

**Requirements Specification**

Semesterprojekt SS 2017

Archivist

Version 1

# Inhaltsverzeichnis

Inhaltsverzeichnis 2

1 Einleitung 4

1.1 Zweck des Dokuments 4

1.2 Gültigkeit des Dokuments 4

1.3 Begriffsbestimmungen und Abkürzungen 4

1.4 Zusammenhang mit anderen Dokumenten 4

1.5 Überblick über das Dokument 4

2 Allgemeine Beschreibung des Produkts 5

2.1 Zusammenhang mit bereits laufenden Projekten 5

2.2 Zusammenhang mit Vorgänger- und Nachfolgeprojekten 5

2.3 Zweck des Produkts 5

2.4 Abgrenzung und Einbettung des Produkts 5

2.5 Überblick über die geforderte Funktionalität 5

2.6 Allgemeine Einschränkungen 5

2.7 Vorgaben zu Hardware und Software 5

2.8 Benutzer des Produkts 6

3 Detaillierte Beschreibung der geforderten Produktmerkmale 7

3.1 Lieferumfang 7

3.2 Abläufe (Szenarien) von Interaktionen mit der Umgebung 7

3.3 Ziele des Benutzers 7

3.4 Geforderte Funktionen des Produkts 7

3.4.1 Funktion Bezeichnung a 7

3.4.2 Funktion Bezeichnung b 8

3.5 Externe Schnittstellen des Produkts 8

3.5.1 Benutzerschnittstellen (User Interfaces) 8

3.5.2 Systemschnittstellen 8

3.6 Sonstige geforderte Produktmerkmale 8

3.6.1 Geschwindigkeitsmerkmale (performance) 8

3.6.2 Ressourcenmerkmale (resources) 9

3.6.3 Schutzmerkmale (security) 9

3.6.4 Sicherheitsmerkmale (safety) 9

3.6.5 Portabilitätsmerkmale (portability) 9

3.6.6 Zuverlässigkeit (reliability) 9

3.6.7 Wartungsmerkmale (maintenance) 9

3.6.8 Wiederverwendbarkeitsmerkmale (reuse) 9

3.6.9 Benutzbarkeitsmerkmale (usability) 9

4 Vorgaben an die Projektabwicklung 10

4.1 Anforderungen an die Realisierung 10

4.1.1 Hardware 10

4.1.2 Software 10

4.2 Fertige und zugekaufte Komponenten 11

4.3 Unterauftragnehmer 11

4.4 Abnahmebedingungen 11

4.5 Lieferbedingungen 12

4.6 Gewährleistung 12

5 Verpflichtungen des Auftraggebers 13

6 Literaturverweise 14

7 Anhang 15

# Einleitung

## Zweck des Dokuments

## Gültigkeit des Dokuments

Für welchen Bereich ist das Pflichtenheft gültig, gilt es für Gesamt- oder Teilprojekt, setzt es auf ein bestehendes Pflichtenheft auf? Wer ist für Änderungen im Pflichtenheft zuständig?

## Begriffsbestimmungen und Abkürzungen

Auflistung von Definitionen und Abkürzungen, auch Begriffe aus der Domäne.

## Zusammenhang mit anderen Dokumenten

Liegt ein Angebot vor, dann darf das Pflichtenheft den Aussagen im Angebot nicht widersprechen. Liegt ein Lastenheft vor, dann muss sich das Pflichtenheft auf dieses beziehen, um die Verfolgbarkeit der Anforderungen zu ermöglichen.

## Überblick über das Dokument

Welchen Inhalt hat der Rest des Pflichtenhefts, wie ist das Pflichtenheft aufgebaut?

# Allgemeine Beschreibung des Produkts

Diese allgemeine Beschreibung soll ein „Management Summary“ sein, also noch keine spezifischen Produktmerkmale beschreiben.

## Zusammenhang mit bereits laufenden Projekten

Falls es Zusammenhänge mit bereits laufenden Projekten gibt, werden diese hier kurz dargestellt.

## Zusammenhang mit Vorgänger- und Nachfolgeprojekten

## Zweck des Produkts

Welcher Zweck wird mit dem Produkt verfolgt, was sind die wichtigsten Merkmale und die entscheidenden Vorteile gegenüber der bisherigen Situation?

## Abgrenzung und Einbettung des Produkts

Festlegen der Systemgrenzen. Welche wesentlichen Leistungen sollen umfasst werden, was gehört nicht dazu, sondern zur Umgebung? Blockdiagramm, das zeigt, wie das zu erstellende Produkt mit anderen Systemen in Beziehung steht.

