|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 05Pflichtenheft | | | | | | | |  | |
| Projekttitel: | | **Projekttitel eingeben**  Alle hellgrau geschriebenen Texte sind Anmerkungen/Ausfüllhilfen. Diese Texte bitte löschen bzw. an das konkrete Projekt anpassen.  Im fertigen Dokument gibt es KEINE hellgrauen Texte mehr (Textfarbe = schwarz)!  Dieses Post-it bitte nach Kenntnisnahme löschen! | | | | | | | |
| Auftraggeber\*in: | | **Vorname Nachname / Unternehmen (eine Person)** | | | | | | | |
| Auftragnehmer\*in: | | **Vorname Nachname Projektleiter\*in / Unternehmen** | | | | | | | |
| Schuljahr: | | | **2021/22** | | Klasse: | **XXX** | | |  |
| **VERSION** | **DATUM** | | | **AUTORIN/AUTOR** | | | **ÄNDERUNG** | | |
| v1.0 | 30.08.2018 | | | Vorname Nachname | | | Beschreibung der Änderung z.B. Erstellung… | | |

Inhalt

1 Vorwort 3

2 Ausgangssituation 3

3 Ziele 3

4 Aufgabenstellung 3

5 Richtlinien 3

6 Allgemeines 3

6.1 Identifikation des Projekts 3

6.2 Leitgedanke 4

6.3 Veränderungsmanagement 4

7 Funktionale Anforderungen 4

7.1 Dummy 1 4

7.2 Dummy 2 4

8 Use Cases 4

8.1 Name und Identifier 4

8.2 Beschreibung des Use Case/Funktion 4

8.3 Anforderungen 4

8.4 Aktoren / Anwender 5

8.5 Voraussetzungen 5

8.6 Ablauf des Use Case 5

8.7 Logisches Ablaufdiagramm 5

8.8 Alternative Abläufe / Ausnahmen / Fehlerfälle 7

8.9 Skizzen des User Interface 7

8.10 Inputdaten 7

8.11 User-Input 7

8.12 Output Daten 7

8.13 Rechte 7

8.14 Layout 7

8.15 Texte 7

9 User Stories 8

9.1 Dummy Bereich 8

9.2 Dummy Bereich 8

10 UML – Diagramme 9

11 Schnittstellen 10

12 Story Board 10

13 Nicht Funktionale Anforderungen 11

13.1 Anwendbarkeit 11

13.2 Zuverlässigkeit 11

13.3 Performanz & Last 11

13.4 Weitere Qualitätskriterien 12

13.5 Test-Funktionalitäten 12

13.6 Logging 12

13.7 Statistik 12

13.8 Navigationsstruktur 13

13.9 Rechtestruktur 13

13.10 Texte 13

13.11 Sitemap 13

13.12 Design 13

13.13 Serverinfrastruktur bzw. Hosting 13

13.14 Domains 13

13.15 Lizenzen & Rechte 13

13.16 Präsentation 14

14 Architektur 15

14.1 Hardware 15

14.2 Software 15

14.3 Datenhaltung 15

15 Deployment & Rollout 16

16 Dokumentation 16

17 Deliverables 17

18 Abgrenzungen 17

# 

# Vorwort

Das vorliegende Template umfasst alle möglichen Bereiche und Eventualitäten eines Pflichtenhefts. Die Überschriften und Kapitel sind somit so anzupassen, dass sie auf das jeweilige Projekt zutreffen. Nicht alle Kapitel werden bei jedem Projekt notwendig sein. Grauer Text dient zur Hilfe und muss durch Text mit der Formatvorlage „Normal“ ersetzt werden.

Zur Verwendung des Pflichtenhefts noch ein paar allgemeine Bemerkungen:

* Folgende Wörter sind zu vermeiden, da diese nicht eindeutig sind:
  + Üblich
  + Alle/jede/sämtliche
  + Manche
  + Passend
  + Wenig/viel
  + Notwendig/erforderlich
* Kapitelüberschriften dürfen nicht gelöscht werden. Falls nicht zutreffend für das Projekt dann dies im Text erwähnen.
* Wenn ein Thema noch nicht geklärt ist, dann dieses Thema deutlich kennzeichnen und in die TO DO Liste aufnehmen.

