|  |  |
| --- | --- |
|  | **kan** |

**Anforderungsdokument**

**Knowledge-Base „Metis“**

Betreuer: Dr. Sebastian Adam

Version: 0.6

Abgabedatum: 24.06.2016

**Dokumentverantwortliche**Marcel Hoffmann  
Isabella Tantalo

# Inhaltsverzeichnis

Inhaltsverzeichnis 2

Abbildungsverzeichnis 4

Tabellenverzeichnis 4

Abkürzungsverzeichnis 6

Änderungsverzeichnis 7

Vorwort 8

1 Einleitung 9

1.1 Zweck des Produkts 9

1.2 Projektziele 9

2 Stakeholder 10

2.1 Auftraggeber 10

2.2 Auftragnehmer 10

2.3 Produkt Anwender 10

3 Projekt Rahmenbedingungen 11

3.1 Einschränkungen 11

3.2 Relevante Annahmen 11

4 Produktkontext 12

4.1 Systemgrenzen und –abhängigkeiten 12

4.2 Product Use Cases 13

4.2.1 Übersicht 13

4.2.2 Details 13

5 Funktionale Anforderungen 18

5.1 Übersicht 18

5.1.1 Mussanforderungen 19

5.1.2 Kannanforderungen 19

5.1.3 Nice-to-have-Anforderungen 19

5.2 Details 20

6 Nicht funktionale Anforderungen 31

6.1 Übersicht 31

6.2 Performance 31

6.3 Zuverlässigkeit 32

6.4 Bedienbarkeit 33

6.5 Portierbarkeit 34

6.6 Skalierbarkeit 35

6.7 Testbarkeit 36

6.8 Änderbarkeit 37

6.9 Erweiterbarkeit 38

6.10 Qualitätsmatrix 39

Glossar 40

Anhang 41

# Abbildungsverzeichnis

Abbildung 1 System Use Case Diagramm 12

# Tabellenverzeichnis

Übersichten Verzeichnis

Übersicht 1: Product Use Cases 13

Übersicht 2: Funktionale Anforderungen 19

Übersicht 3: Muss Anforderungen 19

Übersicht 4: Kann Anforderungen 19

Übersicht 5: Nice-to-Have Anforderungen 20

Übersicht 6: Nicht Funktionale Anforderungen 32

Übersicht 7: Qualitätsmatrix 39

Use Case Verzeichnis

UC01: Artikel erstellen 14

UC02: Artikel bearbeiten 15

UC03: Artikel zurücksetzen 16

UC04: Artikel löschen 17

Funktionale Anforderungen

[FA01: Artikel erstellen 22](#_Toc454054655)

[FA02: Artikel bearbeiten 22](#_Toc454054656)

[FA03: Artikel löschen 23](#_Toc454054657)

[FA04: Artikel anzeigen 23](#_Toc454054658)

[FA05: Bearbeiter benachrichtigen 24](#_Toc454054659)

[FA06: Artikel/Dokument suchen 24](#_Toc454054660)

[FA07: Erweiterte Suche 25](#_Toc454054661)

[FA08: Dateien hochladen 25](#_Toc454054662)

[FA09: Artikel-Inhalte formatieren 26](#_Toc454054663)

[FA010: Artikel speichern 27](#_Toc454054664)

[FA011: Bearbeitung abbrechen 27](#_Toc454054665)

[FA012: Zuletzt gesehene Artikel anzeigen 27](#_Toc454054666)

[FA013: Automatische Titulierung 28](#_Toc454054667)

[FA014: Metadaten anzeigen 28](#_Toc454054668)

[FA015: Dateien referenzieren 29](#_Toc454054669)

[FA016: Vorherige Version anzeigen 29](#_Toc454054670)

[FA017: Version wiederherstellen 30](#_Toc454054671)

[FA018: Dateien herunterladen 30](#_Toc454054672)

[FA019: Artikel Vorlagen 31](#_Toc454054673)

[FA020: Autor- Daten übernehmen 31](#_Toc454054674)

[FA021: Phonetische Suche 32](#_Toc454054675)

[FA022: Reindizierung anstoßen 32](#_Toc454054676)

[FA023: Alert-Button konfigurieren 32](#_Toc454054677)

# Abkürzungsverzeichnis

FA Funktionale Anforderung

NFA Nicht funktionale Anforderung

NTT Nippon Telegraph and Telephone

OSS Open Search Server

UC Use Case

UI User Interface

# Änderungsverzeichnis

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Stand** | **Änderungsbeschreibung** | **Name** |
| 0.0 | 23.05.2016 | Erstellung der Vorlage | Max Becker |
| 0.1 | 24.05.2016 | Grobe Strukturierung und Inhalt hinzugefügt | Isabella Tantalo |
| 0.2 | 27.05.2016 | Product Use Cases hinzugefügt | Isabella Tantalo |
| 0.2 | 28.05.2016 | Review | Marcel Hoffmann |
| 0.3 | 03.06.2016 | Priorisierung, nicht Funktionale Anforderungen und Qualitätsmatrix hinzugefügt  Anforderungen nach User Storys angepasst, wg. Angleichung an Projekt | Isabella Tantalo |
| 0.4 | 06.06.2016 | Aufteilung Tabellenverzeichnis; Abhängigkeiten zu UC, Fas und NFAs hinzugefügt | Isabella Tantalo |
| 0.5 | 14.06.2016 | Überarbeitung und Finalisierung  Tabellenzeilen überarbeitet, Fitkriterien eingefügt | Marcel Hoffmann |
| 0.6 | 18.06.2016 | Reduzierung der NFR auf die nötigsten.  Vorwort überarbeitet.  Definition Anforderungskategorien hinzugefügt.  Kapitel 1 – 3 überarbeitet.  Abkürzungsverzeichnis mit Inhalt gefüllt.  Fehlende UseCases hinzugefügt: Datenbank reindizieren, Artikel suchen und Artikel lesen.  UseCase Übersicht gelöscht.  Unsortierte FA-Übersicht gelöscht.  Zuordnung UC 🡪 FA korrigiert  Glossar angefangen | Marcel Hoffmann |

# Vorwort

Dieses Dokument spezifiziert die Anforderungen an die Knowledge Database „Metis“, welche im Master Software Projekt der Hochschule Mannheim entwickelt wird. Das Projekt beginnt am 09.05.2016 und endet am 22.06.2016.

Die Anforderungen wurden mit Herrn Andreas Maier, einem Vertreter der Firma NTT Data Deutschland GmbH erhoben. Dazu wurden mehrere Interviews und Workshops mit Herrn Maier durchgeführt. Die Protokolle zu diesen Meetings können [hier](https://github.com/kanbanana/knowledgebase/tree/develop/docs/administration/Protokolle) eingesehen werden.

