



PFLICHTENHEFT

„Entwicklung eines Suchprogramms für ein digitales Zeitungsarchiv“

Stand: 22. Mai 2019

Auftraggeber: **Suluhobekra Flugschriften
GmbH**

Frau Teresa Bergmann
Druckereistraße 1
01099 Dresden

Tel.: 0351 / 5553325
E-Mail: bergmann@suluhobekra.de



Auftragnehmer: **SILICON SAXONY
ENGINEERING GmbH**

i.V. Herr Fabian Hovestädt
Manfred-von-Ardenne-Ring 20
01099 Dresden

Tel.: 0351 / 2011907
E-Mail: info@silicon-saxony-eng.de
Internet: www.sse-dresden.to





Inhaltsverzeichnis

1. Zielbestimmung
 - 1.1 Muss-Kriterien
 - 1.2 Kann Kriterien
 - 1.3 Abgrenzungskriterien
2. Produkteinsatz
 - 2.1 Anwendungsbereich
 - 2.2 Zielgruppe
 - 2.3 Produktumgebung
 - 2.3.1 Orgware
 - 2.3.2 Hardware
 - 2.3.3 Software
 - 2.3.4 Schnittstellen
 - 2.4. Betriebsbedingungen
3. Produktanforderungen
 - 3.1 Funktionale Anforderungen
 - 3.1.1 Beschreibung der FA mit Rollen innerhalb der Geschäftsprozesse
 - 3.1.2 Aktivitäten mit Benutzerschnittstelle
 - 3.1.3 Fachliches Klassendiagramm
 - 3.2 Nichtfunktionale Anforderungen
 - 3.2.1 Benutzbarkeit
 - 3.2.2 Zuverlässigkeit
 - 3.2.3 Effizienz
 - 3.2.4 Softwarewartung
 - 3.2.5 Sicherheit
 - 3.2.6 Korrektheit
4. Testung
5. Monitoring/Support bei Übergabe oder ähnliche Leistungen
6. Dokumentation
 - 6.1 Anwenderdokumentation
 - 6.2 Entwicklerdokumentation
 - 6.3 Testdokumentation
7. Vorgehen
8. Glossar



1. Zielbestimmungen

Es soll ein Suchprogramm für ein digitales Zeitungsarchiv entwickelt werden, um Artikel durchsuchen zu können. Die Artikel können entweder in Form eines Archivs oder als **Datei** vorliegen. Nach Eingabe des Dateinamens und des Suchbegriffs werden die betroffenen Textzeilen, die Anzahl der Treffer und der Name der Textdatei ausgegeben. Im graphischen Menü soll der Benutzer die Sprache auswählen und das Programm via Exit Schaltfläche verlassen können.

1.1. Muss-Kriterien

MK-IO-01	Output	Es müssen Textzeilen, in denen der Suchbegriff vorkommt, ausgegeben werden, als auch die Anzahl der Treffer und den Namen der Textdatei. Die Ausgabe muss auf einer graphischen Oberfläche erfolgen.
MK-IO-02	Input	Es muss ein Archiv oder eine Textdatei in das graphische Suchfeld eingegeben werden. Des Weiteren muss der Suchbegriff eingegeben und mit einem Mausklick auf eine Suchschaltfläche bestätigt werden.
MK-BS-01	Exit	An das Betriebssystem muss das Programm den Exit-Code 0 (Null) zurückgeben.
MK-SYS-01	OO – Analyse	Das System muss objektorientiert analysiert werden.
MK-SYS-02	UML2	Für die Modellierung muss UML2 verwendet werden.
MK-IMPL-01	Java	Die Programmierung muss in Java erfolgen
MK-IMPL-02	Code Style	Der Code muss sauber und strukturiert sein. Es sind selbsterklärende Bezeichnungen für Funktionen und Variablen zu verwenden. Außerdem müssen erklärende Kommentare angefügt werden.
MK-GUI-01	User Interface	Das Menü muss einen Exit-Knopf, ein Textfeld für die Eingabe und ein Feld für die Anzeige der Ergebnisse aufweisen. Für das Anlegen des Menüs ist ein GridBagLayout Panel zu verwenden.