# Ausgangssituation

Kurze Beschreibung der inhaltlichen Ausgangssituation des Projekts. Was sind die Rahmenbedingungen, was sind Randbedingungen.

# Ziele

Kurze Beschreibung der inhaltlichen Ausgangssituation des Projekts. Was sind die Rahmenbedingungen, was sind Randbedingungen.

# Aufgabenstellung

Kurze epische Beschreibung der Aufgabenstellung in ein paar Sätzen.

# Richtlinien

Welche Dokumente bzw. Richtlinien müssen zusätzlich erfüllt werden?

# Allgemeines

Kurze Beschreibung des Auftrags

## Identifikation des Projekts

Wie wird das Projekt identifiziert? Projektname und Projektkürzel werden definiert.

## Leitgedanke

Was ist der Leitgedanke des Projekts? Der Leitgedanke sollte im gesamten Team verwendet werden und im Hinterkopf stehen.

## Veränderungsmanagement

Veränderungen bzw. Anpassungen des Pflichtenhefts bedürfen der schriftlichen Form und des Einverständnis von Auftraggeber und Auftragnehmerin.

# Funktionale Anforderungen

## Dummy 1

Konkrete Beschreibung der funktionalen Anforderung.

Funktionale Anforderungen beschreiben Funktionalitäten (das heißt, was soll das System tun bzw. können) eines Systems, dessen Daten oder Verhalten.

## Dummy 2

Lorem Ipsum

Lorem Ipsum

Lorem Ipsum

# Use Cases

Einzelnen Funktionen können bzw. sollen als Use Cases beschrieben werden. Manchmal ist es sinnvoll, hier eigene Dateien zu erstellen. Use Cases müssen für alle Rollen im Berechtigungskonzept erstellt werden.

## Name und Identifier

Jeder Use Case hat einen eindeutigen Identifier (z.B. UC 2.01) und Namen.

## Beschreibung des Use Case/Funktion

Beschreibung und eventuell Diagramm des Use Case.

## Anforderungen

Jede Anforderung hat ein Kürzel, damit Realisierung und Test der Anforderung nachvollzogen werden können.

|  |  |
| --- | --- |
| **Kürzel** | **Anforderung** |
| UC 1.01 |  |
| UC 1.02 |  |

## Aktoren / Anwender

* User 1
* User 2

## Voraussetzungen

Was ist Voraussetzung, damit dieser Use Case starten kann?

## Ablauf des Use Case

Trigger sind die jeweiligen Auslöser des Use Case.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Name** | **Trigger** | **Input** | **Beschreibung** | **Output** |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |
| 3a |  |  |  |  |  |
| 3b |  |  |  |  |  |
| 3c |  |  |  |  |  |

## Logisches Ablaufdiagramm

Eventuell Verweis auf einen grafische Datei.

Beispiel eines Diagramms in Abbildung 1.



Abbildung 1 - Ablauf Bestellprozess

## Alternative Abläufe / Ausnahmen / Fehlerfälle

Beschreibung von alternativen Abläufen und Ausnahmen.

## Skizzen des User Interface

Welche Seiten existieren? Für jeden Links bzw. Button muss beschrieben sein, welche Aktion ausgeführt wird und was danach angezeigt wird.

## Inputdaten

Woher kommen Daten, die für diesen Use Case benötigt werden?

## User-Input

Welche Daten können wo eingegeben werden und werden wie validiert?

## Output Daten

Wo und wie wird Output angezeigt bzw. gespeichert?

## Rechte

Wer darf welche Funktionen (Use Cases) ausführen?

## Layout

Das Layout kann in einem Style Guide bzw. Design Manuel definiert werden.

## Texte

Welche Texte werden dargestellt, wie ist der genaue Wortlaut.

# User Stories

User Stories sind eine Vereinfachung der Use Cases aus dem Bereich der agilen Software Entwicklung.

## Dummy Bereich

|  |  |
| --- | --- |
| User Story („Als User will ich ...“) | Release Version |
| dies oder dies tun können. | 1 |
| Dummy Story | 2 |
| Dummy Story | 1 |

## Dummy Bereich

|  |  |
| --- | --- |
| User Story („Als User will ich ...“) | Release Version |
| Dummy Story | 1 |
| Dummy Story | 2 |

# UML – Diagramme

Bei Bedarf sind hier UML Diagramme einzufügen.