Beim Lesen dieses Dokuments ist zu beachten, dass die Benutzeroberfläche des Systems bereits am 09.05.2016, dem Tag der sogenannten Kick-off-Präsentation, prototypisch entwickelt wurde. Der Entwicklungsprozess wurde agil gehalten, das heißt es wurden, durch regelmäßige, wöchentliche Kundentreffen, die Anforderungen evaluiert und erweitert. Dadurch hat sich die Oberfläche der Software im Kern nicht grundlegend geändert, sondern schrittweise erweitert. Dies ermöglichte die Anforderungen sehr UI-nah zu formulieren.

Das folgende Anforderungsdokument wurde in dieser Form zur Benotung für die Vorlesung Requirements Engineering erstellt und fachmännisch von Dr. Sebastian Adam betreut.

# Einleitung

## Zweck des Produkts

Die Firma NTT Data Deutschland GmbH verwendet momentan Microsoft Sharepoint zum Austausch von Wissen und Dokumenten aus verschiedenen Projekten. Dabei ist es, auf Grund der Rechteverwaltung von MS Sharepoint nicht möglich dieses Wissen projektübergreifend zur Verfügung zu stellen. Das heißt, dass Entwickler nur auf die Dokumente zugreifen können, welche zu Projekten gehören an denen sie beteiligt sind. Dies offenbart teilweise erhebliche Barrieren, welche die Arbeitsabläufe der Mitarbeiter behindern. Ein weiteres Problem ist die Vermischung von frei zugänglichen und sehr sensiblen Daten oder Dokumenten. Auch Kunden der NTT Data Deutschland GmbH haben Zugriff auf den Sharepoint Server, wodurch diese, verständlicherweise, sicherstellen wollen, dass ihre Daten von keinem Dritten erreichbar sein können. Dies schlägt sich in sehr strikten und harten Sicherheitseinstellungen nieder, welche zusätzliche Barrieren schaffen.

Daher wurden die Studenten der Hochschule Mannheim damit beauftragt, eine eigene, Knowledge-Base-Lösung zu entwerfen und zu implementieren. Diese wird dann die Projektdaten beinhalten und diese somit von sensiblen Daten trennen. Außerdem kann durch die Anbindung an das firmeninterne Intranet eine wesentlich einfachere und unkomplizierte Rechtevergabe gewährleistet werden.

Der offizielle Name des Produktes lautet „Metis“.

## Projektziele

Das Ziel dieses Projektes ist es dem Kunden eine Knowledge Base Anwendung zur Verfügung zu stellen. Mit Hilfe dieser werden die firmeninternen Arbeitsabläufe der Mitarbeiter unterstützt und verbessert. Dabei profitieren alle Mitarbeiter von einer einheitlichen und einfach zu bedienenden Wissensdatenbank, indem bekannte Probleme oder Fehler abteilungsübergreifend eingespeist und unabhängig erreicht werden können. Durch diesen zentralen Wissensknoten können beispielsweise auch neue Mitarbeiter sehr einfach und schnell in Projekte eingearbeitet werden. Das Potenzial der Anwendung „Metis“ erstreckt sich von einer unabhängigen Problemlösungshilfe bis hin zu einem Projektinformationsspeicher. Durch die benötigten Kommunikationswege zwischen Mitarbeiter, welche ein (komplexes) Problem entdeckt haben, und einzelnen Spezialisten, welche dieses Problem in der Vergangenheit im Idealfall bereits gelöst haben, entstehen unnötig lange Arbeitsabläufe mit teilweise hohen Abhängigkeiten. Sollte der benötigte Spezialist gerade im Urlaub sein oder gesundheitlich nicht in der Lage sein zur Arbeit zu erscheinen, wer könnte das besagte Problem dann lösen? Soll der Kunde, welcher dieses Problem vielleicht entdeckt hat, mehrere Tage auf eine Lösung warten und mit einem, vielleicht nur teilweise funktionsfähigem, Produkt ausharren? Dies ist mehr als inakzeptabel. Daher bietet „Metis“ eine Möglichkeit Wissen und Know-How zentral zu speichern und unabhängig erreichbar zu machen.

# Stakeholder

## Auftraggeber

Die NTT Data ist Teil der NTT Gruppe, die 1953 als Nippon Telegraph and Telephone Public Cooperation gegründet wurde. Mit über 242.000 Mitarbeiter in 88 Ländern ist die NTT Gruppe ein weltweit führendes IT- und Telekommunikationsunternehmen mit Hauptsitz in Japan und gilt als einer der größten Daten-Center-Betreiber. Als Tochterunternehmen bietet die NTT Data umfassende IT-Service-Leistungen, Beratung, Systemintegration und IT Outsourcing in ihrem Leistungsspektrum. Als Ansprechpartner steht Herr Andreas Maier, von der Tochterfirma NTT Data Deutschland GmbH zur Verfügung. Dieser sucht eine Möglichkeit in seiner Abteilung eine neue, firmeneigene Knowledge-Base zu etablieren. Dabei soll nach Abschluss des Projektes entschieden werden, ob „Metis“ geeignet ist in einem Feldversuch für bis zu 225 Mitarbeiter freigeschaltet zu werden. Bei Erfolg des geplanten firmeninternen Feldversuches hat man von Kundenseite aus eine Erweiterung des Produktes in Aussicht gestellt.

## Auftragnehmer

Die Kanbanana besteht aus 15 Studenten aus verschiedenen Fachsemestern aus dem Studiengang des Masters Informatik mit Schwerpunkt Software Engineering an der Hochschule Mannheim. Im zweiten Semester müssen die Studenten selbstständig ein Projekt planen, durchführen und dabei ein Produkt für einen ausgewählten Kunden entwickeln.

Das Kanbanana Team besteht aus mehreren kleinen Teams. Die Projektleitung besteht aus Max Becker und Mai Chau Nguyen. Für Fragen und den Kontakt mit dem Kunden stehen Robert Kettler und Isabella Tantalo zur Verfügung.

## Produktanwender

Die voraussichtlichen Produktanwender sind Mitarbeiter aus der Abteilung von Herrn Maier. Dieser leitet eine Entwicklungsabteilung, welche eine gewisse technische Affinität und Nähe zur Programmierung besitzt.

# Projekt Rahmenbedingungen

## Einschränkungen

Das Projekt muss bis zum 22.06.2016 abgeschlossen sein. Dabei müssen alle, im Vertrag zwischen NTT Data Deutschland GmbH und dem Team „Kanbanana“ genannten, Leistungen ausgehändigt werden.

Es werden keine speziellen Zulassungen oder Zertifikate benötigt.

## Relevante Annahmen

Da das Endprodukt nur im firmeninternen Intranet von NTT Data Deutschland Verwendung finden wird, wird auf ein Sicherheitskonzept verzichtet.

