tags filtern (Abb. 45).

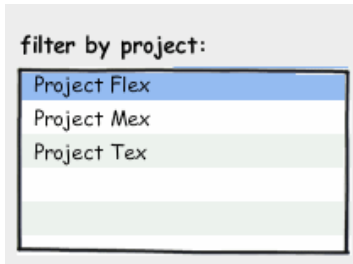


Abb. 43: Projektfilter

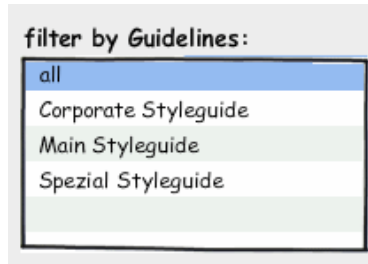


Abb. 44: Styleguidefilter



Abb. 45: Tagfilter

Die Dashboardansicht

Das Dashboard wird im **Content**-Bereich dargestellt und besteht aus einer Ebene, auf der Portlets (7.2.3.7 *Das Portlet-Konzept*) angezeigt werden. Jedes Portlet steht dabei für einen spezifischen Bereich. Die Portlets lassen sich frei anordnen und können für jeden Benutzer nach seinen Bedürfnissen konfiguriert werden. Das Dashboard ist als *Slider* konzipiert, das heisst, es wird über die jeweilige Inhaltsseite eingeblendet, bzw. eingeschoben. Dies ermöglicht es, Informationen auf dem Dashboard einzusehen, ohne dabei die aktuelle Informationsseite verlassen zu müssen.

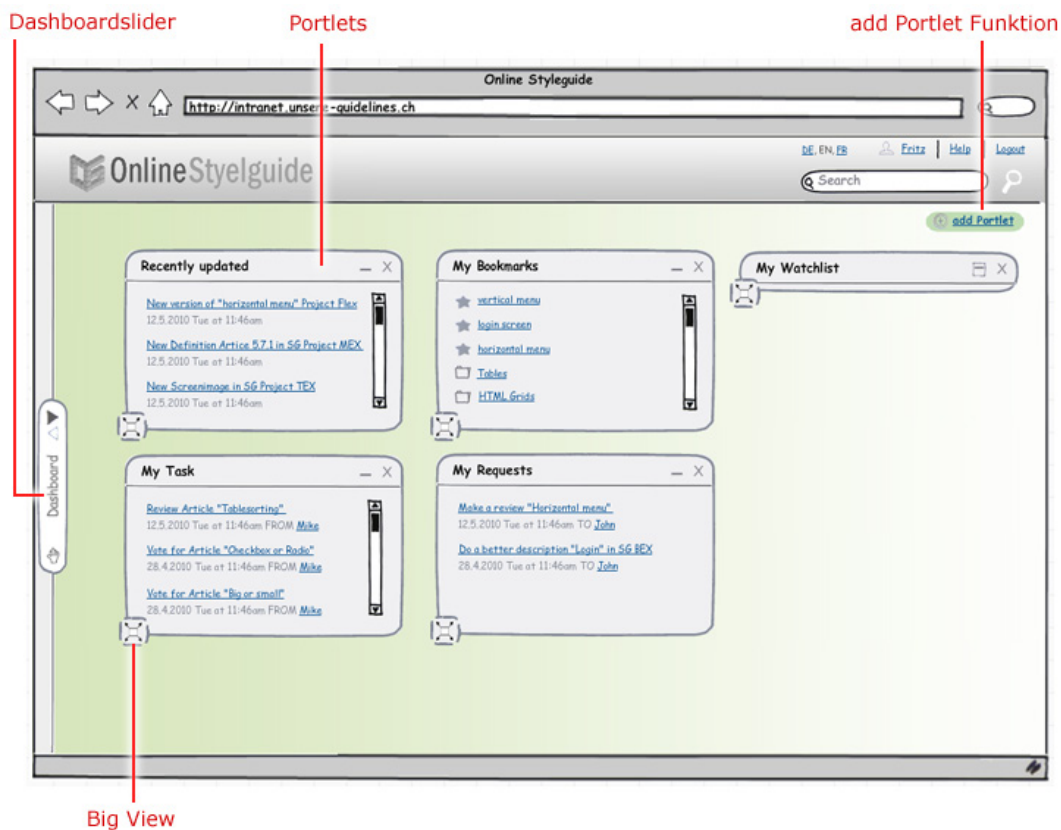


Abb. 46: Dashboardansicht

7.2.3.6 Das Dashboard-Konzept

Grundlage für das Dashboard bilden die Features [FEAT-VIS-3] und [FEAT-VIS-4]. Das Dashboard verfolgt zwei Ziele:

1. Alle bereichsspezifischen Aktivitäten rund um die Styleguides (z.B. Inhaltsaktualisierung und Aufträge) auf einer zentralen Seite im Überblick anzubieten.
2. Dem Benutzer mittels Portlets einen schnellen und bedürfnisgerechten Zugriff auf für ihn relevante Daten im System zu bieten.

Da nicht alle Benutzer die gleichen Ansprüche haben, lässt sich das Dashboard nach deren Nutzen und Bedürfnissen individuell konfigurieren.

7.2.3.7 Das Portlet-Konzept

Die im Dashboard angezeigten Informationen werden, in thematisch gruppierten Portlets angezeigt. Unter Portlets versteht man kombinierbare Komponenten einer Portalseite. Portlets dienen einerseits dazu, Informationen anzuzeigen und andererseits dem Benutzer verschiedene Einstiegspunkte auf den Inhalt von Styleguides zu geben, je nachdem welcher Kontext (tasks, messages, changes, bookmarks, etc.) ihn interessiert.

Da die Personas *Anna Lyst* und *Dave Loper* (4.2.1.2 *Personas in Kürze*) unterschiedliche Anforderungen und Aufgaben haben, sind sie auf unterschiedliche Zugriffsmöglichkeiten angewiesen. Beispielsweise kann *Anna Lyst* Aufträge über das System vergeben, während *Dave Loper* hauptsächlich Informationen im Styleguide sucht. Aus diesem Grund bietet das Dashboard eine feste Anzahl von vordefinierten Portlets, die jeweils spezifische Bereiche abdecken.

Für *Anna Lyst* und *Dave Loper* werden im Prototyp die folgenden Portlets abgebildet:

- My Bookmarks
- My Watchlist
- Recently updated
- My Tasks
- My Requests

Die folgende Tabelle zeigt, welche Portlets von welchen Personas aufgrund ihrer Tasks (*Anhang B: Personas*) verwendet werden:

Persona	My Bookmarks	My Tasks	My Watchlist	My Requests	Recently updated
Anna Lyst	x	x	x	x	x
Dave Loper	x	x	x		

Tabelle 24: Verschiedene Portlets für verschiedene Rollen

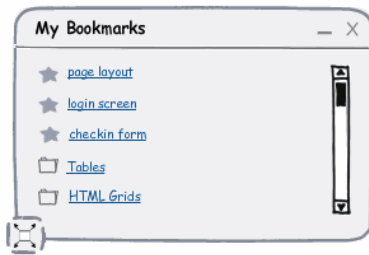


Abb. 47: Portlet „My Bookmarks“

My Bookmarks

Das Portlet enthält Links, die als Bookmarks hinzugefügt werden. Der Benutzer kann Bookmarks zu Inhalten (Topics und Artikel) anlegen. Dies ermöglicht individuelle Einstiegspunkte zu erstellen. Auf häufig genutzte Informationen kann so rasch zugegriffen werden. Grundlage ist das Feature [FEAT-VIS-4] und das Functional-Requirement [FR-8].



Abb. 48: Portlet „My Watchlist“

My Watchlist

Hier werden alle Links dargestellt, die auf einen zu beobachtenden Artikel verweisen. Die Watchlist ermöglicht dem Benutzer, eine Übersicht über Änderungen von Artikeln, die für ihn wichtig sind, zu erstellen. Grundlage sind die Features [FEAT-BEN-1] und [FEAT-VIS-3] und das Functional-Requirement [FR-14].

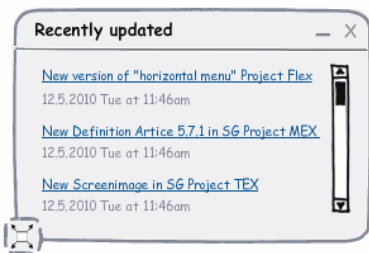


Abb. 49: Portlet „Recently updated“

Recently updated

Hier werden automatisch alle Artikel aufgelistet, in denen eine Änderung stattgefunden hat, z.B. in Form eines hinzugefügten Kommentares. Somit wird gewährleistet, dass der Benutzer eine Übersicht über alle Änderungsaktivitäten geboten bekommt. Grundlage ist das Feature [FEAT-VIS-3] und das Functional-Requirement [FR-14].