# Schnittstellen

Schnittstellen zu anderen Systemen inklusive exakter Beschreibung der Schnittstellen:

* Datenbank-Design
* Funktionen mit allen Parametern

# Story Board

In Projekten mit Animationen oder Videos ist es notwendig, ein Storyboard zu erstellen. Dieses wird in den meisten Fällen als externes Dokument erstellt und auf dieses verwiesen.

# Nicht Funktionale Anforderungen

Hier werden die einzelnen Nicht-Funktionalitäten und inhaltliche Ziele aufgelistet und detailliert beschrieben, wie die Ziele umgesetzt werden. Beinhaltet werden in jedem Fall die Haupt Ziele des Auftrags bzw. des Lastenhefts, möglicherweise noch optionale Ziele, sowie aber auch NICHT Ziele.

Nicht Funktionale Anforderungen sind all jene Dinge, die keine Funktion haben, aber trotzdem zur Umsetzung des Projektes notwendig sind, beziehungsweise gefordert werden, beispielsweise Usability, Design, Dokumentation, Präsentationen, Handbücher, Tests. Bei Eventprojekten jedoch auch alle Arten von Veranstaltungsteilen.

## Anwendbarkeit

Dies umfasst Themen wie

* Browserkompatibilität (Am besten ist es, ganz bestimmte Browser-Betriebssystem-Kombinationen zu nennen, auf welchen die Website getestet wird)
* Anzahl Klicks um eine Aufgabe zu erledigen
* Font Größe
* Max. Navigationstiefe
* Behindertengerecht
* Ist ein Test mit Usability Experten vorgesehen?

Beispiel:

Die gesamte Anwendung ist eine HTML5 Anwendung. Damit werden moderne Browser vorausgesetzt. Folgende Browser und Versionen werden innerhalb des Projektes getestet:

* Google Chrome, Version 16.0 - 26.0
* Firefox, Version 10.0 – 20.0
* Opera, Version 12.14
* Safari, Version 5.0 - 6.0
* IE, Version 8.0 – 10.0 (IE8 auch unter Windows XP)
* iOS 6 (Safari 6.0)
* Android 4.0.3 bis 4.2.x (Chrome Version 28 for Android)

## Zuverlässigkeit

Verfügbarkeit, Fehlertoleranz

Methoden, wie diese Zuverlässigkeit garantiert werden kann

## Performanz & Last

Responsezeiten, Anzahl gleichzeitiger User, Ressourcenverbrauch

Methoden, wie diese Performanz garantiert werden kann

Zum Beispiel:

Die Gesamtperformanz des Front-Ends wird branchenüblich übergeben, d.h. einfache Benutzerinteraktionen sind innerhalb einer Sekundengrößenordnung (1-60 Sekunden), komplexe Benutzerinteraktionen innerhalb einer Minutengrößenordnung (1-10 Minuten) abzuarbeiten sind.

Das System ist für 1 bis 1.000 Benutzer pro Minute ausgelegt. Falls das System für eine größere Anzahl von Benutzern ausgelegt werden soll, kann auf Wunsch des Kunden oder auf Empfehlung des Auftragnehmers diese Optimierung als optionaler Bestandteil angeboten werden.

## Weitere Qualitätskriterien

......................

## Test-Funktionalitäten

Welche Funktionalitäten müssen für einen (automatischen) Test entwickelt werden, zum Beispiel:

* Testen von Emailversand
* Logging

Woher kommen die Testdaten?

Beschreibung der Realisierung der Test-Funktionalitäten

## Logging

Welche Daten müssen protokolliert werden, zum Beispiel:

* Logging von
  + Usereingaben (z.B. POST-Daten)
  + Registrierung
  + Login
  + Bestimmte Datenbank-inserts/updates
  + Bestimmte User-Aktionen
  + Aufruf von bestimmten Seiten/Funktionen
  + Fehlern
* Ein/Ausschalten von Logging
* Logging in Datei/Datenbank
* Periodisches Löschen von Logeinträgen
* Wie und wo genau wird protokolliert?
* Wie können die protokollierten Daten angesehen werden?