## Überblick über die geforderte Funktionalität

Hier soll nur ein Überblick vermittelt werden – die detaillierte Beschreibung folgt in Kapitel 3

## Allgemeine Einschränkungen

Vorgaben hinsichtlich Schnittstellen, Standards, Methoden, die eingehalten werden müssen.

## Vorgaben zu Hardware und Software

In welcher HW- und SW- Umgebung soll das Zielsystem laufen, wo soll es entwickelt werden (z. B. Betriebssysteme, Entwicklungstools)

## Benutzer des Produkts

Wer wird das Produkt nutzen (Ausbildung und Vorkenntnisse, produktive Nutzer und Systemadministratoren; tägliche und fallweise Benutzer)

# Detaillierte Beschreibung der geforderten Produktmerkmale

Hier werden die vom Auftraggeber geforderten und von der Entwicklung zugesagten Produkteigenschaften definiert.

Jedes geforderte Merkmal sollte

* eindeutig bezeichnet sein,
* bei Bedarf mit einer Priorität versehen werden und
* die Herkunft der Anforderung zeigen (auch Bezüge zum Lastenheft).

Die Beschreibung sollte so sein, dass

* möglichst geringer Interpretationsspielraum bleibt,
* eine Überprüfbarkeit gegenüber dem fertig gestellten Produkt gegeben ist.

## Lieferumfang

Produkt und dessen Teilkomponenten, Schulungsunterlagen, Handbücher, …

## Abläufe (Szenarien) von Interaktionen mit der Umgebung

Use Cases. Können oft schon vom Lastenheft übernommen werden.

* Verwenden der Funktionen in 3.4.x
* Jede Funktion aus 3.4.x sollte in mindestens einem Ablauf vorkommen

## Ziele des Benutzers

Beschreiben aller Ziele, die der Benutzer durch die Szenarien erreichen kann.

## Geforderte Funktionen des Produkts

Was soll das zu erstellende System leisten?

### Funktion Bezeichnung a

Jede Funktion muss eine eindeutige Bezeichnung haben, die im Lebenslauf des Projekts wieder gefunden werden soll.

#### Wirkungsweise von Funktion a

#### Abhängigkeiten / Randbedingungen

Alle Abhängigkeiten bzw. Zusammenhänge mit anderen Funktionen müssen hier festgehalten werden. Damit soll erreicht werden, dass bei Änderungen einer hier beschriebenen Funktion auch diese abhängigen Funktionen gezielt betrachtet werden können.

Gibt es Randbedingungen (z. B. bezüglich der Geschwindigkeit), dann müssen diese unten bei den sonstigen Produktmerkmalen beschrieben und von hier aus referenziert werden (z. B. mit einem Verweise auf das entsprechende Geschwindigkeitsmerkmal).

### Funktion Bezeichnung b

## Externe Schnittstellen des Produkts

Das Verhalten im Sinne einer Interaktion des zu erstellenden Produkts sollte bereits oben in Form von Szenarien beschrieben sein. Hier geht es um die Beschreibung, wie die konkreten Schnittstellen aussehen und funktionieren.

### Benutzerschnittstellen (User Interfaces)

Wie sieht das GUI aus, Verweise auf Prototypen.

### Systemschnittstellen

Beschreibung der SW- und HW-Schnittstellen, mit denen das zu erstellende System kommuniziert. Beschreibung der Kommunikationsart, des Datenformats. Verfeinerung des Überblicks in Kapitel 2.

#### Schnittstelle a

* Syntax/Semantik
* Datenformat
* Protokolle
* Datenraten

#### Schnittstelle b

## Sonstige geforderte Produktmerkmale

Beschreiben der nicht funktionalen Anforderungen.

### Geschwindigkeitsmerkmale (performance)

* Reaktionszeiten, Antwortzeiten
* Anlaufzeiten
* Durchsatzrate
* Belegungsdauer

### Ressourcenmerkmale (resources)

* Datenmengen
* CPU-Bedarf
* CPU-Auslastung
* Speicher
* Peripheriegeräte
* Ausgabemenge
* Benötigtes Bedienpersonal

### Schutzmerkmale (security)

Schutz des Produkts gegen Eingriffe von außen (unberechtigter Zugriff, Virenschutz, …)

### Sicherheitsmerkmale (safety)

Unter den Sicherheitsmerkmalen versteht man jene Merkmale, welche die Schadensmöglichkeit nach einem Software- oder Systemausfall begrenzen. Diese Überlegungen können hilfreich sein, kritische Softwareteile zu identifizieren.