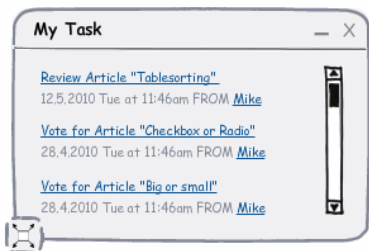


Abb. 50: Portlet „My Tasks“

My Tasks

Der Benutzer hat somit eine zentrale Übersicht über Aufgaben, die ihm zugeteilt sind. Grundlage sind die Features [FEAT-BEN-2] und [FEAT-VIS-3].

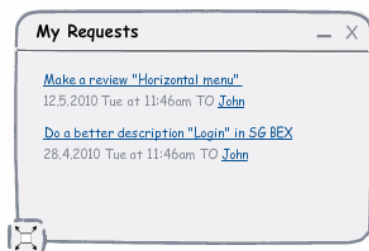


Abb. 51: Portlet „My Requests“

My Requests

Der Benutzer hat somit eine zentrale Übersicht über Aufgaben, die er anderen Benutzer zugeteilt hat. Grundlage ist das Feature [FEAT-KOL-2].

Gemeinsamkeiten der Portlets

1. Um dem Benutzer die Möglichkeit zu bieten, die Portlets nach seinen Bedürfnissen anzuordnen, wird eine freie Platzierung der Portlets ermöglicht. Dies hilft dem Benutzer häufig genutzte Portlets an einer bestimmten Stelle anzuordnen.

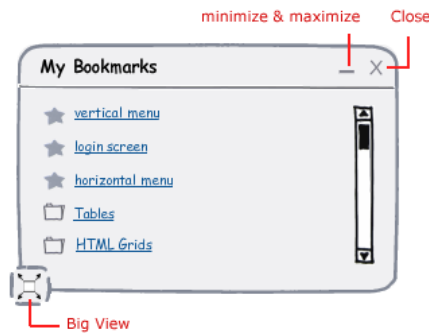


Abb. 52: Portlet-Funktionen „close“, „minimize & maximize“ und „Lightbox“

2. Portlets lassen sich über die Funktion „Close“ schliessen und vom Dashboard entfernen. Eine „add Portlet“-Funktion auf dem Dashboard bietet dem Benutzer Zugang zum Administrationsbereich des Dashboards, in dem entfernte Portlets wieder für die Ansicht im Dashboardbereich aktiviert werden können.

3. Portlets bieten eine „minimize & maximize“-Funktion, um deren Inhaltsbereiche bis auf den Titelfeld zu reduzieren oder wieder zu erweitern. Daraus ergibt sich der Nutzen, dass Inhalte von Portlets nur auf Wunsch angezeigt werden können.

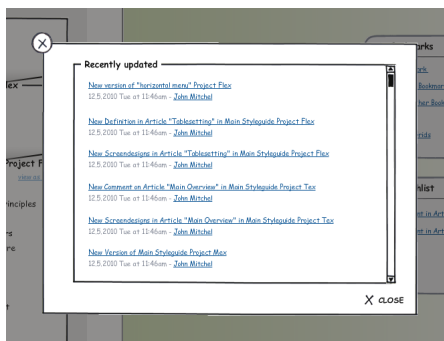


Abb. 53: Lightbox Ansicht eines Portlets

4. Die Portlets bieten eine „Big view“-Funktion. Damit lässt sich der Inhalt eines Portlets in einer Lightbox öffnen (7.2.3.2 *Das Lightbox-Konzept*) und bietet dem Benutzer eine vergrößerte Ansicht auf den Inhalt des jeweiligen Portlets.

7.2.4 Gestaltungskonzept

Der Scope dieser Masterarbeit liegt in der Erstellung eines konzeptionellen Prototyps. Davon ausgehend haben wir auf die Ausarbeitung von Konzepten für Farben, Icons und Typografie verzichtet. Diese würden erst in der späteren Designphase erstellt werden. Die im konzeptionellen Prototyp dargestellten Symbole und Icons haben nur einen sinngemässen Charakter und dienen dem besseren Verständnis des konzeptionellen Prototyps.

7.2.5 Verwendete Patterns

Um für die verschiedenen Designprobleme eine alltagstaugliche Lösung zu finden, haben wir problemangemessene Patterns verwendet.

Folgende Tabelle zeigt die Patterns, die im Prototyp angewendet werden:

Pattern	Systemansicht	Quelle
Dashboard	Wird für die Basis der Dashboardansicht angewendet	(Lammi, 2010a)
Portlet	Wird in der Dashboardansicht angewendet.	(Ajaxpatterns, 2010)
Autocomplete Search	Wird in der Suchmaske des Headers angewendet.	(Toxboe, 2010b)
Indexing	Wird in der Styleguideansicht angewendet.	(Infragistics, 2010)
Split Navigation	Wird in der Styleguideansicht angewendet.	(van Welie, 2008b)
Article list	Wird in der Styleguideansicht angewendet.	(Toxboe, 2010a)
Tags	Wird in der Styleguideansicht angewendet.	(Lammi, 2010d)
Refining Search	Wird für die Filterung der Suchresultate angewendet.	(Lammi, 2010c)
Paging	Wird in der Liste der Suchresultate angewendet.	(Toxboe, 2010c)
Accordion	Wird für die Darstellung der erweiterten Inhalte eines Artikels angewendet.	(van Welie, 2008a)
Two-Panel Selector	Wird als Basis der Styleguideansicht und der Suchresultateansicht angewendet.	(Tidwell, o. J. b)
Center Stage	Wird als Basis für die Darstellung der Inhalte der Styleguides und der Suchresultate angewendet	(Tidwell, o.J. a)
Lightbox	Wird für die Darstellung der erweiterten Inhalte in der Dashboardansicht, Styleguideansicht und der Suchresultateansicht angewendet	(Lammi, 2010b)

Tabelle 25: Verwendete Patterns

8 Der Prototyp im Test – Evaluation

In diesem Kapitel wird erläutert, wie der konzeptionelle Prototyp hinsichtlich seiner Gebrauchstauglichkeit aus Sicht der Personas *Anna Lyst* und *Dave Loper* (*Anhang B: Personas*) für die Use-Cases *Visualize Content* und *Search Content* (6.2.3 *Use-Case Model*) evaluiert wird. Die Analyse des Prototyps hinsichtlich der Anforderungen der beiden Personas wird anhand eines Usability Walkthroughs (3.1.11 *Evaluation des Konzeptes – Usability Testing*) durchgeführt. Der erste Teil dieses Kapitels geht zuerst auf die Methode ein. Im zweiten Teil wird die Fragestellung für die Evaluation beschrieben und welche Test-Szenarios dafür definiert werden. Der dritte Teil widmet sich dem eigentlichen Testablauf und im vierten Teil werden die Findings aus der Evaluation aufgelistet.

8.1 Methode

Ein Usability-Walkthrough ist eine Methode, die zum Ziel hat, mit relativ geringem Zeit- und Kostenaufwand, eine grösstmögliche Anzahl Usability-Probleme zu identifizieren. Ein Usability-Walkthrough durchläuft zwei Phasen. In der ersten Phase wird mittels heuristischer Evaluation versucht, möglichst viele potenzielle Usability-Probleme zu erkennen. In der zweiten Phase wird ein Usability-Test durchgeführt. Dazu handeln die Testpersonen eine Liste von Usability-Szenarios ab.

8.2 Umsetzung

Vorgehen heuristische Evaluation

Für die heuristische Evaluation haben wir den Prototyp im Projektteam diskutiert und überprüft ob die UI-Elemente entsprechend ihrer Benutzbarkeit logisch platziert sind. Anschliessend haben wir geprüft, ob die definierten Anforderungen (*Anhang D: Features und Requirements*) für Use-Cases *Visualize Content* und *Search Content* vollumfänglich abgebildet sind. Wir haben die Resultate der heuristischen Evaluation festgehalten und in eine weitere Iteration des Prototyps einfliessen lassen.

