

Pflichtenheft

Projektname

-

Softwareprojekt SoSe19

Gruppe X

Logo

Vorname1	Nachname1
Vorname2	Nachname2
Vorname3	Nachname3
Vorname4	Nachname4
Vorname5	Nachname5
Vorname6	Nachname6
Vorname7	Nachname7
Vorname8	Nachname8



19. August 2019

# Tipps und Hilfen

**Information:** Dieses Kapitel und alle folgenden grauen Boxen dienen als Hilfestellungen und sollen im fertigen Dokument nicht enthalten sein.

Zur Versionsverwaltung während des Softwareprojekts muss *Git* genutzt werden. Git führt Textdokumente mit unterschiedlichen Zeilenbearbeitungen automatisch zusammen. Wir empfehlen den Einsatz von  $\text{\LaTeX}$  für alle Textdokumente. Um das Auto-Merging zu unterstützen, sollte nach jedem Satzende eine neue Zeile im Quelltext begonnen werden. Die *.tex*-Datei dieser PDF verdeutlicht dies. Erkennt Git, dass eine gleiche Zeile bearbeitet wurde, wird ein Konflikt auftreten. Dieser kann in der entsprechenden Datei von Hand mittels eines Texteditors behoben werden.

Fußnoten<sup>1</sup> werden für Homepages genutzt. Zitierungen können mittels eines *cite*-Befehls gesetzt, z.B. *citep* [1].

Tipps zur UML-Modellierung können im SE-Wiki<sup>2</sup> nachgelesen werden. Achtet darauf, dass eure Diagramme stets lesbar (Vektor-Grafiken!) und gut strukturiert sind. Oftmals ist es sinnvoll ein bis zwei Sätze zusätzlich für Diagrammelemente zu formulieren. So können Missverständnisse ausgeschlossen werden, was einen Einfluss auf die Korrektur haben kann. Diagramme für unwichtige Tätigkeiten (z.B. Login / Logout, User erstellen / löschen, Passwort ändern etc.) sind nicht erforderlich.

<sup>1</sup><https://www.se.informatik.uni-kiel.de/en>

<sup>2</sup><https://git.informatik.uni-kiel.de/ag-se/teaching-public/wikis/home>

So kann eine TODO-Notiz erzeugt werden



So kann eine Placeholder-Grafik beispielsweise in den Text eingefügt werden.

Abbildung 1: Beschreibung

# Inhaltsverzeichnis

<b>1</b>	<b>Lizenz</b>	<b>1</b>
<b>2</b>	<b>Zielbestimmungen</b>	<b>2</b>
<b>3</b>	<b>Produkteinsatz</b>	<b>3</b>
<b>4</b>	<b>Produktumgebung</b>	<b>4</b>
<b>5</b>	<b>Produktfunktionen</b>	<b>5</b>
5.1	Anwendungsfalldiagramm - App . . . . .	5
5.2	Anwendungsfalldiagramm - Server . . . . .	7
<b>6</b>	<b>Testfälle</b>	<b>9</b>
<b>7</b>	<b>Produktdaten</b>	<b>10</b>
<b>8</b>	<b>Benutzeroberfläche</b>	<b>11</b>
<b>9</b>	<b>Glossar</b>	<b>12</b>

# Kapitel 1

## Lizenz

Die Abgabe der Software und des Pflichtenhefts muss eine genaue Angabe der Lizenz enthalten, unter der die zu entwickelnde Software lizenziert wird. Um eine spätere Weiterverwendung und einen Praxiseinsatz der Software zu ermöglichen, empfehlen wir die Apache Lizenz 2.0 <sup>1</sup>. In diesem Kapitel soll die verwendete Lizenz notiert werden.

<sup>1</sup><http://www.apache.org/licenses/LICENSE-2.0>

## Kapitel 2

# Zielbestimmungen

Die Zielbestimmungen dienen dazu, die Ziele der Anforderungen nach Priorität zu sortieren. Es wird zwischen *Muss*-, *Soll*-, *Kann*- und *Abgrenzungskriterien* unterschieden, wobei weitere Einteilungen (z.B. nach Gerät oder Benutzer) innerhalb der Kategorien möglich sind.

