Archivist

Requirements Specification

Version 1

Boris Fuchs

Paul Schmutz

# Einleitung

## Zweck des Dokuments

## Gültigkeit des Dokuments

Dieses Pflichtenheft ist für das Gesamte Projekt Archivist (Version 1) gültig (eventuell wird das Projekt in späteren Semester erweitert, bzw. fortgeführt). Es setzt auf kein bestehendes Pflichtenheft auf. Für Änderungen ist das Projektteam (Paul Schmutz, Boris Fuchs) zuständig.

## Begriffsbestimmungen und Abkürzungen

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

## Zusammenhang mit anderen Dokumenten

Derzeit liegt weder ein Angebot noch ein Lastenheft vor.

## Überblick über das Dokument

* Allgemeine Beschreibung des Produkts
* Detaillierte Beschreibung der geforderten Produktmerkmale
* Vorgaben an die Projektabwicklung (Hardware, Software, Betriebssystem)
* Verpflichtungen des Auftraggebers
* Literaturverweise
* Anhang

# Allgemeine Beschreibung des Produkts

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

## Zusammenhang mit bereits laufenden Projekten

Dieses Projekt ist vollkommen eigenständig und weist keine Zusammenhänge mit anderen Projekten auf.

## Zusammenhang mit Vorgänger- und Nachfolgeprojekten

## Zweck des Produkts

Zweck des Produktes/Projektes ist es ein digitales Notenarchiv für Kapellen und Musikvereine zur Verfügung zu stellen. Da im Moment in einem Großteil der Musikvereine das Notenarchiv nicht digital gehalten wird, soll hier unser Produkt erhebliche Vorteile bieten, wie zum Beispiel:

* Schnelleren Zugriff auf Daten
* Keine räumlichen Abhängigkeiten
* Bessere Gliederung und Strukturierung möglich
* Permanenter Datenzugriff möglich

## 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

Archivist ist eine digitale Datenbank zur Speicherung von digitalisierten Notenblättern, sowie deren Sortierung in verschiedenste Kategorien.

## Allgemeine Einschränkungen

?????

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

## Vorgaben zu Hardware und Software

Das System ist eine browserabhängige Anwendung und wird vorrangig von einem PC/Notebook benutzt. Fokussiert wird die Nutzung über alle Betriebssysteme, wichtig ist hierbei nur eine aktive Internetanbindung. Eine Erweiterung in Form einer App für Android ist angedacht. Es kommen bei der Umsetzung speziell Angular JS, MySQL,….. zur Anwendung.

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

## Benutzer des Produkts

Das System ist für die Tägliche Nutzung ausgelegt. Der Notenarchivar des Musikvereins sollte vorzugsweise der Systemadministrator sein und auch die Berechtigungen vergeben können. Vor allem aber wird das System leicht und verständlich zu bedienen sein und es jedem Nutzer ermöglichen sich in kürzester Zeit perfekt durch das Interface navigieren zu können. 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

## Anforderungen an die Realisierung

### Hardware

Die benötigten Hardware-Komponenten zur Durchführung des Projekts:

* 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

#### Bibliotheken

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

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

## Unterauftragnehmer

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 (Resfehlerquote)
  + Werden die nicht funktionalen Anforderungen erfüllt?
* Abnahmenunterlagen
  + 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).