Vorgehen Usability-Test

Wir haben insgesamt fünf Personen der primären und sekundären Personagruppen *Anna Lyst* und *Dave Loper* rekrutiert. Diese Anzahl an Testpersonen erachten wir für einen ersten Usability Test Durchlauf als genügend, da laut Nielsen (2000) mit fünf Benutzern 85% der Usability-Probleme gefunden werden.

Usability-Szenarios

Auf Grund der unterschiedlichen Use-Cases, Features und Requirements von *Anna Lyst* und *Dave Loper* haben wir zwei Sets von Usability-Szenarios zusammengestellt, die wir in der zweiten Phase des Usability-Walkthroughs angewendet haben.

Folgende Fragestellungen haben uns als Grundlage für die Usability-Szenarios gedient:

- Werden gesuchte Informationen schnell gefunden?
- Sind die Links und Icons erwartungskonform und semantisch korrekt interpretiert?
- Wie werden die unterschiedlichen Navigationselemente verstanden? Sind diese intuitiv bedienbar?
- Wird die Struktur des Prototyps richtig verstanden?
- Wird das Dashboard als Übersichtsseite verstanden? Wird auf einen Blick deutlich, welche Inhalte und Portlets zur Verfügung stehen?
- Sind die Interaktionssequenzen für die Use-Cases *Visualize Content* und *Search Content* logisch?

Für beide Personas haben wir je ein Dokument mit einem Set von fünf Usability-Szenarios erstellt, mit denen die zentralen Tasks (Use-Cases) *Visualize Content* und *Search Content* behandelt werden. Damit den Testpersonen die konkrete Ausgangslage verständlich ist, haben wir die Dokumente mit einer einleitenden Erklärung ergänzt. Da ein Teil der Testpersonen englischsprachig war, haben wir die Dokumente auf Englisch abgefasst. Die Usability-Szenarios befinden sich im Anhang (*Anhang K: Walkthrough-Szenarien*).

8.3 Findings

Aus dem Usability-Test lassen sich nachfolgende Findings tabellarisch zusammenfassen. Zur besseren Übersicht werden Findings gruppiert nach *Präsentation/Struktur der Daten*, *Interaktion*, *Allgemein* und *Positive und negative Kriterien*. Die Gewichtung der beobachteten Ergebnisse werden hierbei nach den Kriterien **gering**, **erheblich** und **kritisch** vergeben, wobei **gering** für ein geringfügiges und **kritisch** für ein schwerwiegendes Usability-Problem definiert wird.

Präsentation/Struktur der Daten

Dashboard	Testp.	Gewicht.	Optimierungsvorschlag
Ist zu wenig stark vom Rest der Applikation abgegrenzt.	TP1, TP3	erheblich	Dashboard immer in einer Vollansicht darstellen.
Slider wurde nicht als solcher erkannt.	TP3	gering	Die Sliderfunktion ist zu überdenken und ggf. das Dashboard als eigenständige Webseite implementieren.
Das Label „Dashboard“ ist vertikal schlecht lesbar.	TP4	gering	Dashboard-Button horizontal anordnen und eventuell als Tab-Menü anbieten.
„Big-View“-Icon auf den Portlets wird nicht erkannt.	TP1-TP5	erheblich	Fullscreen Modus überdenken und ggf. weglassen.
Managementfunktion der Portlets wird direkt im jeweiligen Portlet rechts oben erwartet.	TP2	erheblich	Managementfunktion im Portlet integrieren.

Dashboard	Testp.	Gewicht.	Optimierungsvorschlag
Scrollfunktion in einem Portlet ist nicht erwünscht.	TP3	gering	Kann momentan vernachlässigt werden, da die Menge des Inhaltes vorgibt, ob gescrollt werden muss oder nicht.
Ein Portlet für „My Tags“ wird erwünscht.	TP2	erheblich	Ein Portlet „My Tags“ zur Verfügung stellen.
Das Konzept des Portlets „My Watchlist“ ist nicht verständlich.	TP1, TP3, TP4	erheblich	Die Vorteile der beiden Konzepte „My Watchlist“ und „My Bookmarks“ könnten eventuell zusammengeführt werden.
Eine Mailnotifikation über einen Änderungsstatus anstelle von „My Watchlist“ ist erwünscht.	TP3	gering	Das Konzept „My Watchlist“ sollte auch eine Mailnotifikation ermöglichen.
Die Anzeige der „Kommentare“ als Änderung ist im Portlet „Recently updated“ nicht erwünscht.	TP3	gering	Über die Managementfunktion des Portlets soll konfiguriert werden können, welche Änderungen angezeigt.

Tabelle 26: Ergebnisse zum Dashboard

Verzeichnisstruktur	Testp.	Gewicht.	Optimierungsvorschlag
Index wurde nicht benutzt.	TP1-TP5	erheblich	Der Index ist für die Applikation evtl. nicht relevant und könnte eliminiert werden.
Die Struktur wird nicht als selbsterklärend empfunden. Man weiss nicht, welche Kategorie welche Inhalte enthalten.	TP3	kritisch	Design Vorschlag: Verzeichnisstruktur in einer ganzen Ansicht darstellen, z.B. als Topicblock (Doormat Navigation).

Tabelle 27: Ergebnisse zur Verzeichnisstruktur in der linken Spalte

Artikel	Testp.	Gewicht.	Optimierungsvorschlag
Das „Info“-Icon wurde nicht als solches erkannt. Es wird nicht damit assoziiert, dass noch Metainformationen zum Artikel „dahinter verborgen“ sind.	TP2-TP5	erheblich	Besser wäre evtl. eine Integration der Metadaten am Artikelende. Allenfalls ein aussagekräftiges Icon definieren und auf der rechten Seite bei den anderen Artikel-Icons platzieren.
Comments			
Eine Mailnotifikation wäre erwünscht, wenn jemand auf einen Kommentar geantwortet hat.	TP3	erheblich	Ev. eine Mailnotifikation anbieten, wenn jemand auf einen Kommentar antwortet.
Related Article			
Link zu „Related Article“ wurde nicht als solches erkannt, Testperson wusste nicht, dass man das anklicken kann.	TP1	erheblich	Die Link-Auszeichnung muss visuell besser hervorgehoben werden.
Source Code			
Codeangabe sollte als Referenz deklariert sein. Nur „Klassen“-Angabe und ev. eine Verlinkung auf eine Code-Bibliothek anbieten.	TP3	erheblich	Man sollte als Empfehlung im Styleguide angeben, wie der Code benutzt werden kann/soll.
Es wurde nicht erkannt, dass der Menulink „Source Code“ zu dem Artikel- Source-Code verweist.	TP5	erheblich	„Source Code“ sollte oberhalb von „Examples“ dargestellt werden, so könnte der Eindruck vermieden werden, der „Source Code“ gehöre zu den „Examples“.

Artikel	Testp.	Gewicht.	Optimierungsvorschlag
Examples			
Ansichten von verschiedenen Versionen inkl. Beispielcode werden gewünscht.	TP2, TP4	gering	Bilder von verschiedenen Versionen inkl. Beispielcode anbieten.
Tags			
Möchte nicht selber Tags vergeben.	TP2, TP4	erheblich	Styleguide-Autoren können Tags vergeben
Tags werden nicht als Links erkannt.	TP2	gering	Die Link-Auszeichnung der Tags visuell besser hervorheben

Tabelle 28: Ergebnisse zu einem Artikel

Interaktion

Suchfunktion	Testp.	Gewicht.	Optimierungsvorschlag
Es ist nicht klar ob sich die Suche auf ein oder alle Projekte bezieht.	TP1,TP2	erheblich	In den Suchresultaten ein Hinweis platzieren, was und wo gesucht wurde.
Der Inhalt im Suchvorschlag ist nicht klar. Sind es nur Tags, ein Artikel oder sämtliche Artikel zum Suchbegriff?	TP1,TP2	erheblich	Anmerkung: Speziell Personen, die häufig Tags benützen, erwarten eine Suche die sich auf Tags bezieht.
Eine „Advanced Search“ wurde gewünscht.	TP2,TP4	erheblich	Die Suchmaske mit Möglichkeit für eine „Advanced Search“ erweitern.
Die Filterung der Suchresultate (linke Spalte) wurde nie benutzt.	TP1-TP5	erheblich	Ev. ein Problem der Usability-Szenarios, ev. kann die Filterung als „Advanced Search“ implementiert werden.
Suchkriterien nach „Bilder“, „Code“ etc. wären hilfreich.	TP3	gering	Suchkriterien könnte mit der „Advanced Search“ umgesetzt werden.