## Statistik

Was ist gefordert?

Anzahl Aufrufe pro Seite, Anzahl Registrierungen/Logins/Bestellungen/Fehler/...

Wie wird die Statistik erstellt, welches Tool wird verwendet? Wo liegen die Daten? Wie werden Statistiken aufgerufen?

## Navigationsstruktur

Welche Navigationsbereiche existieren mit welchen Links, Administrator-Navigation nicht vergessen (z.B. Header, Hauptnavigation, Footer, Admin-Header, Admin-Hauptnavigation, etc.).

Wie sind die verschiedenen Use-Cases/Funktionalitäten miteinander verbunden? Gibt es Bereiche die nicht von überall erreichbar sind.

## Rechtestruktur

Falls eine Registrierung und/oder einen Administratorbereich gibt: Wer ist berechtigt, sich welche Seiten anzusehen, was kann ein/ausgeblendet sein, welche Funktionen erfordern eine bestimmte Benutzergruppe?

Genaue Beschreibung mit detaillierter Tabelle Benutzergruppen zu Seiten/Funktionen.

## Texte

Wer definiert, schreibt und kontrolliert Texte? Gibt es Mehrsprachigkeit (vorausplanen!) Wer liefert wann welche Texte? Wo werden Texte gespeichert?

## Sitemap

Alle Seiten müssen in der Sitemap vorhanden sein.

## Design

Welche Vorgaben gibt es für das Design? Auch für den Admin Bereich sollte es Designrichtlinien geben. Genaue Screenshots mit Maßangaben.

Eventuell kann das Design in einem eigenem Style Guide definiert werden, siehe Template.

## Serverinfrastruktur bzw. Hosting

Beschreibung des Hostings oder der Server Infrastruktur.

## Domains

Registrierung von etwaigen Domains

## Lizenzen & Rechte

Beschreibung von Lizenzen. Bleibt das Recht am Quellcode beim Urheber (ist stark zu empfehlen – wird das Urheberrecht abgetreten, so ist das meist mit Mehrkosten in der Höhe von 100% verbunden)

## Präsentation

Lorem Ipsum

Lorem Ipsum

# Architektur

## Hardware

Hardware-Architekturdiagramm mit Schnittstellen und deutlicher Unterscheidung was Teil des Projekts ist und was nicht.

Gibt es Systeme für Entwicklung, Test, Produktiveinsatz. Sind diese unterschiedlich?

Genaue Angaben, z.B. Rechnernamen, URLs, Versionsnummern, Login-Daten, etc..

Üblicherweise wird eine mehrstufige Architektur verwendet:

* Entwicklungssystem (nur hier wird entwickelt)
* Testsystem (manchmal auch mehrstufig für interne Tests, Usability Tests mit externen Usern und Kundentests)
* Produktivsystem

## Software

Software-Architekturdiagramm mit Schnittstellen und deutlicher Unterscheidung was Teil des Projekts ist und was nicht.

Objektmodell, Klassendesign

## Datenhaltung

Datenbankdesign. Wo werden welche Daten gespeichert?

# Deployment & Rollout

Wie bringt man die Applikation reproduzierbar auf den Produktionsserver?

# Dokumentation

Welche Dokumentation wird geliefert, zum Beispiel:

* Installationsanleitungen
* Benutzerhandbuch
* Administratorhandbuch
* Testprotokoll
* Onlinehilfe
* Systembeschreibung
* Schulungsunterlagen

In welcher Genauigkeit?

* Inline Kommentar
* Ausführliches Word-Dokument (Eine Angabe von der zu erwartenden Anzahl Seiten des Dokuments weckt keine falschen Erwartungen beim Kunden)

In welcher Sprache(n)?

# Deliverables

Was wird wie dem Kunden geliefert?

* Was: Dokumente, SW-Programm, Testprotokoll
* Wie: Per Email, CD

# Abgrenzungen

Was (Funktionalitäten, Dokumente, …) ist nicht Bestandteil des Projekts?

Die Rechte am Code verbleiben beim Auftragnehmer (siehe 13.15).