1.2 Kann-Kriterien

KK-Lang-01	Sprache	Der Anwender sollte zwischen verschiedenen Menüsprachen wählen können. Zu implementieren sind: Deutsch und Englisch.
------------	---------	--

1.2. Abgrenzungskriterien

AK-Kons-01	Konsolenanwendung	Das System kann nicht auf der Konsole ausgeführt werden.
AK-Stat-01	Statusmeldungen	Es sollen keine Statusmeldungen vor der Ausgabe des Ergebnisses ausgegeben werden.
AK-Int-01	Onlinefunktionalität	Das System soll nicht auf Dateien aus dem Internet zugreifen können. Sie müssen sich zuvor im Speicher befinden.



2. Einsatz

2.1 Anwendungsbereich

Die SULUsearch-Anwendung wird dazu benutzt um Textdateien und **Archive** zu durchsuchen. Die SULUHOBEKRA Flugschriften GmbH kann das Programm zur Recherche in älteren Zeitungsartikeln verwenden und dient als Hilfestellung für das Verfassen von neuen Beiträgen.

2.2 Zielgruppen

Das Programm wird von Mitarbeitern der SULUHOBEKRA Flugschriften GmbH benutzt. Diese treten lediglich in der Rolle des Hauptbenutzers auf.

2.3 Produktumgebung

2.3.1 **Orgware**

Um die Projektziele zu erreichen, wird das Programmierwerkzeug **Topcased** für die Modellierung verwendet.

2.3.2. Hardware

Der Zugriff auf die Festplatte ist zwingend notwendig. Es bestehen keine spezifischen Hardwareanforderungen.

2.3.3. Software

Das System benötigt ein entsprechendes **Java Runtime Environment**, um Programme ausführen zu können. Um das **GUI** zu realisieren wird das **AWT** Packet von Java verwendet. Zur Entwicklung des **Quellcodes** wird das Entwicklungswerkzeug **Eclipse** verwendet.

2.3.4. Schnittstellen

Als Schnittstelle ist die Arbeit mit externen Textdateien und Textarchiven vom Massenspeicher vorgesehen.

2.4 Betriebsbedingungen

Das System soll auf den Rechnern der SULUHOBEKRA Flugschriften GmbH benutzt werden. Ein beaufsichtigter Betrieb ist nicht notwendig. Auch ein bestimmtes Betriebssystem ist für die Anwendung nicht notwendig, jedoch wird ein Java Runtime Environment benötigt. Das Programm muss in der Konsole gestartet werden. Die Rechner sollten regelmäßig mit aktueller **Hardware** erneuert werden und eine ununterbrochene Stromversorgung besitzen. Die Betriebsräume müssen klimatisiert sein und eine geringe Staubbelastung vorweisen.



3. Produktfunktionen/Anforderungen

3.1 Funktionale Anforderungen

3.1.1 Beschreibung der FA mit Rollen innerhalb der Geschäftsprozesse

Die Mitarbeiter führen das Programm an ihrem Arbeitsplatz aus. Es hat eine benutzerfreundliche grafische Oberfläche.

AF-01	Sprachfunktion	In dem Menü kann der Benutzer zwischen Deutsch und Englisch entscheiden. Daraufhin werden die Schaltflächen bezeichnungen in die jeweilige Sprache übersetzt.
--------------	----------------	--

AF-02	Suchfunktion	Der Benutzer gibt in einem Textfeld einen Textdateipfad oder Archivpfad ein. In einem weiteren Textfeld wird das Stichwort eingegeben. Ausgegeben werden die Zeilen in der sich das Stichwort befindet, der Name der Textdatei, sowie die absolute Häufigkeit. Über eine Schaltfläche kann das Programm verlassen werden. Das Programm gibt den Exit-Code 0 zurück.
--------------	--------------	--