Tabelle 29: Ergebnisse zur Suchfunktion

Allgemein

Allgemeine Ergebnisse	Testp.	Gewicht.	Optimierungsvorschlag
Home-Funktion wird vermisst!	TP1,TP3, TP4	erheblich	Einen Home-Link anbieten, um z.B. zum Dashboard zu gelangen.
Der Benutzer weiss nicht wo er sich befindet und vermisst eine Übersichts-funktion.	TP2,TP3, TP4	erheblich	Eine Breadcrumb-Funktion für die allgemeine Übersicht implementieren.

Tabelle 30: Allgemeine Ergebnisse

Positive und negative Kriterien

Positive und negative Kriterien zum System	Testp.	Kriterium
Das Dashboard wird von allen Benutzern geschätzt und als hilfreich erachtet.	TP1-TP5	positiv
„Related Content“-Funktion wird geschätzt.	TP2-TP4	positiv
Das System wird als intuitiv bedienbar empfunden.	TP3,TP5	positiv
Benutzer sind motiviert das System einsetzen	TP2,TP5	positiv
Dashboardslider-Funktion wird nicht als praktikabel empfunden.	TP1-TP3,	negativ
Eingesetzte Icons sind nicht immer gut erkennbar.	TP4,TP5	negativ

Tabelle 31: Positive und negative Kriterien

9 Der Einsatz projektspezifischer Styleguides – Guidelines

Dieses Kapitel befasst sich mit unseren Empfehlungen bezüglich des Einsatzes von Styleguides in Unternehmen für Projekte. Sie sind eine Zusammenfassung der, unserer Meinung nach, wichtigsten Erkenntnisse aus der Literatur und der Praxis. Die Empfehlungen sind dabei von genereller Natur und nicht auf ein bestimmtes technisches System oder Konzept bezogen.

Guidelines zum Prozess

Guideline	Beschreibung
Involviere das Management	Die Sicherstellung der Usability eines Produktes ist eine Angelegenheit von höchster Stelle. Das Management sollte Vorgaben erlassen, welche die Einhaltung Konsistenz fordert. Allerdings muss klar gemacht werden, dass Styleguides nicht die Lösung aller Usability-Probleme sind, sondern nur eines der Mittel dafür ist.
Kenne das Umfeld, in dem der Styleguide eingesetzt werden soll	Es muss klar sein, für welche Personen und für welches Produkt ein Styleguide eingesetzt werden soll. Der Entwicklungsprozess, in dem der Styleguide angewendet wird, ist ebenfalls zentral.
Definiere ein klares Ziel, den Anwendungsbereich und die Verantwortlichkeiten	Es muss klar sein, wofür ein Styleguide erstellt werden soll, was dessen Ziel ist und wofür er nicht gedacht ist. Die Verantwortlichkeit für einen Styleguide muss schon früh festgelegt werden. Es muss klar sein, wer die Ansprechpartner sind und wer Änderungen durchführt.
Starte früh	Es sollte früh damit begonnen werden einen Styleguide zu erstellen, damit er vor der Implementation eines User-Interfaces angewendet werden kann. Ein guter Zeitpunkt wäre nach der Analyse der Anforderungen an ein User-Interface.
Involviere die Benutzer und plane grosszügig	Ein Styleguide sollte in Zusammenarbeit mit dessen Benutzern erstellt werden. Nur so ist sichergestellt, dass er akzeptiert wird, benutzbar ist und die relevanten Informationen enthält. Die Benutzer sollten ermuntert werden, aktiv am Erstellungsprozess teilzunehmen. Die Zeit, die es für die Bildung eines Konsens braucht, sollte aber nicht unterschätzt werden.
Kommuniziere Änderungen	Bei Änderungen an einem Styleguide soll dafür gesorgt werden, dass die betroffenen Personen darüber informiert werden. Nicht nur nachdem Änderungen stattgefunden haben, sondern auch bevor .

Tabelle 32: Guidelines zum Prozess

Guidelines zur Struktur und Form

Guideline	Beschreibung
Stelle den Styleguide online zur Verfügung	Damit Benutzer jederzeit und von verschiedenen Orten Zugriff auf den Styleguide haben, sollte er online zur Verfügung gestellt werden. Dadurch lassen sich auch die Vorteile von Computern ausnutzen. Ein einfacher Styleguide könnte beispielsweise auf Basis eines Wiki (z.B. CONFLUENCE) sein.
Erarbeite die Struktur in Zusammenarbeit mit den Benutzern	Damit ein Styleguide eine benutzbare Struktur und die relevanten Informationen aufweist, sollte sie in Zusammenarbeit mit den Benutzern erstellt werden.
Biete verschiedene Möglichkeiten um Informationen zu finden und darauf zuzugreifen	Ein Styleguide sollte hauptsächlich als Nachschlagewerk angesehen werden. Er sollte daher verschiedene Möglichkeiten zum Auffinden von Informationen bieten. Dies kann beispielsweise in Form eines Inhaltsverzeichnis und Index geschehen. Auch der Einsatz von Tags ist sinnvoll. Informationen, die in Bezug zueinander stehen, sollten miteinander „verlinkt“ werden.

Tabelle 33: Guidelines zur Struktur und Form

Guidelines zum Inhalt

Empfehlungen zum Inhalt von Styleguides beziehen sich **nicht** auf konkrete Themenbereiche, die in einem Styleguide enthalten sein sollten. Auf die möglichen Themenbereiche wurde in Kapitel 2.4.4 *Inhalt von Styleguides* eingegangen. An dieser Stelle sprechen wir allgemeine Empfehlungen zum Umgang mit Inhalten in Styleguides aus.

Guideline	Beschreibung
Hilf dem Benutzer gute Entscheidungen zu fällen! Beschreibe den Nutzungskontext und liefere Begründungen	Der Nutzungskontext eines Designelementes (z.B. Tabellen, Listen, Fenster, etc.) sollte beschrieben sein. Es sollte beschrieben sein, wann ein bestimmtes Element eingesetzt werden soll und wann nicht. Dies kann beispielsweise anhand von Vergleichstabellen (Vor- und Nachteile von Tabellen, Radio-Buttons, etc.) geschehen. Dasselbe gilt auch für Interaktionsstile (z.B. Eine Abfolge von Interaktionen). Damit der Sinn eines Designelementes klar wird, sollte begründet werden inwiefern es die Usability erhöht.
Integriere Ergebnisse aus Benutzeranalysen und Usability-Tests	Falls Ergebnisse von Benutzeranalysen oder Usability-Tests für eine bestimmte Guideline vorliegen, sollten sie im Styleguide integriert werden. Dies erhöht deren Glaubwürdigkeit und ausserdem wird dadurch besser ersichtlich, wann eine bestimmte Guideline evtl. geändert werden müsste, falls sich die Benutzergruppe verändert.
Beschreibe bekannte und gängige Designelemente nicht im Detail	Bekannte, bzw. gängige Designelemente (z.B. Slider, Buttons, etc.) sollten nicht im Detail beschrieben werden. Es sollte dafür eher deren Nutzungskontext beschrieben und in welchen Kombinationen mit anderen Komponenten sie verwendet werden können oder sollten.
Mache klar, welche Guidelines befolgt werden müssen und welche nur eine Empfehlung sind	Es sollte aus dem Styleguide klar hervorgehen, welche der darin enthaltenen Guidelines eingehalten werden müssen und welche nicht, bzw. nur eine Empfehlung darstellen.
Integriere viele Beispiele	Es sollten viele Beispiele, insbesondere auch „Negativbeispiele“ erstellt und integriert werden. Dies erleichtert es eine Guideline und auch dessen Nutzungskontext besser zu verstehen.
Biete verschiedene Arten von Inhalten	Damit Informationen eines Styleguide möglichst effektiv kommuniziert werden können, sollte er verschiedene Arten von Inhalten bieten. Idealerweise sollte ein Styleguide nicht nur Bilder und Text, sondern auch Code (z.B. CSS-Templates) oder interaktive Elemente enthalten.
Integriere Checklisten	Ein Styleguide sollte Checklisten enthalten, mit denen die Überprüfung der Konformität des User-Interfaces schnell durchgeführt werden kann.
Integriere Diskussionsthemen und Kommentare	Um die Transparenz und Nachvollziehbarkeit von Entscheidungen zu erhöhen, sollte der Styleguide ebenfalls Diskussionsthemen und Kommentare enthalten.