# Produktkontext

## Systemgrenzen und –abhängigkeiten



Abbildung System Use Case Diagramm

Aus den, von Seiten des Kunden beschriebenen, Geschäftsfällen wurden die oben im Diagramm gezeigten Use Cases abgeleitet. Es ergeben sich zwei Benutzergruppen: Administrator und Benutzer. Dabei müssen alle Benutzer Artikel anlegen, lesen, bearbeiten und löschen können. Ebenfalls muss man Artikel suchen können, um diese bei größerem Datenbestand erreichen zu können. Für die Suche wird es zwei Varianten geben: die einfache und die erweiterte. Bei der normalen Suche gibt der Benutzer einen Suchbegriff ein und erhält alle Artikel, welche diesen Suchbegriff, ganz oder in Teilen, enthält. Bei der erweiterten Suche ist vorgesehen, dem Benutzer die Möglichkeit zu geben, Artikel nach Name des Autors oder Erstelldatum des Artikels suchen zu können. Ein Administrator des Systems hat die Möglichkeit eine Reindexierung manuell anstoßen zu können, um diesen langwierigen Prozess gezielt nach Feierabend, über Nacht durchlaufen lassen zu können.

## Product Use Cases

### Details

|  |  |
| --- | --- |
|  |  |
| **Name** | Artikel erstellen |
| **Ziel** | Der Benutzer möchte einen neuen Artikel erstellt im System und speichern. |
| **Rolle** | Benutzer |
| **Voraussetzung und eingehende Daten** | Der Benutzer befindet sich auf der „Searchresult-View“;  Benötigte Daten: Suchbegriffe |
| **Beschreibung** | 1. Der Benutzer klickt auf den Button „New“. 2. Das System wechselt zur Erstellen-Ansicht und fügt den Suchbegriff als Titel ein. 3. Der Benutzer gibt seinen Namen, seine E-Mail-Adresse, den Artikel-Inhalt (Text, Bilder, Tabellen) ein und fügt ggfs. Dokumente ein. Der Benutzer drückt auf den Button „Save“. 4. Das System speichert den Artikel und hochgeladene Dateien dauerhaft. 5. Das System stößt eine Indexierung an. 6. Das System wechselt zur Lese-Ansicht des Artikels und gibt eine Meldung aus. |
| **Ausnahmefälle** | Zu 4.: Das System kann den Artikel nicht speichern:  4.10 Das System gibt eine Fehlermeldung aus. Weiter mit 3. |
| **Alternative Abläufe** | Zu 3.: Der Benutzer klickt auf den Button „Cancel“:  4.20 Das System gibt eine Sicherheitsabfrage aus.  4.21 Der Benutzer kann auf den Button „Leave“ oder den Button „Stay“ drücken.  4.211 Falls der Benutzer „Leave“ drückt: Das System verwirft den Artikel und wechselt zur vorherigen Ansicht.  4.212 Falls der Benutzer „Stay“ drückt: Weiter mit 3. |
| **Geschäftsregeln** | Pflichtfelder: Titel des Artikels |
| **Qualitätsanforderungen** | 6.4 Bedienbarkeit  6.5 Portierbarkeit  6.6 Skalierbarkeit |
| **Systemfunktionen** | FA01: Artikel erstellen  FA013: Automatische Titulierung  FA08: Dateien hochladen  FA010: Artikel speichern  FA04: Artikel anzeigen |
| **Ergebnisse und ausgehende Daten** | Es befindet sich ein neuer Artikeleintrag in der Datenbank |

UC0: Artikel erstellen

|  |  |
| --- | --- |
|  |  |
| **Name** | Artikel bearbeiten |
| **Ziel** | Der Benutzer möchte Teile eines bereits vorhandenen Artikels verändern und speichern. |
| **Rolle** | Benutzer |
| **Voraussetzung und eingehende Daten** | Der Benutzer befindet sich in der Lese-Ansicht des Artikels. |
| **Beschreibung** | 1. Der Benutzer klickt auf den Button „Edit“. 2. Das System wechselt in die „Bearbeiten“-Ansicht des Artikels. 3. Der Benutzer ändert den Artikel-Inhalt, den Titel des Artikels, lädt neue Dokumente hoch oder löscht Dokumente aus dem Artikel. Der Benutzer drückt auf den Button „Save“.   4. Das System speichert den Artikel.  5. Das System wechselt in die Lese-Ansicht des Artikels und gibt eine Erfolgsmeldung aus. |
| **Ausnahmefälle** | Zu 4.: Das System kann den Artikel nicht speichern. |
| **Alternative Abläufe** | Zu 3.: Der Benutzer drückt den Button „Cancel“.  4.10 Das System gibt eine Sicherheitsabfrage aus.  4.11 Der Benutzer kann auf den Button „Leave“ oder den Button „Stay“ drücken.  4.112 Falls der Benutzer „Leave“ drückt: Das System verwirft den Artikel und wechselt zur vorherigen Ansicht.  4.113 Falls der Benutzer „Stay“ drückt: Weiter mit 3.  Zu 3.: Der Benutzer drückt den Button „Delete“:  siehe UC04: Artikel löschen  Zu 3.: Der Benutzer drückt auf den Button „previouse Version of“:  siehe UC03: Artikel zurücksetzen |
| **Geschäftsregeln** | Der ursprüngliche Autor und E-Mail-Adresse darf nicht geändert werden. |
| **Qualitätsanforderungen** | 6.5 Portierbarkeit  6.6 Skalierbarkeit |
| **Systemfunktionen** | FA02: Artikel bearbeiten  FA08: Dateien hochladen  FA09: Artikel-Inhalte formatieren  FA010: Artikel speichern  FA011: Bearbeitung abbrechen  FA015: Dateien referenzieren |
| **Ergebnisse und ausgehende Daten** | Der geänderte Artikel befindet sich in einer neuen Version in der Datenbank, der Artikel vor der Änderung ist die vorherige Version. Weitere vorherige Versionen können gelöscht werden. |

UC0: Artikel bearbeiten

|  |  |
| --- | --- |
|  |  |
| **Name** | Artikel lesen |
| **Ziel** | Der Benutzer kann den Inhalt eines Artikels vollständig lesen. |
| **Rolle** | Benutzer |
| **Voraussetzung und eingehende Daten** | Der Benutzer befindet sich auf der Searchresult-View. |
| **Beschreibung** | 1. Der Benutzer klickt auf den Titel des zu lesenden Artikels. 2. Das System lädt die Leseansicht des angeklickten Artikels. |
| **Qualitätsanforderungen** | 6.4 Bedienbarkeit  6.5 Portierbarkeit |
| **Systemfunktionen** | FA02: Artikel bearbeiten  FA03: Artikel löschen |
| **Ergebnisse und ausgehende Daten** | Der Inhalt des Artikels wurde vollständig in der Leseansicht geladen. |