3.1.2 Aktivitäten mit Benutzerschnittstelle (UI)

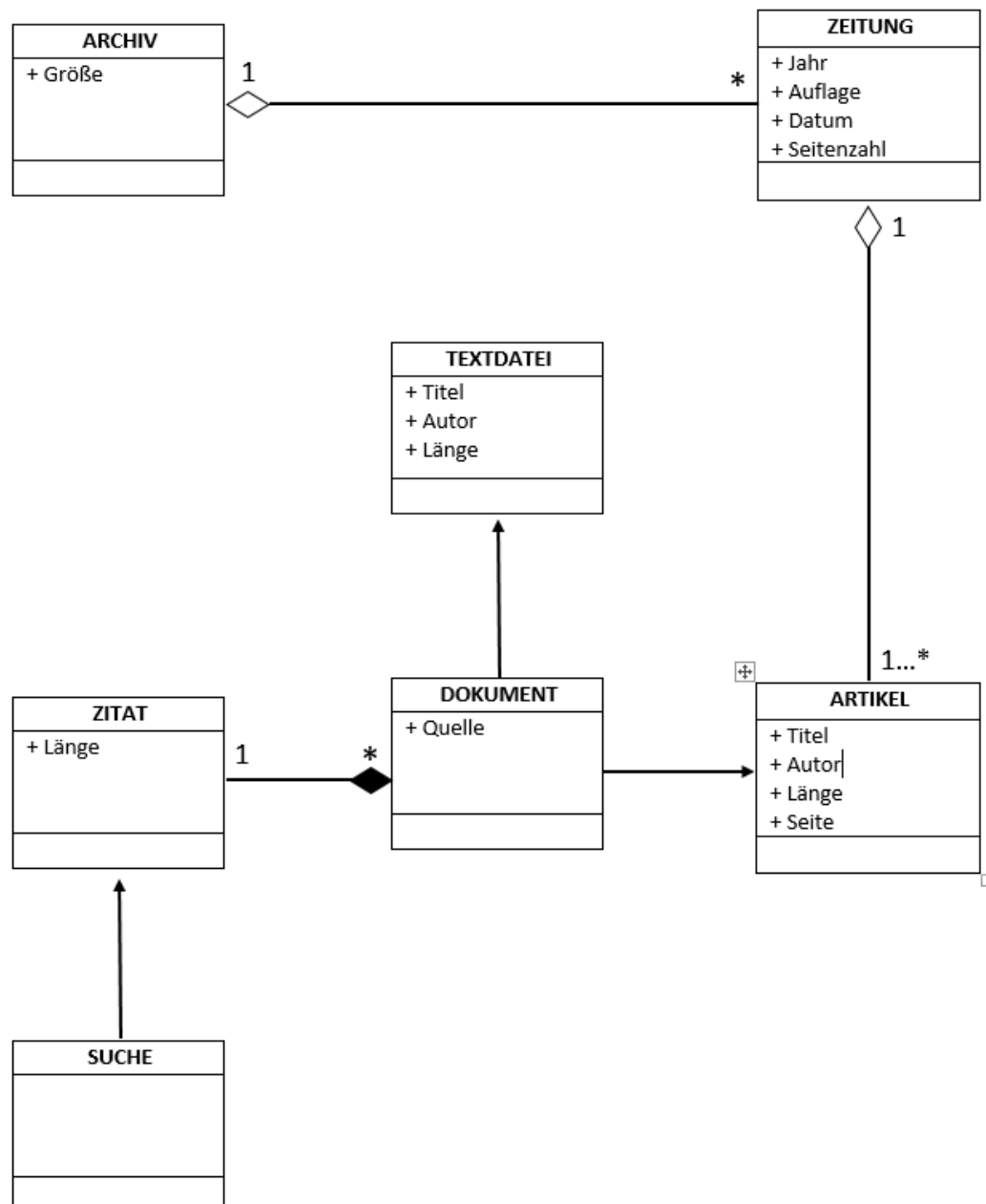
Anwendungsfall ID	AF-01
AF Name	Sprachfunktion
Akteur	angemeldeter Nutzer
Vorbedingung	Computer eingeschaltet & Programm gestartet
Auslösendes Ereignis	Betätigung der Deutsch - oder Englischschaltfläche
Nachbedingung Erfolg	Die Schaltflächenbezeichnungen werden in die jeweilige Sprache übersetzt.
Nachbedingung Fehlschlag	/
Ablauf	<ul style="list-style-type: none"> - Start des Programms durch Ausführung der „suluSearch.jar“-Datei - Klicken der jeweiligen Sprachschaltfläche - Sprache wird auf Englisch oder Deutsch geändert
Benutzerschnittstelle	(Sprachschaltfläche)