Tabelle 34: Guidelines zum Inhalt

10 Schlussfolgerungen

Dans ce chapitre nous commençons par passer en revue les difficultés survenues dans ce travail ainsi que ses limites. Ensuite nous discutons des méthodologies employées. Les réflexions sur les résultats et l'utilité du travail sont ensuite exposées, suivie des perspectives ouverte par cette recherche. Nous terminons avec une vue d'ensemble du déroulement du projet ainsi que les conclusions personnelles de chaque membre.

10.1 Schwierigkeiten und Limitationen

La difficulté principale du projet a été de définir un scope réaliste. Il n'a été clair que tardivement de ce qui pouvait être fait dans le cadre du présent travail. Le volume du travail a au début été sous-estimé, d'une part parce que le domaine de recherche est vaste et compliqué. Avec la démocratisation des styleguides, la terminologie et le vocabulaire utilisé sont devenus de moins en moins clairs. Il y a beaucoup d'interprétations et de compréhensions différentes du thème et de l'objet de recherche. Cela a nécessité un plus grand effort pour l'analyse et la synthèse de la littérature pour le cadre de la recherche. D'autre part le projet n'a pas de mandant externe à l'école. Ce qui signifie une très grande liberté de décision et aucun cahier des charges précis à réaliser pour une entreprise. Le champ de recherche était totalement ouvert, d'où la difficulté à définir un scope faisable dans le temps imparti.

Le fait de ne pas avoir de mandant a généré une deuxième difficulté: obtenir rapidement suffisamment d'interviews pour l'analyse des utilisateurs et des personnes tests pour l'évaluation du prototype. La plus part des contacts proviennent de connaissances ou de collègues de travail.

Goodwin (2009 S. 88) recommande quatre interviews par Persona pour une application normale de „consumer". L'application *Online-Styleguide* n'appartient pas à cette catégorie, en raison du nombre de rôles et de métiers différents [PRAX-1] et de la complexité du domaine. Le contexte est également très variable: de multiples types d'applications, de plateformes, de méthodologies [PRAX-3] et de sortes de styleguides [PRAX-15]. Il aurait fallu plusieurs dizaines d'interviews pour réaliser l'analyse.

Avec seize interviews, l'analyse des utilisateurs pose quelques problèmes de représentativité et l'hypothèse [HYP-1] n'est qu'en partie vérifiée. Faute de données suffisantes, la persona *Hans Writer* est toujours provisoire. Les Business Analystes étaient sous représentés lors de l'évaluation du prototype et des interviews. Il demeure encore des questions et points ouverts chez les utilisateurs. De plus seul des utilisateurs d'entreprise ayant mis en place des styleguides ont été interviewés. La vérification des Personas doit être encore réalisée.

L'hypothèse [HYP-3] n'a pas pu être validée ou invalidée en raison du manque de données. Seul une personne a mentionné réutiliser des parties de styleguides. Ce qui n'a rien d'étonnant étant donné que *Hans Writer* écrit le styleguide et sa Persona est encore provisoire. A cause de la restriction du scope, seul le Use-Case Model des utilisateurs a été esquissé. Seul deux System Use-Cases réalisés dans le prototype ont été spécifiés en détail. Le manque de temps en est la cause.

L'objectivité et l'impartialité de l'évaluation des concurrents peut être partiellement remise en cause. Nous avons une opinion négative de QUINCE PRO à cause de son design trop lourd et trop sombre, que nous jugeons inadapté pour une application à utiliser tous les jours. L'outil nous est également inconnu, contrairement à CONFLUENCE que nous connaissons très bien. Ces facteurs sont à prendre en considération dans la lecture des résultats.

10.2 Reflexion der Methodologie

La formulation des hypothèses de travail n'a pas été des plus scientifiques. Nous avons formulé les questions qui nous semblaient pertinentes d'après la littérature. Reprendre des perspectives d'autres travaux aurait été une démarche plus orientée et peut-être plus efficace. Est-ce que les résultats des Personas auraient été les mêmes, si l'hypothèse avait été basée sur les groupes définis dans *Abb. 10: Benutzergruppen von Styleguides (Quesenbery, 2001a)* (utilisateur, créateur, modificateur)?

Nous sommes d'avis que peu de chose aurait changé dans le choix et le déroulement des interviews. Nous n'avions pas un grand éventail de personnes à disposition. Contrairement à l'analyse qui aurait pu être plus aisée et les résultats être différents. Au lieu d'être marginaux dans notre modèle, [U18] et [U28] qui modifient également les styleguides auraient pu être représenté par la Persona „Modificateur“.

Le choix des techniques d'acquisition de données se sont révélés adéquats. Les interviews ethnographiques condensées –où c'était possible– étaient adaptées à notre situation de projet. Les utilisateurs n'avaient pas beaucoup de temps à nous consacrer, des Contextual Inquiry étaient donc impossible. De plus, ils n'auraient probablement pas amené plus d'informations. Une amélioration possible concernerait l'organisation des interviews: plus "agressifs" dès le début, c'est-à-dire démarcher plus largement et relancer systématiquement en cas d'absence de réponse.

Nous avons tenté d'innover en employant des nouvelles techniques non enseignées en cours. Il s'agit du Concept Model de Brown (2006) ainsi que les sessions de Debriefings de Holtzblatt et al. (2005). Ces deux techniques ont été réalisées et maîtrisées avec succès, l'une nous a permis de représenter les concepts du système, l'autre de traiter rapidement les données brutes des interviews pour les réutiliser par la suite.

10.3 Diskussion der Ergebnisse

L'application *Online-Styleguide* ne résout pas tous les problèmes, elle n'a rien d'une solution miraculeuse. Elle ne répond pas aux problèmes de management [LIT-PK-3] ou aux problèmes de structure et de contenu (4.1.3 *Themenbereich: Inhalt*, 4.1.4 *Themenbereich: Struktur und Form*). Dans ce cas, y a-t-il vraiment besoin d'un système de gestion des styleguides? Si oui, peut-il être générique, comme formulé dans [HYP-4] et serait-il bien reçu par les utilisateurs [HYP-2]?

Comme nous l'avons vu précédemment, la réponse à la première question est incontestablement oui. Un média traditionnel n'est plus suffisant dans le contexte actuel de développement d'application. Il ne remplit que faiblement le cahier des charges et les recommandations de la littérature comme le montre l'analyse concurrentielle (5.3 *Konkurrenzanalyse*).

La réponse à la deuxième question est plus nuancée. Comme abordé précédemment, le contexte des entreprises est très varié. Une standardisation des rôles, des dénominations, des tâches, des workflows nous paraît dès lors très complexes. Un système générique demanderait de gros efforts de développement pour être suffisamment flexible à tous les cas de figures. Selon nous, il serait possible de réaliser un système générique, mais il risque de coûter cher. Les interviews et les Walkthroughs ont montré sans équivoque un intérêt pour un outil de gestion des styleguides.

Une question plus intéressante serait: est-il nécessaire de développer un système dédié et générique? L'analyse concurrentielle livre les premiers éléments de réponse. Beaucoup de fonctionnalités néces-

saires sont en partie déjà disponibles dans les outils collaboratifs et les CMS. Nous pensons qu'il vaudrait la peine d'envisager les produits de substitutions tels que CONFLUENCE ou TYPO3 comme plateforme de base à étendre. Il n'existe d'ailleurs pas sur le marché, à notre connaissance, d'outils qui se veulent génériques. Même QUINCE PRO sorti en mars 2010, est spécialisé dans les Patterns.

Selon nous, ce travail est un apport utile pour la recherche, car il fait le point des connaissances actuelles sur le thème des styleguides en général. Grâce au chapitre relatif aux problèmes et solutions de la littérature (*4.1 Recherchen in der Literatur*), une partie importante du scope de ce travail a été réalisé avec succès. Nous n'avons trouvé aucun bilan de ce type dans la littérature spécialisée. Notre recherche défriche un sujet vaste et ouvre la voie à d'autres travaux d'approfondissements, particulièrement concernant le „Dashboard“. La création d'un prototype nous a permis de vérifier que cet élément – le Dashboard – est une solution intéressante dans une application comprenant plusieurs Personas primaires. Il s'adapte bien lorsqu'il y a beaucoup de métiers et d'intérêts différents. Pouvoir personnaliser le point d'entrée de l'outil et les informations à afficher est un atout majeur. Le „Dashboard“ ainsi que les „Portlets“ ont reçu un très bon écho lors des tests. Ce travail est enfin utile pour les entreprises, grâce à la compilation de conseils disponible au chapitre *8.3 Findings*.