### Portabilitätsmerkmale (portability)

### Zuverlässigkeit (reliability)

Maßzahlen z. B. Ausfallszeiten in Minuten / Jahr und MTBF (Mean Time Between Failure)

### Wartungsmerkmale (maintenance)

Angaben zur Wartungsfreundlichkeit, welche Tools können verwendet werden?

### Wiederverwendbarkeitsmerkmale (reuse)

Ist eine spätere Wiederverwendung von Produktteilen gewünscht, werden die Anforderungen daran festgelegt.

### Benutzbarkeitsmerkmale (usability)

Z. B. Angabe, wie viele Bedienungsfehler nach einer vierstündigen Einschulung bei einem Usability-Test noch gemacht werden dürfen.

# Vorgaben an die Projektabwicklung

Im Folgenden werden die Voraussetzungen zur Abwicklung des Projekts hinsichtlich technischer sowie organisatorischer Bedingungen dargestellt.

## Anforderungen an die Realisierung

### Hardware

Die benötigten Hardware-Komponenten zur Durchführung des Projekts beschränken sich auf einen standardmäßigen PC. Zum Einlesen von Notenblättern wird entweder ein Scanner oder eine Kamera (bzw. Smartphone) verwendet oder bestehendes digitales Notenmaterial verwendet.

Zusammengefasst bestehen folgende Hardware-Anforderungen:

* Standard PC
* Scanner (optional) oder
* Smartphone/Kamera (optional)

### Software

#### Betriebssystem

Für das Betriebssystem des **Entwicklungsrechners** wird bevorzugt **Windows 10** verwendet. Auch andere Betriebssysteme erfüllen die Anforderungen für die Entwicklungstools, jedoch wird in diesem Dokument ausschließlich der Umgang mit Windows 10 aufgezeigt; etwaige Abweichungen für andere Systeme werden nicht behandelt.

Zusammenfassend kann gewählt werden zwischen den Betriebssystemen:

* Windows 7/10 (bevorzugt)
* Linux
* Mac

Bezüglich Betriebssystem stimmen die Anforderungen an das **Endsystem** mit jenen des Entwicklungssystems überein.

Es kann wiederum gewählt werden zwischen:

* Windows 7/10 (bevorzugt)
* Linux
* Mac

#### Entwicklungsumgebung, -software und -bibliotheken

Zur Erstellung der Software werden einige Softwaretools benötigt. Alle zur  
**Entwicklung** des Backends und Frontends benötigten Bestandteile sind:

|  |  |
| --- | --- |
| **Spring Tool Suite (STS)**  Bildergebnis für spring tool suite | STS dient hauptsächlich als Entwicklungsumgebung für das Spring Boot Backend.  Zusätzlich kann STS auch für die Frontend-Entwicklung verwendet werden (das Frontend ist eine einfache HTML-Seite, für die jeder beliebige Text-Editor verwendet werden kann).  Nachdem STS eine modifizierte Version von Eclipse ist, werden die den Plattformen Windows, Linux und Mac unterstützt. |
| **Wildfly**  Bildergebnis für wildfly | WildFly ist ein Application Server nach dem Java-EE-Standard (Open-Source).  Im Projekt werden das Spring Boot Backend sowie das Frontend auf einem WildFly-Server gehostet. Alternativ können auch alle Java-EE-fähigen Applikationsserver verwendet werden.  Da WildFly mit Java implementiert wurde, kann dieser plattformunabhängig genutzt werden. |
| **MySQL**  Bildergebnis für mysql | Die Daten des Projekts werden mithilfe des relationalen Datenbankmanagementsystems MySQL (Open-Source) verwaltet. |
| **MySQL Workbench**  Bildergebnis für mysql workbench logo | Als wichtiges Planungswerkzeug wird die MySQL Workbench verwendet. D. h. es wird damit das notwendige Datenmodell (ERD) erstellt und eine neue MySQL-Datenbank daraus erzeugt. |
| **AngularJS**  Bildergebnis für angularjs | Zur Entwicklung der Frontend-Anwendung kommt HTML + JavaScript zum Einsatz. AngularJS ist dabei ein JavaScript-Framework zur Erstellung von Single-Page-Webanwendung. Die gesamte Frontend-Programmlogik wird mithilfe AngularJS entwickelt.  Zusätzlich zu AngularJS wird auch ein AngularJS-Tool **UI-Router** verwendet, das die Navigation zwischen einzelnen Seiten in der AngularJS-App erleichtert. |
| **Bootstrap**  Bildergebnis für bootstrap | Als CSS-Framework für die Frontend-Anwendung dient Bootstrap. Dadurch werden einige Standard-Designs für HTML-Elemente sowie ein Grid-System zur Anordnung von Elementen (auch auf verschiedenen Displaygrößen von Endgeräten) bereitgestellt. |