Anwendungsfall ID	AF-02
AF Name	Suchfunktion
Akteur	angemeldeter Nutzer
Vorbedingung	Computer eingeschaltet & Programm gestartet
Auslösendes Ereignis	Bestätigung des Suchvorgangs mit einem Mausklick auf eine Suchschaltfläche
Nachbedingung Erfolg	Es werden die Zeilen in der sich das Stichwort befindet, der Name der Textdatei, sowie die absolute Häufigkeit ausgegeben.
Nachbedingung Fehlschlag	Textdateipfad oder Archivpfad konnte nicht gefunden werden. Ausgabe der Fehlermeldung.
Ablauf	<ul style="list-style-type: none"> - Start des Programms durch Ausführung der „suluSearch.jar“-Datei - Textdateipfad oder Archivpfad, sowie das Stichwort in Suchfeld eingeben - Suchschaltfläche per Mausklick betätigen - Ausgabe der Zeilen in der sich das Stichwort befindet, der Name der Textdatei, sowie die absolute Häufigkeit
Benutzerschnittstelle	(Suchfeld)



3.1.3 Fachliches Klassendiagramm (domain model) / Produktdaten

Für die suluSearch-Anwendung ist dauerhaft ein Archiv gespeichert, welches während der Laufzeit durchsucht wird.





3.2 Nichtfunktionale Anforderungen

3.2.1 Benutzbarkeit

Es handelt sich um ein einfaches Programm mit wenig Funktionen. Bedient wird das Programm über ein benutzerfreundliches Menü. Die Schaltflächen sind selbsterklärend. Selbst für computerfremde Personen ist das Programm leicht zu erlernen.

3.2.2 Zuverlässigkeit

Bei der Eingabe eines nichtexistierenden Datei - oder Archivpfades wird eine dementsprechende Fehlermeldung ausgegeben. Sollte das Suchwort nicht in dem angegebenen Pfad existieren bricht das Programm nicht ab, sondern gibt aus, dass es keine erfolgreichen Treffer gab. Das Programm besitzt keine Fehlertoleranz bei falscher Rechtschreibung.

3.2.3 Effizienz

Für einzelne Textdateien ist das Zeitverhalten konstant effizient. Bei der Suche in größeren Archiven kann sich die **Laufzeit** stark verlängern.

3.2.4 Softwarewartung

Für die Softwarewartung übernimmt Silicon Saxony Engineering GmbH keine Verantwortung.

3.2.5 Sicherheit

Im Softwaresystem werden keine besonderen Sicherheitsvorkehrungen **implementiert**.

3.2.6 Korrektheit

Sollte sich der Suchbegriff als Teilbestand innerhalb eines Wortes befinden, wird dies vom Programm erkannt und es wird nicht als Trefferergebnis akzeptiert. Damit wird die Korrektheit der Suchergebnisse sichergestellt.

4 Testung

Für die **Anwendungsfälle** AF-01 und AF-02 werden Funktionstests durchgeführt. Für den Anwendungsfall AF-01 wird die Menüsprache von Deutsch auf Englisch und wieder zurück geändert. Für den Anwendungsfall AF-02 wird in einem Beispielarchiv und in einer Beispieldatei ein Stichwort gesucht. Die Tests gelten als bestanden, wenn die Menüsprache erfolgreich geändert wurde und bei der Beispielsuche die erwartete Anzahl von Treffern ausgegeben wird. Die Tests müssen bei der Übergabe (siehe 5.) bestanden werden.



5 Support bei Übergabe

Der Support bei der Übergabe umfasst eine Schulung der Mitarbeiter der SULUHOBEKRA Flugschriften GmbH und wird in Rechnung gestellt. Es ist vorgesehen die Tests bei der Übergabe durchzuführen und dabei gleichzeitig das Programm zu erklären.