10.4 Weitere Schritte

Pour un prochain travail de recherche, nous conseillons d'approfondir les pistes ouvertes dans ce travail et que nous n'avons pu achever. D'abord il faudrait terminer l'analyse des utilisateurs, en interrogeant d'autres rôles supplémentaire (par ex: des testeurs) ainsi que d'autres entreprises (qui n'utilisent pas de styleguides ou dont l'introduction était infructueuse). Ensuite nous recommandons de récolter les préférences des utilisateurs à l'aide d'un questionnaire online. Pour finir nous conseillons une évaluation plus approfondie des produits de substitutions, c'est-à-dire en les installant et en les testant de manière plus systématique. Les solutions internes aux entreprises pourraient aussi être analysées en détail. Il serait enfin judicieux de finaliser la documentation, ainsi que les diagrammes.

10.5 Projektablauf und persönliche Fazits

Le projet s'est déroulé globalement de manière satisfaisante. Le contenu des phases a été modifié (abandon des questionnaires) suite aux retards pris dans les interviews. La phase Elaboration 2 (prototype interactif) a été annulée. Malgré le conseil d'IBM, nos phases RUP se sont légèrement chevauchées, pour permettre de continuer le travail sans devoir attendre sur les résultats finaux de la recherche empirique. Les chiffres romains dans le graphique ci-dessous représentent les itérations et correspondent aux phases RUP réelles. Si les rôles avaient été attribués plus tôt dans le projet (chef de projet, chef de thèse, chef méthodologie/artefacts), nous aurions pu être plus efficaces et productifs.

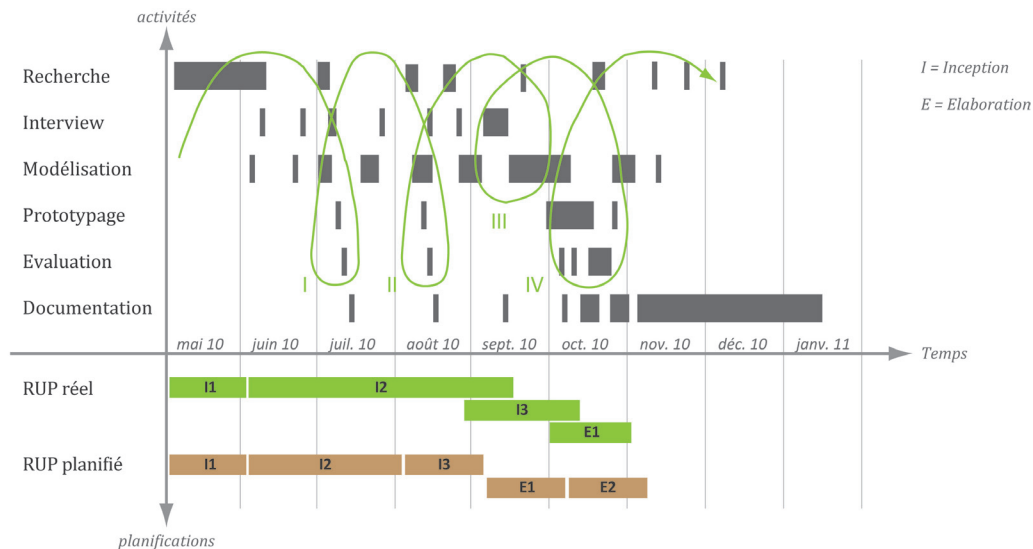


Abb. 54: Déroulement du projet

Stefano Morandi

Als UI-Entwickler wurde ich bislang in zwei verschiedenen Projekten in zwei verschiedenen Unternehmen mit Styleguides konfrontiert – und beide Male sind sie kläglich gescheitert. Wir Entwickler konnten die Styleguides einfach nicht ernst nehmen. Viele der darin enthaltenen Guidelines waren konfus beschrieben und technisch nicht, oder nur mit enormen Aufwand, umsetzbar. Ausserdem waren es ausgesprochen hässliche anzusehende Dokumente... Ich konnte mir einfach nicht erklären, weshalb sich „irgendjemand“ so viel Mühe gemacht hat derart nutzlose Dokumente zu erstellen.

Im Verlaufe dieser Masterarbeit wurde mir aber klar vor Augen geführt wo die Knackpunkte liegen, wie unendlich umfangreich dieses Gebiet ist und welch babylonischer Begriffswirrwarr darin herrscht. Gute, hilfreiche und benutzbare Styleguides sind ist nicht *irgendein* Produkt eines *einzelnen* Geistes. Sie müssen sorgfältig geplant und erstellt werden, denn insbesondere projektspezifische Styleguides bilden einen Knotenpunkt des Informationsaustausches zwischen HCI-Experten, Designern, Entwicklern, Marketing, Firmenleitung und Endanwender einer Applikation.

Die Vielschichtigkeit der in Styleguides enthaltenen Informationen, die Vielzahl unterschiedlicher Prozesse denen sie unterworfen sind, die verschiedenen Benutzer und deren unterschiedliche Ansprüche an den Inhalt, sowie letztendlich auch die Vielfältigkeit der praktischen Verwendung, machten Recherche, Analyse und Konzeption eines „Online-Styleguide“ zu einer echten Herausforderung, die sehr viel Zeit und Energie verschlang.

Schlussendlich bleibt mir zu sagen, dass ich während dieser arbeitsintensiven Zeit einerseits viel über die Thematik von Styleguides gelernt habe und andererseits benutzerzentriertes Vorgehen anwenden, vertiefen und festigen konnte. Ich bedanke mich an dieser Stelle herzlich bei meinen Teamkollegen für eine in thematischer und menschlicher Hinsicht überaus lehrreiche Zusammenarbeit. Dass es uns gelungen ist, wenn auch nicht im ursprünglich „angepeiltem“ Umfang, die Ziele dieser Masterarbeit zu erreichen, lässt mich mit gutem Gewissen auf weniger stressige Zeiten hoffen...

Sébastien Niclasse

L'expérience au sein de ce projet a été pour moi très enrichissante tant au plan humain que professionnel. Je remercie encore tout spécialement mes collègues ainsi que les Coaches de m'avoir permis d'écrire le diplôme dans ma langue maternelle. Ce travail m'a permis entre autres de progresser dans la modélisation des processus et des Use-Cases. On ne peut pas selon moi apprendre dans les livres à faire de bons Use-Cases ou Business Use-Cases. C'est l'entraînement, l'expérience et les discussions avec des pairs expérimentés qui peuvent nous faire progresser. J'ai également beaucoup appris sur la gestion de projet. Je retiens deux choses importantes de cette activité: il faut d'une part être optimiste et convaincu pour pouvoir rassurer et motiver ses troupes, d'autre part il faut avoir une vision claire du travail et des prochaines étapes pour encadrer le projet. Ce travail m'a encore prouvé qu'il est important de pouvoir communiquer et de dire les choses directement, même si parfois ça ne fait pas plaisir. Une fois les esprits calmés et avec un peu de bonne volonté, on trouve toujours des solutions aux problèmes.

Sans avoir révolutionné le domaine, je pense que nous avons apporté notre pierre à l'édifice et que la solution pour beaucoup d'entreprise passera par les produits de substitutions. Je serai particulièrement attentif à CONFLUENCE qui intègre déjà un outil de Mockup (BALSAMIQ) et qui, enrichi avec de nouveaux plugins, pourrait être très approprié pour gérer les styleguides et la collaboration. Je ne connaissais pas bien le sujet avant de commencer ce diplôme, je me réjouis maintenant de pouvoir mettre en pratique les concepts appris dans mon entreprise.

Chris Turtschi

Diese Masterarbeit war für mich eine äusserst interessante Arbeit, gerade weil ich selber in meinem beruflichen Umfeld mit ähnlichen Problemen im Zusammenhang mit Styleguides konfrontiert bin. Das Thema und die Erfahrungen mit den angewandten Vorgehens-Methoden werden für mich sicherlich wegweisend für die nähere Zukunft sein. Als grösste Herausforderung während der Masterarbeit erachte ich sicher das Schreiben der Thesis, bedingt einerseits, durch die fehlende Erfahrung im Umgang mit wissenschaftlichen Arbeiten und andererseits durch die ausgeprägte berufliche Beanspruchung.