Die Anforderungen für **Endsysteme** bestehen aus der Installation folgender Komponenten, welche sich zum Teil mit den eben erwähnten Tools für die Entwicklung überschneiden:

* Anforderungen „lokales Backend“
  + MySQL …zum Bereitstellen und Betreiben der Datenbank
  + WildFly …zum Bereitstellen der http-Schnittstelle für die auf  
     WildFly gehostete Frontend-Webapplikation
* Anforderungen Frontend
  + Browser (Unterstützung von AngularJS notwendig)
    - Beispiele:
      * Google Chrome
      * Firefox
      * Safari
      * Microsoft Edge
      * Internet Explorer 9-11

**Anmerkung**: Frontend und Backend werden auf ein und demselben Rechner eingerichtet!

Welche HW, SW, Tools usw. müssen vorhanden sein?

* Hardware
  + Entwicklungsrechner
  + Messgeräte
  + Testanlagen
* Software
  + Betriebssysteme (Host und Target)
  + Compiler, Bibliotheken
  + Entwicklungsumgebung
* Sonstiges
  + Entwicklungsmethode
  + Vertraulichkeitsgrad

## Fertige und zugekaufte Komponenten

Ein Zukauf von Hardware-Komponenten ist für das Projekt nicht notwendig, da das Projektteam bereits Rechner für die Entwicklung als auch Scanner bzw. Kameras für das Einlesen von Notenblättern besitzt.

Da die benötigten Software-Komponenten kostenlos erhältlich sind, wird auch dahingehend kein Zukauf getätigt.

Beschreiben der Komponenten, die zugekauft werden oder schon fertig vorhanden sind.

## Unterauftragnehmer

Folgende Abhängigkeiten von Unterauftragnehmern bestehen:

|  |  |  |  |
| --- | --- | --- | --- |
| **Lieferobjekt** | **Unterauftragnehmer** | **Liefertermin** | **Status** |
| Logo (in unterschiedlichen Ausführung jeweils als Bilddatei) | Birgit Haselmayr | 19.05.2017 | 🗸 |

Beschreiben der Beiträge von Unterauftragnehmern und deren Liefertermine.

## Abnahmebedingungen

* Rahmenbedingungen
  + Wogegen wird abgenommen (Pflichtenheft)?
  + Wie wird abgenommen (gemeinsam beim Kunden, oder Kunde testet)
  + Wer stellt Testdaten bereit? Wann sind diese Daten bereitzustellen?
  + Wo wird abgenommen?
  + Wer unterzeichnet das Abnahmeprotokoll.
* Abnahmekriterien
  + Festlegung des Abnahmetests
  + Wann ist die Abnahme erfolgreich (Restfehlerquote)
  + Werden die nicht funktionalen Anforderungen erfüllt?
* Abnahmeunterlagen
  + Testprotokolle
* Gutachten, Sicherheitsnachweise
  + Sind Gutachten oder Sicherheitsnachweise beizubringen?
* Erfüllung von Vorschriften und Normen
  + Hält das Produkt die vorgeschriebenen Normen und Vorschriften ein?

## Lieferbedingungen

* Wann werden die einzelnen Komponenten geliefert?
* Wie wird geliefert (elektronisch, auf CD, Source Code, nur Executable)?

## Gewährleistung

Vereinbarung über die Gewährleistungsdauer, Umfang der Gewährleistung, das Fehlermeldungsverfahren.

# Verpflichtungen des Auftraggebers

Auflistung, wozu der Auftraggeber im Rahmen des Projekts verpflichtet ist, z. B.

* Bereitstellung von HW
* Bereitstellung von SW
* Zur Verfügung stellen von Dokumentation
* Schulungsmaßnahmen
* Zuständigkeiten und Ansprechpartner bekannt geben
* Testdaten zur Verfügung stellen, Zugang zu Testanlagen ermöglichen
* Vorgehen, falls Änderungen gewünscht sind
* …

# Literaturverweise

# Anhang

Falls erforderlich, können hier Materialien eingefügt werden, die nicht in die hier definierte Gesamtstruktur passen und dennoch zum Pflichtenheft gehören. Beispiele dafür sind Modelle (z. B. SDL, UML).