UC0: Artikel lesen

|  |  |
| --- | --- |
|  |  |
| **Name** | Artikel zurücksetzen |
| **Ziel** | Der Benutzer möchte die vorherige Version des Artikels wiederherstellen. |
| **Rolle** | Benutzer |
| **Voraussetzung und eingehende Daten** | Der Benutzer befindet sich in der Bearbeiten-Ansicht des Artikels. |
| **Beschreibung** | 1. Der Benutzer drückt auf den Button „Previouse Version“. 2. Das System öffnet die vorherige Version des Artikels. 3. Der Benutzer kopiert den Text aus der vorherige Version und fügt sie in der aktuellen Version ein (Copy&Paste in die aktuelle Version) und drückt auf den Button „Save“ 4. Das System speichert den Artikel und gibt eine Erfolgsmeldung aus. |
| **Ausnahmefälle** | Zu 2.: Vorherige Version des Artikels kann nicht geladen werden.  Zu 3.: Speichern des Artikels schlägt fehl. |
| **Qualitätsanforderungen** | 6.5 Portierbarkeit  6.6 Skalierbarkeit  6.8 Erweiterbarkeit |
| **Systemfunktionen** | FA02: Artikel bearbeiten  FA016: Vorherige Version anzeigen  FA017: Version wiederherstellen |
| **Ergebnisse und ausgehende Daten** | Der geänderte Artikel befindet sich in einer neuen Version in der Datenbank, der Artikel vor der Änderung ist die vorherige Version. Weitere vorhergehende Versionen können gelöscht werden. |

UC0: Artikel zurücksetzen

|  |  |
| --- | --- |
|  |  |
| **Name** | Artikel löschen |
| **Ziel** | Der Benutzer möchte einen im System gespeicherten Artikel dauerhaft aus dem System entfernen. |
| **Rolle** | Benutzer |
| **Voraussetzung und eingehende Daten** | Der Benutzer befindet sich in der Bearbeiten-Ansicht des Artikels |
| **Beschreibung** | 1. Der Benutzer klickt auf den Button „Delete Article“. 2. Das System gibt eine Sicherheitsabfrage aus. 3. Der Benutzer drückt auf den Button „Delete“ 4. Das System löscht den Artikel und die dazugehörigen Dateien. 5. Das System wechselt zur Startseite. 6. Das System startet eine Reindexierung. |
| **Ausnahmefälle** | Zu 6.: Das Löschen des Artikels schlägt fehl. |
| **Alternative Abläufe** | Zu 3.: Der Benutzer drückt auf den Button „Cancel“  Siehe Alternativ Ablauf UC02: Artikel bearbeiten |
| **Qualitätsanforderungen** | 6.3 Zuverlässigkeit  6.4 Bedienbarkeit |
| **Systemfunktionen** | FA02: Artikel bearbeiten  FA03: Artikel löschen |
| **Ergebnisse und ausgehende Daten** | Der Artikel wurde erfolgreich aus dem System gelöscht. |

UC0: Artikel löschen

|  |  |
| --- | --- |
|  |  |
| **Name** | Artikel suchen |
| **Ziel** | Es werden Artikel, welche den eingegebenen Suchbegriff teilweise oder ganz enthalten in einer Liste auf der Startseite des Benutzers angezeigt. |
| **Rolle** | Benutzer |
| **Voraussetzung und eingehende Daten** | Der Benutzer befindet sich auf der Startseite des Systems. Der Benutzer muss einen Suchbegriff in das Suchfeld eingeben. |
| **Beschreibung** | 1. Der Benutzer gibt einen Suchbegriff in das Suchfeld ein und drückt die „Enter-Taste“. 2. Das System führt die Suche durch, lädt eine Liste mit Suchergebnissen und zeigt diese an. |
| **Ausnahmefälle** | Zu 2. Es wird kein Ergebnis zu dem eingegebenen Suchbegriff gefunden. |
| **Qualitätsanforderungen** | 6.2 Performance  6.3 Zuverlässigkeit  6.4 Bedienbarkeit  6.6 Skalierbarkeit |
| **Systemfunktionen** | FA06: Artikel/Dokument suchen  FA07: Erweiterte Suche  FA021: Phonetische Suche |
| **Ergebnisse und ausgehende Daten** | Es werden alle gefundenen Suchergebnisse als Liste angezeigt. |

UC0: Artikel suchen

|  |  |
| --- | --- |
|  |  |
| **Name** | Datenbank reindizieren |
| **Ziel** | Der Admin stößt manuell eine Reindizierung des Systems an. |
| **Rolle** | Admin |
| **Voraussetzung und eingehende Daten** | Der Admin hat das Webinterface der OSS Suchengine geöffnet. |
| **Beschreibung** | 1. Der Admin verwendet das Webinterface um diverse Konfigurationen vorzunehmen. 2. Das System führt die Reindizierung durch. |
| **Qualitätsanforderungen** | 6.7 Änderbarkeit |
| **Systemfunktionen** | FA022: Reindizierung anstoßen |
| **Ergebnisse und ausgehende Daten** | Das System hat die Reindizierung gestartet. |

UC0: Datenbank reindizieren

# Funktionale Anforderungen

Alle funktionalen Anforderungen sind in eine von drei Priorisierungskategorien eingeteilt: Muss, Kann oder Nice-to-have. Muss-Anforderung bedeutet dabei, dass diese Anforderung essentiell für die Funktionalität des Systems ist. Kann-Anforderungen beschreiben jene, die nicht essentiell für die Funktionsfähigkeit des Systems sind, aber weitergehende Unterstützung der Geschäftsprozesse bedeuten können. Nice-to-have-Anforderungen sind Anforderungen die dem System einen sehr kleinen, detaillierten Mehrwert verschaffen können und daher keine hohe Priorität besitzen.

### Mussanforderungen

|  |  |  |
| --- | --- | --- |
| **Nr.** | **Name** | **Priorität** |
| FA01 | Artikel erstellen | muss |
| FA02 | Artikel bearbeiten | muss |
| FA03 | Artikel löschen | muss |
| FA04 | Artikel anzeigen | muss |
| FA06 | Artikel/Dokument suchen | muss |
| FA08 | Dateien hochladen | muss |
| FA09 | Artikel-Inhalte formatieren | muss |
| FA010 | Artikel speichern | muss |
| FA011 | Bearbeitung abbrechen | muss |
| FA016 | Vorherige Version anzeigen | muss |
| FA018 | Artikel Vorlagen | muss |
| FA022 | Reindizierung anstoßen | muss |

Übersicht : Muss Anforderungen

### Kannanforderungen

|  |  |  |
| --- | --- | --- |
| **Nr.** | **Name** | **Priorität** |
| FA015 | Dateien referenzieren | kann |
| FA05 | Bearbeiter benachrichtigen | kann |
| FA07 | Erweiterte Suche | kann |

Übersicht : Kann Anforderungen

### Nice-to-have-Anforderungen

|  |  |  |
| --- | --- | --- |
| FA012 | Zuletzt gesehene Artikel anzeigen | nice-to-have |
| FA013 | Automatische Titulierung | nice-to-have |
| FA019 | Artikel Vorlagen | nice-to-have |
| FA020 | Autor- Daten übernehmen | nice-to-have |
|  | Phonetische Suche | nice-to-have |