Das Arbeiten im Team empfand ich dagegen als grosse Bereicherung. Jeder konnte seine Ansichten und Auffassungen einbringen wodurch immer für anregende Diskussionen gesorgt war. An dieser Stelle besten Dank an meine Teamkollegen für die tolle Zusammenarbeit! Mit dem Konzept und den Empfehlungen als Endresultat, denke ich, haben wir eine gute Ausgangslage geschaffen um interessierten Personen detaillierte Informationen im Umgang mit Styleguides bieten zu können. Mich persönlich motiviert es, das Konzept bei meinem Arbeitgeber entsprechend anzuwenden und umzusetzen. In der Hoffnung, dass die bestehenden Umgangsschwierigkeiten mit Styleguides ausgemerzt werden können.

Anhänge

Die Anhänge sind nicht Teil dieses Berichtes und daher in einem separaten Dokument abgelegt.

Anhang A: Projektplanung

Anhang B: Personas

Anhang C: Business Use-Cases

Anhang D: Features und Requirements

Anhang E: Konkurrenzanalyse

Anhang F: Concept Model

Anhang G: Strukturdiagramm

Anhang H: Zustandsdiagramm

Anhang I: Use-Case Spezifikationen

Anhang J: Screenshots Prototyp

Anhang K: Walkthrough-Szenarien

Anhang L: Walkthrough-Analyse

Anhang M: Hintergrund zu Concept Models

Anhang N: Glossar

Literatur und Quellen

Ajaxpatterns Portlet [Online] // ajaxpatterns.org. - 12. August 2010. - September 2010. - <http://ajaxpatterns.org/Portlet>.

Arnoult Stéphane ANALYSE DES CONCURRENTS [Online] // b2b-marketing.fr. - 2006. - 12. 11 2010. - <http://www.b2b-marketing.fr/1-analyse-des-concurrents.php>.

Atlassian Confluence Everyone on the same page [Online] // <http://www.atlassian.com>. - 2010. - 09. 11 2010. - <http://www.atlassian.com/software/confluence/>.

Bathelot Bertrand Définition Benchmark [Online] // www.definitions-marketing.com. - 02. 07 2010. - 02. 12 2010. - <http://www.definitions-marketing.com/Definition-Benchmark>.

Beck Kent [et al.] Manifesto for Agile Software Development [Online]. - 2001. - 26. 08 2010. - <http://agilemanifesto.org/>.

Beier Betsy und Vaughan Misha M. The Bull's-Eye: A Framework for Web Application User Interface Design Guidelines [Online]. - April 2003. - 14. November 2010. - <http://www.cc.gatech.edu/projects/PageSleuth/references/p489-beier.pdf>.

Bircher Thomas Vorgehensmodelle II, Design [PDF]. - Rapperswil : [s.n.], 09 2009. - MAS HCID 2009.

Bittner Kurt und Spence Ian Use Case Modeling [Buch]. - Amsterdam : Addison-Wesley Longman, 2002.

Brandenburg Hans und Wojtyna Jean-Pierre L'approche processus - Mode d'emploi [Buch]. - [s.l.] : Editions d'Organisation, 2003.

Brown Daniel M. A taste of Concept Models and Flow Charts [Online] // [Communicating Design](http://communicatingdesign.com). - 19. August 2010b. - 26. Oktober 2010. - <http://communicatingdesign.com/2010/08/19/a-taste-of-concept-models-and-flow-charts/>.

Brown Daniel M. Communicating Design [Buch]. - Berkeley : New Riders, 2006.

Chaney R. Garnet Confluence vs. Mediawiki [Online] // www.bobsgear.com. - 27. 03 2007. - 09. 11 2010. - <http://www.bobsgear.com/display/bobsgear/Confluence+vs.+Mediawiki>.

Cockburn Dr. Alistair Using Both Incremental and Iterative Development [Online]. - 2008. - 26. 08 2010.

Cooper Alan, Reimann Robert und Cronin David About Face 3: The Essentials of Interaction Design [Buch]. - [s.l.] : John Wiley & Sons Ltd, 2007. - 3Rev Ed.

Courage Catherine und Baxter Kathy Understanding Your Users - A practical guide to user requirements [Buch]. - San Francisco : Elsevier, 2005.

Dhakar Lokesh Lightbox JS [Online] // lokeshdhakar.com. - 4. Dezember 2007. - 3. November 2010. - <http://www.lokeshdhakar.com/projects/lightbox2/>.

Dix Alan [et al.] Human-Computer Interaction (Third Edition) [Buch]. - Harlow : Pearson, Prentice Hall, 2004a.

Dix Alan [et al.] resources [Online] // Human-Computer Interaction (3rd Edition). - 2004b. - 15. November 2010. - <http://www.hcibook.com/e3-docs/slides/notes-pdf/e3-chap-07-6up.pdf>.

Edmonds Andy Style Guide Primer for Agile Teams [Online] // UX Agile. - 12. 12 2008. - 26. 08 2010. - <http://uxagile.com/2008/12/style-guide-primer-for-agile-teams/>.

Fadeyev Dmitry Quince: A User Interface Patterns Explorer [Online] // www.usabilitypost.com. - 26. 02 2009. - 09. 11 2010. - <http://www.usabilitypost.com/2009/02/26/quince-a-user-interface-patterns-explorer/>.

Faraday P. Visually critiquing web pages [Konferenz] // Proceedings of the 6th Conference on Human Factors and the Web. - Austin : [s.n.], 2000.

Fowler Martin The New Methodology [Online] // martinfowler.com. - 2005. - 26. 08 2010. - <http://martinfowler.com/articles/newMethodology.html>.

Free Host Review TYPO3 CMS Review [Online] // <http://www.freehostreview.com>. - 2010. - 09. 11 2010. - <http://www.freehostreview.com/content/typo3-cms-review.html>.

Gale Stephen A Collaborative Approach to Developing Style Guides [Online] // CHI 96 - Electronic Proceedings. - 1996. - 24. Oktober 2010. - http://www.sigchi.org/chi96/proceedings/papers/Gale/srg_txt.htm.

Goodwin Kim Designing for the Digital Age: How to Create Human-Centered Products and Services [Buch]. - Indianapolis : Wiley Publishing Inc., 2009.

Griffiths Richard N. und Pemberton Lyn Don't Write Guidelines Write Patterns! [Online]. - 2003. - 10. November 2010. - <http://www.it.bton.ac.uk/staff/lp22/guidelinesdraft.html>.

Hart Geoff J. The style guide is "dead": long live the dynamic style guide! [Online] // Geoff-Hart.com (editing, writing and translation). - 12. März 2000. - 30. Oktober 2010. - <http://www.geoff-hart.com/resources/2000/dynamicstyle.htm>.

Henninger Scott Supporting Design through Usability Guidelines and Patterns [Online] // CiteSeerX. - 2001. - 14. November 2010. - <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.4.868&rep=rep1&type=pdf>.

Hewett [et al.] CHAPTER 2: Human-Computer Interaction [Online] // ACM SIGCHI Curricula for Human-Computer Interaction. - 1996. - 13. November 2010. - <http://old.sigchi.org/cdg/cdg2.html>.

Highsmith III James A. Adaptive Software Development: A Collaborative Approach to Managing Complex Systems [Buch]. - [s.l.] : Dorset House Publishing Company, Incorporated, 1999.

Holtzblatt Karen, Wendell Jessamyn Burns und Wood Shelley Rapid contextual design: a how-to guide to key techniques for user-centered design [Buch] = Rapid contextual design. - San Fransisco : Elsevier Inc., 2005.

Hübscher Christian Skript User Centered Design I · MAS HCID 2008/9 «Vorgehensmodelle I» [PDF]. - 2008.

Infragistics Quince Pro - Build your own Private Design Libraries [Online] // infragistics. - 2010. - 09. 11 2010. - <http://www.infragistics.com/quince-pro.aspx#Overview>.

ISO 9241-11: Ergonomic requirements for office work with visual display terminals (VDTs) - Part 11 : Guidance on usability [Bericht]. - Genf : International Organization for Standardization (ISO), 1998.

Johnson H. Tom Social Rules for Creating a Style Guide [Online] // TECHWR-L. - 29. August 2010. - 30. Oktober 2010. - <http://www.techwr-l.com/socialrulesstyleguide>.

Koukoulletsos Kostas [et al.] Teaching Usability Principles with Patterns and Guidelines [Online] // CiteSeerX. - 2007. - 14. November 2010. - <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.85.1933&rep=rep1&type=pdf>.