6 Dokumentation

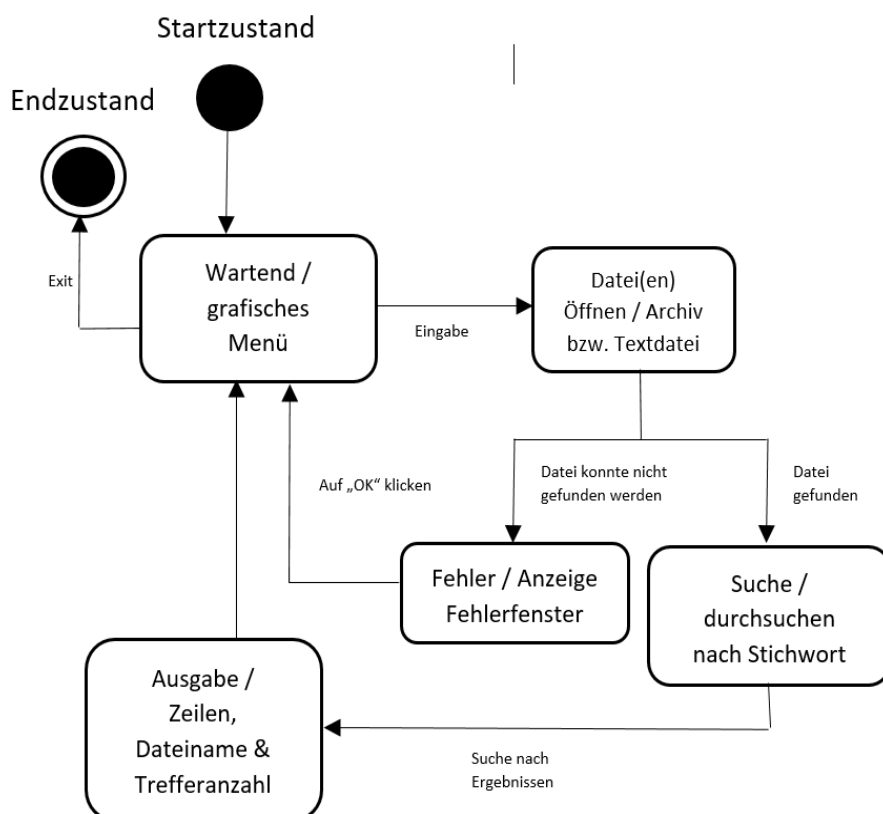
6.1 Anwenderdokumentation

Für die Anwenderdokumentation ist eine PDF-Datei im GitHub-Repository innerhalb des „final“ Ordners hinterlegt.

6.2 Entwicklerdokumentation

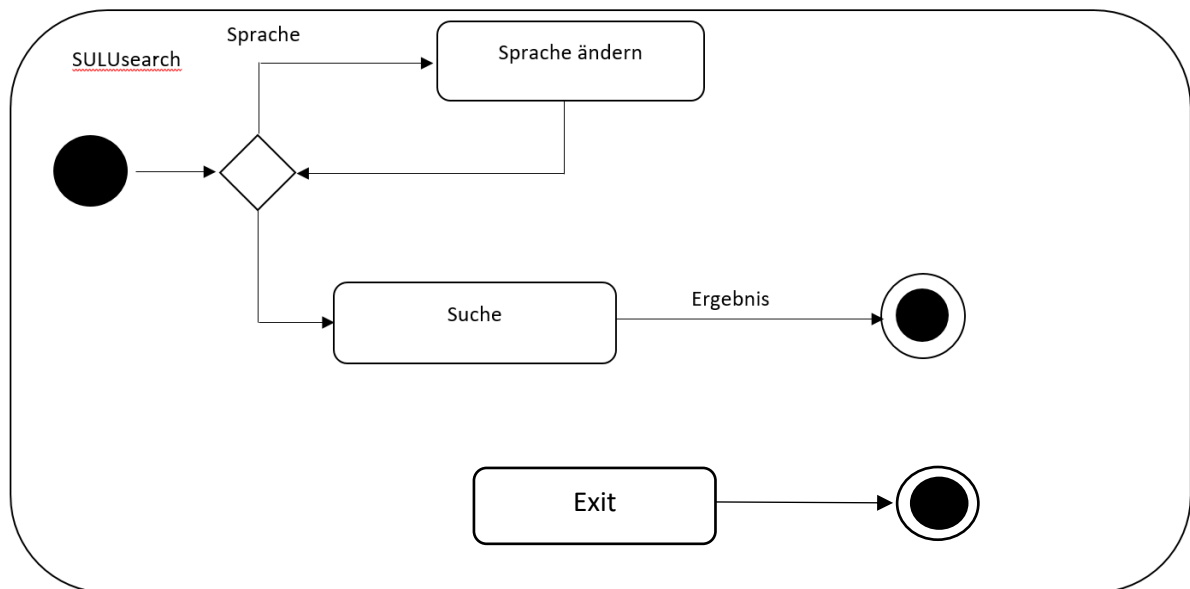
Für die Entwicklerdokumentation ist eine Javadoc-Datei im GitHub-Repository innerhalb des „final“ Ordners hinterlegt.