Übersicht : Nice-to-Have Anforderungen

## Details

|  |  |  |  |
| --- | --- | --- | --- |
| **Anforderungsnr**. |  | **Use Cases** | UC01 |
| **Name** | Artikel erstellen | | |
| **Beschreibung** | Als Benutzer muss ich Artikel verfassen können, um mein Wissen anderen Benutzern bereit zu stellen. | | |
| **Begründung** | Das schriftliche festhalten von Wissen soll das Wissen des Autors an andere Benutzer verteilt werden. | | |
| **Fit Kriterium** | Durch den Klick auf den „new“-Button in der Search-View gelangt man in die „create/edit“-View, um einen neuen Artikel anlegen zu können. | | |
| **Priorität** | muss | | |
| **Abhängigkeiten** | FA08: Dateien hochladen  FA09: Artikel-Inhalte formatieren  FA011: Bearbeitung abbrechen  FA013: Automatische Titulierung  FA015: Dateien referenzieren  FA019: Artikel Vorlagen  FA020: Autor- Daten übernehmen | | |
| **Weiterführende Informationen** | Ein Artikel besteht aus:   * Titel * Inhalt (Text, Bilder, Tabellen, Formatierung) * Artikel Metadaten (Autor, Erstellungsdatum, Letzter Bearbeiter, Datum letzte Bearbeitung, # Views) * Optional: Dokumente | | |

FA0: Artikel erstellen

|  |  |  |  |
| --- | --- | --- | --- |
| **Anforderungsnr.** | FA02 | **Use Cases** | UC02; UC03; UC04 |
| **Name** | Artikel bearbeiten | | |
| **Beschreibung** | Als Benutzer muss ich Artikel bearbeiten können, um die Möglichkeit zu haben gespeicherte Fehler korrigieren zu können. | | |
| **Begründung** | Artikel können Fehler enthalten, die verbessert werden müssen. Auch können Artikel um weitere Informationen oder Dateien erweitert werden, um die Verständlichkeit zu erhöhen. | | |
| **Fit Kriterium** | Webseite zeigt Success-Meldung und das Backend speichert die geänderte, neue Version des Artikels. | | |
| **Priorität** | muss | | |
| **Abhängigkeiten** | FA010: Artikel speichern  FA09: Artikel-Inhalte formatieren  FA08: Dateien hochladen  FA011: Bearbeitung abbrechen  FA015: Dateien referenzieren  FA020: Autor- Daten übernehmen | | |
| **Weiterführende Informationen** | Bearbeitet werden dürfen:   * Den Artikel-Inhalt * Den Titel * Dateien hinzugefügt oder entfernt werden (hochladen und löschen) * Formatierung * Kompletten Artikel löschen | | |

FA0: Artikel bearbeiten

|  |  |  |  |
| --- | --- | --- | --- |
| **Anforderungsnr**. | FA03 | **Use Cases** | UC04 |
| **Name** | Artikel löschen | | |
| **Beschreibung** | Als Benutzer muss ich Artikel löschen können, um obsolete Informationen beseitigen zu können. | | |
| **Begründung** | Falsche oder veraltete Artikel verlangsamen die Suche nach Informationen. | | |
| **Fit Kriterium** | Die Webseite navigiert automatisch zum Startbildschirm des Systems und das Backend löscht den Artikel aus dem Filesystem und der Datenbank. Der Artikel kann über die Suchfunktion des Systems nicht mehr gefunden werden. | | |
| **Priorität** | muss | | |
| **Abhängigkeiten** | FA02: Artikel bearbeiten | | |
| **Weiterführende Informationen** | Das Löschen der Artikel muss in der Bearbeiten-Ansicht erfolgen. Siehe Mockups „Edit“-Ansicht | | |

FA0: Artikel löschen

|  |  |  |  |
| --- | --- | --- | --- |
| **Anforderungsnr.** | FA04 | **Use** **Cases** | UC01; UC02; UC03 |
| **Name** | Artikel anzeigen | | |
| **Beschreibung** | Als Benutzer muss ich Artikel lesen können, um mir benötigtes Wissen zu einem Thema anzueignen. | | |
| **Begründung** | Die Suche zeigt nur eine Kurzfassung der Artikel. Ein Anwender sollte den ganzen Artikel lesen können. | | |
| **Fit Kriterium** | Das System zeigt den Detailview zum ausgewählten Artikel an. | | |
| **Priorität** | muss | | |
| **Abhängigkeiten** | FA01: Artikel erstellen  FA010: Artikel speichern  FA06: Artikel/Dokument durchsuchen  FA07: Erweiterte Suche  FA012: Zuletzt gesehene Artikel anzeigen | | |

FA0: Artikel anzeigen

|  |  |  |  |
| --- | --- | --- | --- |
| **Anforderungsnr.** | FA05 | **Use** **Cases** | UC02; UC03 |
| **Name** | Bearbeiter benachrichtigen | | |
| **Beschreibung** | Als Benutzer kann ich den letzten Bearbeiter per E-Mail benachrichtigen, um ihn auf Fehler oder Ungereimtheiten in seinem Artikel aufmerksam zu machen. | | |
| **Begründung** | Die Benachrichtigung soll den letzten Bearbeiter auf eine neue Änderung oder auf gemachte Fehler aufmerksam machen. | | |
| **Fit Kriterium** | Das System öffnet das Standard Emailsystem des BenutzerPC’s und fügt die angegebene Emailadresse in die Adressatenzeile ein. | | |
| **Priorität** | kann | | |
| **Abhängigkeiten** | FA 01 Artikel erstellen  FA02 Artikel bearbeiten | | |
| **Weiterführende Informationen** | Siehe Mockups „Detail“-Ansicht.  Der Kunde war mit einer Benachrichtigung über E-Mail einverstanden. Dabei ist es auch HTML mailto ausreichend.  Als letzter Bearbeiter gilt der Autor, der zuletzt den Artikel bearbeitet hat. | | |

FA0: Bearbeiter benachrichtigen

|  |  |  |  |
| --- | --- | --- | --- |
| **Anforderungsnr.** | FA06 | **Use Cases** | UC01; UC02; UC03; UC04; UC06 |
| **Name** | Artikel/Dokument suchen | | |
| **Beschreibung** | Als Benutzer muss ich Artikel suchen können, um Artikel zu finden, welche basierend auf den eingegebenen Suchterm, relevante Informationen bereitstellen. | | |
| **Begründung** | Artikel und Dokumente sollen leicht zu finden sein | | |
| **Fit Kriterium** | Das System zeigt eine Auflistung von Suchergebnissen, die den eingegebenen Suchbegriff beinhalten, an. | | |
| **Priorität** | muss | | |
| **Abhängigkeiten** | FA01: Artikel erstellen  FA010: Artikel speichern | | |