Kreitzberg Dr. Charles B. und Little Ambrose Agile Ux Development [Online] // MSDN Magazine. - 2009. - 26. 08 2010. - <http://msdn.microsoft.com/en-us/magazine/dd882523.aspx>.

Lammi Janne Information Dashboard [Online] // Patternry. - 14. Juni 2010a. - September 2010. - <http://patternry.com/p=information-dashboard/>.

Lammi Janne Overlay [Online] // Patternry. - Juni 2010b. - September 2010. - <http://patternry.com/p=overlay/>.

Lammi Janne Refining Search [Online] // Patternry. - 14. Juni 2010c. - September 2010. - <http://patternry.com/p=refining-search>.

Lammi Janne Tag an Object [Online] // Patternry. - Juli 2010d. - September 2010. - <http://patternry.com/p=tag-object>.

Lehu Jean-Marc L'encyclopédie du marketing / Hrsg. d'Organisation Editeurs. - 02. 09 2004.

Lynch Patrick J. und Horton Sarah Page Structure and Site Design [Online] // Web Style Guide, 3rd Edition. - 2008. - 5. Oktober 2010. - <http://www.webstyleguide.com/wsg3/6-page-structure/3-site-design.html>.

Mariage Céline, Vanderdonckt Jean und Pribeanu Costin State of the Art of Web Usability Guidelines [Buchabschnitt] // Handbook of Human Factors in Web Design (Human Factors and Ergonomics) / Buchverf. Proctor Robert W. und Vu Kim-Phuong L.. - Mahwah : Lawrence Erlbaum Associates, 2004.

Mayhew Deborah J. The Usability Engineering Lifecycle [Buch]. - San Diego : Academic Press, 1999.

MediaWiki.org Welcome to MediaWiki.org [Online] // <http://www.mediawiki.org>. - 24. 01 2007. - 09. 11 2010. - <http://www.mediawiki.org>.

Myers Brad A. Why are Human-Computer Interfaces Difficult to Design and Implement? [Online] // CiteSeerX. - Juli 1993. - 30. Oktober 2010. - <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.45.4103>.

Ng Tinny Part 3: Design the workflow management component [Online] // Managing content in an employee workplace. - 30. April 2004. - 13. November 2010. - <http://www.ibm.com/developerworks/ibm/library/i-conmgt3/>.

Nielsen Jakob Guidelines for Multimedia on the Web [Online] // useit. - Dezember 1995. - 14. November 2010. - <http://www.useit.com/alertbox/9512.html>.

Nielsen Jakob Ten Usability Heuristics [Online] // useit. - 2005. - 14. November 2010. - http://www.useit.com/papers/heuristic/heuristic_list.html.

Nielsen Jakob Usability 101: Introduction to Usability [Online] // useit. - 25. August 2003. - 13. November 2010. - <http://www.useit.com/alertbox/20030825.html>.

Nielsen Jakob Why You Only Need to Test with 5 Users [Online] // useit. - 19. März 2000. - 23. Oktober 2010. - <http://www.useit.com/alertbox/20000319.html>.

Ohnemus Kenneth R. Web Style Guides: Who, What, Where [Konferenz] // SIGDOC'97: Proceedings of the 15th annual international conference on Computer documentation. - 1997. - S. 189-197.

Oracle Java Look And Feel Design Guidelines (Second Edition) [Online]. - 2001. - 30. Oktober 2010. - <http://java.sun.com/products/jlf/ed2/book/>.

Quesenbery Whitney Balancing the 5Es: Usability [Online] // WQusability. - 2004. - 13. November 2010. - <http://www.wqusability.com/articles/5es-citj0204.pdf>.

Quesenbery Whitney Building a Better Style Guide [Online] // WQusability. - 2001a. - 23. Oktober 2010. - <http://www.wqusability.com/articles/better-style-guide-paper.pdf>.

Quesenbery Whitney Building a Better Style Guide (Presentation Handouts) [Online] // WQusability. - 2001b. - 24. Oktober 2010. - <http://www.wqusability.com/articles/better-style-guide-slides.pdf>.

Richter Michael und Flückiger Markus Usability Engineering kompakt [Buch]. - München : Elsevier: Spektrum Akademischer Verlag, 2007.

RUP Rational Unified Process [Online] // Rational Unified Process. - 2003. - 11. Dezember 2010. - <http://rup.hops-fp6.org/>.

Shaikh A. Dawn und Lenz Kelsi Usability News [Online] // SURL. - Februar 2006. - 29. September 2010. - <http://surl.org/usabilitynews/81/webobjects.asp>.

Shneiderman Shneiderman's Eight Golden Rules of Interface Design [Online]. - 2004. - 14. November 2010. - <http://faculty.washington.edu/jtenenbg/courses/360/f04/sessions/schneidermanGoldenRules.html>.

Spool Jared M. Design Patterns: An Evolutionary Step to Managing Complex Sites [Online]. - 2003. - 14. November 2010. - http://www.uie.com/articles/design_patterns/.

Spool Jared M. Evolution Trumps Usability Guidelines [Online] // User Interface Engineering. - 08. September 2002. - 30. Oktober 2010. - http://www.uie.com/articles/evolution_trumps_usability/.

Steimle Toni Software Engineering am Beispiel vom Unified Process [PDF]. - 2008.

Tetzlaff Linda und Schwartz David R. The Use of Guidelines in Interface Design [Konferenz] // CHI '91 Proceedings of the SIGCHI conference on Human factors in computing systems. - [s.l.] : ACM, 1991. - S. 329-333.

Tidwell Jenifer Center Stage [Online] // Designing interfaces. - o.J. a. - September 2010. - http://designinginterfaces.com/Center_Stage.

Tidwell Jenifer Two-Panel Selector [Online] // Designing Interfaces. - o. J. b. - September 2010. - http://designinginterfaces.com/Two-Panel_Selector.

Tidwell Jennifer Common Ground: A Pattern Language for Human-Computer Interface Design [Online] // MIT - Massachusetts Institute of Technology. - 1999. - 30. Oktober 2010. - http://www.mit.edu/~jtidwell/common_ground_onefile.html.

Toxboe Anders Article list [Online] // UI Patterns. - 2010a. - September 2010. - <http://ui-patterns.com/patterns/ArticleList>.

Toxboe Anders Autocomplete [Online] // UI Patterns. - 2010b. - September 2010. - <http://ui-patterns.com/patterns/Autocomplete>.

Toxboe Anders Pagination [Online] // UI Patterns. - 2010c. - September 2010. - <http://ui-patterns.com/patterns/Pagination>.

TYPO3 What is TYPO3? [Online] // typo3.com. - 2010. - 09. 11 2010. - <http://typo3.com/about.1231.0.html>.

Usability.gov The Research-Based Web Design & Usability Guidelines [Online] // Usability.gov. - 2003. - 15. November 2010. - http://www.usability.gov/guidelines/guidelines_book.pdf.

van Welie Martijn Accordion [Online] // welie.com. - 2008a. - September 2010. - <http://www.welie.com/patterns/showPattern.php?patternID=accordion>.

van Welie Martijn Split Navigation [Online] // welie.com. - 2008b. - September 2010. - <http://www.welie.com/patterns/showPattern.php?patternID=split-navigation>.

Vanderdonckt Jean Development Milestones Towards a Tool for Working With Guidelines [Online] // CiteSeerX. - 1999. - 30. Oktober 2010. - <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.39.1075&rep=rep1&type=pdf>.

Vladimirskiy Aleksandr New workflows in Plone 3 [Online]. - 4. Februar 2009. - 12. November 2010. - <http://plone.org/documentation/kb/new-workflows-in-plone-3>.

Vogt Thomas Difficulties in Using Style Guides for Designing User Interfaces [Online] // CiteSeer. - 2001. - 24. Oktober 2010. - <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.87.9370&rep=rep1&type=pdf>.

W3C Web Content Accessibility Guidelines (WCAG) 2.0 [Online] // W3C. - Dezember 2008. - 14. November 2010. - <http://www.w3.org/TR/WCAG20/>.

Wikipedia Human-computer interaction [Online] // Wikipedia. - 11. November 2010. - 12. November 2010. - http://en.wikipedia.org/wiki/Human-computer_interaction.

Wilson Chauncey E. Guidance on Style Guides: Lessons Learned [Online] // STC - Usability & User Experience Community. - April 2001. - 30. Oktober 2010. - <http://www.stcsig.org/usability/newsletter/0104-style.html>.