Zustandsdiagramm:

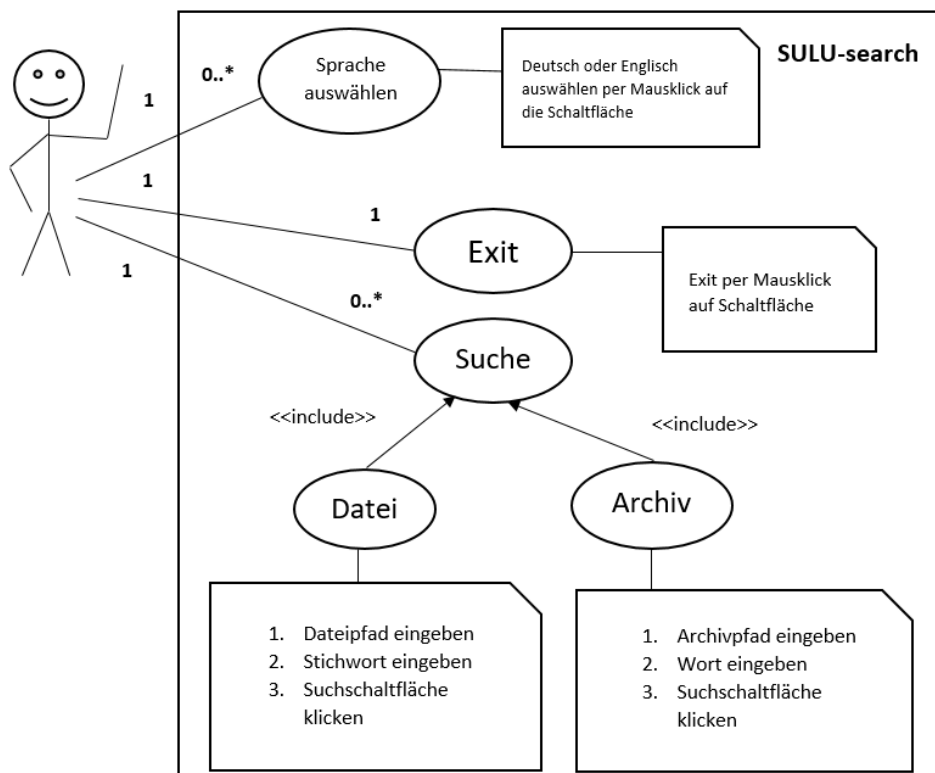




Aktivitätsdiagramm:



Anwendungsfalldiagramm:





6.3 Testdokumentation

Für die Testdokumentation ist eine PDF-Datei im GitHub-Repository innerhalb des „final“ Ordners hinterlegt.

7 Vorgehensweise

Der Start des Projektes beginnt mit der Erstellung des Pflichtenheftes. Die Fertigstellung des Modells erfolgt mit UML2. Auf Basis von Produktfunktionen und Anforderungen wird der Prototyp erstellt und die Implementierung erfolgt. Im Anschluss darauf wird das Programm erstmals getestet und gilt somit als Release Candidate. Letzten Endes erfolgt die Übergabe.