FA0: Artikel/Dokument suchen

|  |  |  |  |
| --- | --- | --- | --- |
| **Anforderungsnr.** | FA07 | **Use Cases** | UC03; UC06 |
| **Name** | Erweiterte Suche | | |
| **Beschreibung** | Als Benutzer muss ich Artikel nach erweiterten Kriterien suchen können, um Artikel zu finden, die den Kriterien entsprechen zu finden. | | |
| **Begründung** | Die erweiterte Suche kann das Suchergebnis begrenzen, sodass ein Anwender schneller ein passendes Ergebnis finden kann. | | |
| **Fit Kriterium** | Es kann nach Autorennamen gesucht werden. Dabei werden alle Artikel gefunden wo der eingegebene Name als Autor oder als letzter Bearbeiter hinterlegt wurde. | | |
| **Priorität** | kann | | |
| **Abhängigkeiten** | FA01: Artikel erstellen  FA010: Artikel speichern | | |
| **Weiterführende Informationen** | Erweiterte Kriterien:   * Nach Metadaten suchen | | |

FA0: Erweiterte Suche

|  |  |  |  |
| --- | --- | --- | --- |
| **Anforderungsnr.** | FA08 | Use Cases | UC01; UC02 |
| **Name** | Dateien hochladen | | |
| **Beschreibung** | Als Benutzer muss ich Dateien zu einem Artikel hochladen zu können, um die Informationen, die sich in den Dokumenten befinden, leicht an andere Benutzer weiterzugeben. | | |
| **Begründung** | Das Hochladen von Dateien ermöglicht eine einfache Weitergabe von Dokumenten und Informationen an allen Anwendern, ohne eine manuelle Duplizierung der Information. | | |
| **Fit Kriterium** | Das System blendet eine Successmeldung ein. Außerdem wird die hochgeladene Datei als Anhang zum Artikel angezeigt. Außerdem wird die Datei im Filesystem gespeichert. | | |
| **Priorität** | muss | | |
| **Abhängigkeiten** | FA01: Artikel erstellen  FA02: Artikel bearbeiten  FA06: Artikel/Dokument durchsuchen | | |
| **Weiterführende Informationen** | Unterstützte Dateiformate:   * Doc * Docx * Pdf * Txt | | |

FA0: Dateien hochladen

|  |  |  |  |
| --- | --- | --- | --- |
| **Anforderungsnr.** | FA09 | **Use Cases** | UC01; UC02 |
| **Name** | Artikel-Inhalte formatieren | | |
| **Beschreibung** | Als Benutzer muss ich beim Verfassen und Bearbeiten von Artikeln Texte formatieren können, um die Lesbarkeit und die Verständlichkeit des Artikels erhöhen zu können. | | |
| **Begründung** | Die Formatierung der Texte soll die Informationen besser verdeutlichen. | | |
| **Fit Kriterium** | Siehe Abschnitt 9.2.4 im Architekturdokument. | | |
| **Priorität** | muss | | |
| **Abhängigkeiten** | FA01: Artikel erstellen  FA02: Artikel bearbeiten | | |
| **Weiterführende Informationen** | Formatierungsoptionen:  - Schriftgröße/Überschrift/Fett/Kursiv/Unterstrichen  - Aufzählung  - Tabellen | | |

FA0: Artikel-Inhalte formatieren

|  |  |  |  |
| --- | --- | --- | --- |
| **Anforderungsnr.** | FA010 | **Use Cases** | UC01; UC02 |
| **Name** | Artikel speichern | | |
| **Beschreibung** | Als Benutzer muss ich erstellte und bearbeitete Artikel speichern können. | | |
| **Begründung** | Erstelle Artikel und Artikel Änderungen müssen dauerhaft sein. | | |
| **Fit Kriterium** | Nach dem Klick auf den „Save“ Button in der „create“ oder „edit“ View, wird eine Meldung „Success“ angezeigt, der Artikelinhalt im Filesystem hinterlegt oder die vorherige Version des Artikels wird aktualisiert. Außerdem werden dessen Metadaten in die Datenbank geschrieben. Danach ist der neue Artikel in der Suche auffindbar. | | |
| **Priorität** | muss | | |
| **Abhängigkeiten** | FA01: Artikel erstellen  FA02: Artikel bearbeiten | | |

FA0: Artikel speichern

|  |  |  |  |
| --- | --- | --- | --- |
| **Anforderungsnr.** | FA011 | **Use Cases** | UC01; UC02 |
| **Name** | Bearbeitung abbrechen | | |
| **Beschreibung** | Das System muss dem Benutzer die Möglichkeit bieten, die Bearbeitung oder Erstellung eines Artikels abzubrechen. | | |
| **Begründung** | Es muss möglich sein Fehler im Artikel zu verbessern oder Informationen auf einen neueren Stand zu bringen. | | |
| **Fit Kriterium** | Beim Klick auf den Abbrechen-Button in der Edit-View wird man zur Leseansicht des Artikels geleitet.  Beim Abbrechen der Bearbeitung eines neuen Artikels, gelangt man zur Startseite zurück. | | |
| **Priorität** | muss | | |
| **Abhängigkeiten** | FA01: Artikel erstellen  FA02: Artikel bearbeiten | | |
| **Weiterführende Informationen** | Siehe Mock-ups, „Cancel“-Button | | |

FA0: Bearbeitung abbrechen

|  |  |  |  |
| --- | --- | --- | --- |
| **Anforderungsnr.** | FA012 | **Use Cases** | UC03 |
| **Name** | Zuletzt gesehene Artikel anzeigen | | |
| **Beschreibung** | Als Benutzer kann ich auf die von mir zuletzt besuchten Artikel auf der Startseite zugreifen, um Zeit zu sparen und die Benutzerfreundlichkeit zu erhöhen. | | |
| **Begründung** | Erspart Zeit, da nicht erneut nach einem Dokument oder Artikel gesucht werden muss. | | |
| **Fit Kriterium** | Auf der Startseite des Produktes werden die 5 zuletzt gesehenen Artikel aufgelistet. | | |
| **Priorität** | Nice-to-have | | |
| **Abhängigkeiten** | FA01: Artikel erstellen  FA04: Artikel anzeigen  FA06: Artikel/Dokument suchen | | |
| **Weiterführende Informationen** | Siehe Mock-ups „Startseite“ | | |

FA0: Zuletzt gesehene Artikel anzeigen

|  |  |  |  |
| --- | --- | --- | --- |
| **Anforderungsnr.** | FA013 | **Use Cases** | UC01 |
| **Name** | Automatische Titulierung | | |
| **Beschreibung** | Fall ich bei der Suche keinen passenden Artikel finde und ich einen neuen Artikel verfasse, dann übernimmt das System automatisch den Suchbegriff aus der Suche als Titel des Dokuments. | | |
| **Begründung** | Der Autor eines Artikels sollte den Titel nicht erneut eingeben müssen. | | |
| **Fit Kriterium** | Ein Suchbegriff zu dem kein Artikel gefunden wurde, wird beim Anlegen eines neuen Artikels als Titel in die View zum Anlegen eines Artikels übernommen. | | |
| **Priorität** | Nice-to-have | | |
| **Abhängigkeiten** | FA01: Artikel erstellen | | |
| **Weiterführende Informationen** | Es wird davon ausgegangen, dass der Suchbegriff meistens dann der Titel sein wird oder zumindest dieser enthalten sein wird. Der Titel sollte dennoch bearbeitet werden können. | | |

FA0: Automatische Titulierung

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Anforderungsnr.** | FA014 | **Use Cases** | UC01; UC02; UC03 | |
| **Name** | Metadaten anzeigen | | | |
| **Beschreibung** | Das System muss dem Benutzer automatisch die Metadaten eines Artikels anzeigen. | | | |
| **Fit Kriterium** | Die Views Searchresults und Detail enthalten die Metadaten zu den angezeigten Artikeln. (Autor, letzter Änderer, Datum + Uhrzeit) | | |
| **Priorität** | kann | | |
| **Abhängigkeiten** | FA04: Artikel anzeigen | | |

FA0: Metadaten anzeigen

|  |  |  |  |
| --- | --- | --- | --- |
| **Anforderungsnr.** | FA015 | **Use Cases** | UC01; UC02 |
| **Name** | Dateien referenzieren | | |
| **Beschreibung** | Als Benutzer kann ich hochgeladene Dateien im Artikel-Inhalt referenzieren. | | |
| **Begründung** | Ein Benutzer beschreibt im Artikel-Inhalt die Datei. Eine Verlinkung der Datei im Text reduziert Missverständnisse und erhöht die Usability. | | |
| **Fit Kriterium** | Man kann Dateien per Hyperlink im Text referenzieren. | | |
| **Priorität** | kann | | |
| **Abhängigkeiten** | FA01: Artikel erstellen  FA02: Artikel bearbeiten | | |
| **Weiterführende Informationen** | Siehe Mock-ups „Detailansicht“ | | |

FA0: Dateien referenzieren

|  |  |  |  |
| --- | --- | --- | --- |
| **Anforderungsnr.** | FA016 | **Use Cases** | UC03 |
| **Name** | Vorherige Version anzeigen | | |
| **Beschreibung** | Als Benutzer kann ich die vorherige Version des Artikels einsehen, um Änderungen nachverfolgen zu können. | | |
| **Begründung** | Nachverfolgbarkeit der zuletzt gemachten Änderungen | | |
| **Fit Kriterium** | Die vorherige Version kann per Klick auf den Button „previous version“ per Pop-up-Fenster angezeigt werden. | | |
| **Priorität** | muss | | |
| **Abhängigkeiten** | FA01: Artikel erstellen  FA02: Artikel bearbeiten | | |

FA0: Vorherige Version anzeigen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Anforderungsnr.** | FA017 | **Use Cases** | UC03; UC05 | |
| **Name** | Version wiederherstellen | | | |
| **Beschreibung** | Das System muss dem Benutzer die Möglichkeit bieten, die vorherige Version eines Artikels wiederherzustellen. | | | |
| **Begründung** | Wurde ein Artikel falsch geändert und gespeichert, kann dies so rückgängig gemacht werden. | | | |
| **Fit Kriterium** | Man kann per copy und paste im Pop-up-Fenster die alte Version des Dokumentes einfügen und direkt speichern. | | |
| **Priorität** | kann | | |
| **Abhängigkeiten** | FA016: Vorherige Version anzeigen | | |
| **Weiterführende Informationen** | Eine Version zurück ist ausreichend. | | | |

FA0: Version wiederherstellen

|  |  |  |  |
| --- | --- | --- | --- |
| **Anforderungsnr.** | FA018 | **Use Cases** | UC03 |
| **Name** | Dateien herunterladen | | |
| **Beschreibung** | Als Benutzer muss ich, an Artikel angehängte, Dokumente herunterladen können um diese lokal zu speichern und bearbeiten zu können. | | |
| **Begründung** | Das Teilen von Informationen erfolgt auch über bestehende Dateien. | | |
| **Fit Kriterium** | Die heruntergeladene Datei befindet sich auf der Festplatte des Benutzer PC’s. | | |
| **Priorität** | muss | | |
| **Abhängigkeiten** | FA04: Artikel anzeigen | | |
| **Weiterführende Informationen** | Diese Funktionalität ist von den Anwendern durch Sharepoint bekannt. | | |

FA0: Dateien herunterladen

|  |  |  |  |
| --- | --- | --- | --- |
| **Anforderungsnr.** | FA019 | **Use Cases** | UC01 |
| **Name** | Artikel Vorlagen | | |
| **Beschreibung** | Als Benutzer möchte ich beim Verfassen von Artikeln Formatvorlagen verwenden können, um Standardisierungen bei der Artikelerstellung einfacher etablieren zu können. | | |
| **Begründung** | Vorlagen sollen dem Autor des Artikels das Verfassen erleichtern und sich wiederholende Strukturen müssen so nicht ständig manuell verfasst werden. | | |
| **Fit Kriterium** | Es existiert eine Liste mit Vorlagen im Edit-View. | | |
| **Priorität** | Nice-to-have | | |
| **Abhängigkeiten** | FA01: Artikel erstellen  FA02: Artikel bearbeiten | | |

FA0: Artikel Vorlagen

|  |  |  |  |
| --- | --- | --- | --- |
| **Anforderungsnr.** | FA020 | **Use Cases** | UC01 |
| **Name** | Autor- Daten übernehmen | | |
| **Beschreibung** | Als Benutzer möchte ich meinen Namen und meine Email Adresse nicht erneut eingeben müssen, falls ich diese Bereits im System bereits eingegeben habe. | | |
| **Fit Kriterium** | Beim Anlegen eines zweiten Artikels werden die Felder „Autor“ und „Email“ bereits mit den Benutzerdaten ausgefüllt angezeigt. | | |
| **Priorität** | Nice-to-have | | |
| **Abhängigkeiten** | FA01: Artikel erstellen | | |
| **Weiterführende Informationen** | Realisiert kann dies auch über ein Cookie, wird der Cookie gelöscht, muss der Benutzer diese Daten erneut eingeben. | | |

FA0: Autor- Daten übernehmen

|  |  |  |  |
| --- | --- | --- | --- |
| **Anforderungsnr.** | FA021 | **Use Cases** | UC06 |
| **Name** | Phonetische Suche | | |
| **Beschreibung** | Als Benutzer möchte ich auch Suchergebnisse zu einem Suchterm erhalten, wenn ich mich vertippt habe. | | |
| **Fit Kriterium** | Das System liefert auch für ähnlich klingende Wörter Suchbegriffe. Zum Beispiel werden die gleichen Suchergebnisse zum Thema „Datenbank“ angezeigt, auch wenn man „Dtaenbank“ schreibt. | | |
| **Priorität** | Nice-to-have | | |
| **Abhängigkeiten** | FA06: Artikel/Dokument durchsuchen | | |