Meilensteine:

Datum	Meilenstein
10.01.2019	Start des Projekts
11.01.2019	Erstellen des Pflichtenhefts
10.04.2019	Fertigstellung des Modells mit einer Entwicklungsumgebung
25.04.2019	Prototyp
10.05.2019	Testung
15.05.2019	Release Candidate
30.06.2019	Übergabe



Die Fortschrittskontrolle erfolgt anhand folgender Indikatoren:

Indikator		Start des Projekts	Erstellen des Pflichtenheftes	Fertigstellung des Modells mit einer Entwicklungs-umgebung	Prototyp	Testung	Release Candidate	Übergabe
Pflichtenheft [% erledigte Gliederungs-Punkte]	Soll	0%	5%	15%	55%	15%	5%	5%
	Ist	0%	5%					
Umgebung [Tools]	Soll	Word	Word	Topcased-UML2	Topcased – Java	Java Runtime Environment, Konsole	Eclipse	CD-ROM
	Ist	Word	Photo-shop/ Word	Topcased-UML2	Ninja-Mock, Eclipse	Eclipse, JUnit	Eclipse	Git-Repository
Diagramme [Anzahl]	Soll	-	4	1	-	-	-	-
	Ist	-	4		-	-	-	-
Quellcode [LOC]	Soll	-	-	50	350	50	25	-
	Ist	-	-					-
Verhältnis LOC/ Kommentare	Soll	-	-	10:1	7:1	7:1	6:1	-
	Ist	-	-					-
Qualitätsprüfung	Soll	-	Gegenlesen & Zustimmung des Auftraggebers	Modell entspricht den Standards der UML	Rückkopplung mit Auftraggeber	Bestehen der vereinbarten Tests	Erfüllen der Festlegungen im Pflichtenheft	Nach Schulung können Mitarbeiter mit dem Softwaresystem sicher umgehen
	Ist	-	-					
Anwenderdoku [Worte]	Soll	-	-	-	-	-	300	-
	Ist	-	-	-	-	-		-
Entwicklerdoku [Worte]	Soll	-	-	100	100	100	-	-
	Ist	-	-				-	-



8. Glossar

Wort	Erklärung
Datei	Paket zusammengehöriger Informationen auf Speichermedium
Output	Ausgabe
Input	Eingabe
Exit-Code	Rückgabestatus oder auch Rückgabewert
UML2	ist eine grafische Modellierungssprache
GridBagLayout Panel	Fenster für Anordnung von grafischen Elementen
Konsole	Programm zur Eingabe von Befehlen für das System
Archive	Ordner
Orgware	Benutzerhandbücher, Konzepte für IT-Projekte und Sicherheitsanforderungen
Topcased	<i>Werkzeugsammlung für die Entwicklung</i>
Java Runtime Environment	Laufzeitumgebung für Java
GUI	Grafische Benutzeroberfläche
AWT	Paket in Java
Quellcodes	für Menschen lesbarer, in einer Programmiersprache geschriebener Text eines Computerprogrammes
Eclipse	quelloffenes Programmierwerkzeug zur Entwicklung von Software verschiedenster Art
Javadoc	Javadoc ist ein Software-Dokumentationswerkzeug, das aus Java-Quelltexten automatisch HTML-Dokumentationsdateien erstellt.
GitHub	GitHub ist eine freie Software zur verteilten Versionsverwaltung von Dateien, die durch Linus Torvalds initiiert wurde.
Hardware	Oberbegriff für die physischen Komponenten eines datenverarbeitenden Systems, als Komplement zu Software.
Schaltfläche	Dient zur Interaktion zwischen Nutzer und Software
Archivpfad	Speicherort indem sich der Ordner befindet
Laufzeit	Zeitdauer, die ein Programm zur Bewältigung einer Aufgabe benötigt.
Implementiert	Einbau oder die Umsetzung von festgelegten Strukturen und Prozessabläufen in einem System unter Berücksichtigung der Rahmenbedingungen, Regeln und Zielvorgaben, so im Sinne einer Spezifikation
Anwendungsfälle	bündelt alle möglichen Szenarien, die eintreten können
Modell	Repräsentation eines Systems von Objekten, Beziehungen und/oder Abläufen.
Prototyp	Vorläufiges Produkt
Release Candidate	Fertigstellzustand