FA0: Phonetische Suche

|  |  |  |  |
| --- | --- | --- | --- |
| **Anforderungsnr.** | FA022 | **Use Cases** | UC07 |
| **Name** | Reindizierung anstoßen | | |
| **Beschreibung** | Als Admin möchte ich die Möglichkeit haben die Reindizierung manuell anzustoßen. | | |
| **Begründung** |  | | |
| **Fit Kriterium** | Über die Anbindung des Webinterfaces der Suchengine kann der Admin eine Reindizierung manuell anstoßen. Anschließend führt das System die Reindizierung durch. | | |
| **Priorität** | muss | | |
| **Abhängigkeiten** | FA06: Artikel/Dokument suchen | | |

FA0: Reindizierung anstoßen

|  |  |  |  |
| --- | --- | --- | --- |
| **Anforderungsnr.** | FA023 | **Use Cases** | UC02; UC03 |
| **Name** | Alert-Button konfigurieren | | |
| **Beschreibung** | Als Administrator möchte ich die Möglichkeit haben, den Alert-Button auf globaler Ebene deaktivieren zu können. | | |
| **Fit Kriterium** | Der Alert-Button verfügt über zwei Zustände: Ist der Button durch die Konfiguration aktiviert, befinden sich auf der „Detail-View“ zu jedem Artikel ein „Alert“-Button. Ist der Button durch die Konfiguration deaktiviert, ist dieser Button auf der Detail-View nicht zu finden. | | |
| **Priorität** | Nice-to-have | | |
| **Abhängigkeiten** | FA06: Artikel/Dokument suchen | | |

FA0: Alert-Button konfigurieren

# Nicht funktionale Anforderungen

## Übersicht

|  |  |  |
| --- | --- | --- |
| **Nr.** | **Name** | **Priorität** |
|  |  | Neutral |
|  |  | Wichtig |
|  | Bedienbarkeit Einarbeitungszeit | Wichtig |
|  | Bedienbarkeit Schnelle Artikelverfassung | Sehr Wichtig |
|  | Portierbarkeit Browserunterstützung | Wichtig |
|  | Portierbarkeit iPhone-Unterstützung | Weniger Wichtig |
|  | Skalierbarkeit Clientzugriffe | Wichtig |
|  | Skalierbarkeit Anzahl Dokumente | Neutral |
|  | Änderbarkeit Suchmaschine | Wichtig |
|  | Erweiterbarkeit Funktionalitäten | Sehr Wichtig |

Übersicht : Nicht Funktionale Anforderungen

## Performance

Die Suchzeit muss angemessen sein. Der Benutzer soll nicht das Gefühl haben, dass die Suche zu lange dauert, denn eine lange Suchzeit erhöht die Antwortzeit des Systems. Dabei muss das System innerhalb von einer Sekunde (serverseitiger) Suchzeit das Suchresultat liefern können, um die Arbeitsabläufe zu verbessern und nicht auszubremsen.

## Zuverlässigkeit

Um einen möglichst hohen Mehrwert aus der Software ziehen zu können und den Arbeitsablauf zu verbessern, darf die Software nur, dem eingegebenen Suchterm entsprechend, relevante Suchergebnisse auflisten.

## Bedienbarkeit

Als Benutzer möchte ich einen direkten Mehrwert aus der Software ziehen, um meine Arbeitsabläufe zu beschleunigen, zu vereinfachen und/oder mehr Effizienz zu erlangen. Dabei muss das System nach einer Einarbeitungszeit von maximal 15 Minuten bedienbar sein. Darüber hinaus darf es maximal 5 Minuten dauern, einen Artikel mit ca. 100 Worten Inhalt zu erstellen.

## Portierbarkeit

Um den Einsatzmöglichkeiten zu erweitern und die Gebrauchsflexibilität der Software zu erhöhen, muss es auf mobilen Geräten (iPhone 5 und iPhone 6) möglich sein Artikel zu lesen. Außerdem muss die Software mit der Version 51.0.2704.63 von Google Chrome, sowie der Version 46.0.1 von Mozilla Firefox, vollumfänglich kompatibel sein.

## Skalierbarkeit

Da die Software firmenintern und abteilungsübergreifend eingesetzt werden soll, muss das System, die unter 6.2 genannten Kriterien auch einhalten, wenn es zu einer Spitze von bis zu 200 lesende und 25 schreibende Zugriffe parallel auf das System kommt. Darüber hinaus muss das System mit bis zu 1000 verschiedenen Dokumenten ohne Leistungseinbrüche arbeiten können.

## Änderbarkeit

Die verwendete Suchengine OSS darf nur über wenige Berührungspunkte, sogenannte Schnittstellen

## Erweiterbarkeit

* Spätere Erweiterbarkeit der Funktionalität durch den Kunden

Das System sollte so aufgebaut sein, dass sie später um weitere Funktionen, wie beispielsweise noch nicht implementierte Nice-to-Have Anforderungen, erweitert werden kann.

## Qualitätsmatrix

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | **Wichtigkeit** | | | |
| **Sehr wichtig** | **Wichtig** | **Neutral** | **Weniger Wichtig** |
| **Qualitätsattribut** | **Performance** |  |  | X |  |
| **Zuverlässigkeit** |  | X |  |  |
| **Bedienbarkeit** | X |  |  |  |
| **Portierbarkeit** |  |  |  | X |
| **Skalierbarkeit** |  |  | X |  |
| **Testbarkeit** |  | X |  |  |
| **Änderbarkeit** | X |  |  |  |
| **Erweiterbarkeit** | X |  |  |  |

Übersicht : Qualitätsmatrix

Die Qualitätsmatrix gibt einen Überblick über die Wichtigkeit der oben ausgeführten Qualitätsattribute. Diese wurden an Hand der Interviews mit dem Vertreter der Auftraggeberfirma NTT Data Deutschland erhoben. Dabei wurde auf die Betonung oder Mehrfachnennung einzelner Attribute geachtet und durch detailliertes Nachfragen konnten ebenfalls bedeutende Qualitätsanforderungen ermittelt werden. Dadurch konnten einige Anforderungen als wichtiger als andere identifiziert werden. Die Matrix oben soll einen Überblick geben welche Wichtigkeit die erhobenen Qualitätsanforderungen besitzen.

# Glossar

|  |  |
| --- | --- |
| **Artikel**  **Indexierung**  **Reindizierung**  **Sicherheitsabfrage** | Ein Artikel kann aus Text, Grafiken und Tabellen bestehen. Er dient dazu firmeninternes Wissen zu kommunizieren. An einen Artikel können Dokumente angehängt werden.  Die Indexierung ist ein notwendiger, technischer Prozess, der dem Inhalt einer Datenbank einen Index zuweist. Dieser wird bei der Suche und beim Sortieren des Inhaltes der Datenbank benötigt.  Dabei handelt es sich um eine einfache Pop-up Meldung, in der der Benutzer seine Aktion bestätigen muss. Zu bestätigende Aktionen sind beispielsweise das Schließen des Bearbeitungsfensters vor dem Speichern der Änderungen. |

# Anhang

MockUps

start

search

detailed search

new

newchancedpage

saved

savedemptyfields

alert

detail

edit

editchacedpage

deletearticle