Die *Musskriterien* umfassen alle Ziele und Funktionalitäten, die für einen Einsatz des entwickelten Produktes unabdingbar sind. Sie müssen daher ohne Kompromisse implementiert werden. Ein Wegfall eines einzelnen Musskriteriums würde das Produkt außer Betrieb setzen.

*Sollkriterien* (auch Wunschkriterien genannt) sind gewünschte Funktionen, die ebenfalls implementiert werden müssen, deren Wegfall auf Grund von unüblichen Umständen aber nicht den Einsatz des Produkts hindern würde.

Die *Kannkriterien* sind alle Ziele, die wünschenswert sind, aber nicht zwingend notwendige Funktionen darstellen. Oftmals werden diese nach Beendigung der höher priorisierten Kriterien umgesetzt.

*Abgrenzungskriterien* dienen dazu die Grenzen des Produkts zu definieren. Es soll erkennbar sein, was explizit **nicht** umgesetzt wird, damit Kunden nichts Falsches erwarten und Ziele stets klar definiert bleiben.

Für die Auflistung der Zielbestimmungen können Fließtexte oder auch Auflistungen mit ganzen Sätzen genutzt werden.

## Kapitel 3

# Produkteinsatz

In diesem Kapitel werden die folgenden drei Punkte erläutert:

1. *Anwendungsgebiete*: Was ist der Zweck des Produkts?
2. *Zielgruppen*: Für welche Benutzer (oder auch Rollen) ist das Produkt bestimmt? Welche Qualifikationen brauchen die Personen?
3. *Betriebsbedingungen*: Automatische oder manuelle Datensicherung? Autonomer oder beobachtender Betrieb?

Die einzelnen Teile des Produkteinsatzes werden üblicherweise als Fließtexte geschrieben.

## Kapitel 4

# Produktumgebung

In diesem Kapitel werden die folgenden Punkte erläutert. Eine jeweilige Unterteilung in Client und Server ist sinnvoll.

1. *Software*: Welche Software (Betriebssystem, Datenbanken, Webserver, externe Programme, etc.) ist auf den Zielsystemen für einen Betriebseinsatz erforderlich?
2. *Hardware*: Welche Hardware ist für den Produkteinsatz notwendig? Insbesondere Mindestanforderungen sind hier zu erwähnen.
3. *Orgware*: Umfasst organisatorische Anforderungen an die Produktumgebung, welche nicht unter die ersten beiden Kategorien fallen. Dieser Punkt ist stark abhängig vom Projekt und kann auch nur weniger interessante Informationen, wie z.B. Zugang zum Internet umfassen.
4. *Produktschnittstellen*: Welche Schnittstellen werden zur Laufzeit von dem zu entwickelnden System genutzt (kurze textuelle Beschreibung)?

Die einzelnen Abschnitte der Produktumgebung können als Fließtexte oder Absätze / Paragraphen mit ganzen Sätzen geschrieben werden.

## Kapitel 5

# Produktfunktionen

Die Produktfunktionen beschreiben jede einzelne Funktion des Produkts mittels Anwendungsfalldiagrammen und Anwendungsfaltabellen. Diese sollen möglichst ausschlaggebend für das zu entwickelnde System sein und nicht simple Produktfunktionen wie z.B. Login, Account erstellen, Gruppe beitreten, Passwort ändern oder ähnliches zeigen. Abbildung 5.3 stellt eine exemplarische Tabelle für die Beschreibung eines Anwendungsfalls dar. Stil und Formatierung sind variabel. Nicht jede Zelle muss immer gefüllt sein.

In Tabelle Abbildung 5.1 werden alle auftretenden Akteure beschrieben.

Akteur	Beschreibung	Verwendet in Anwendungsfall
Informatiker	Programmiert tolle Sachen	Programmieren, Kaffee trinken, Schlafen

Abbildung 5.1: Beschreibung der Akteure

### 5.1 Anwendungsfalldiagramm - App

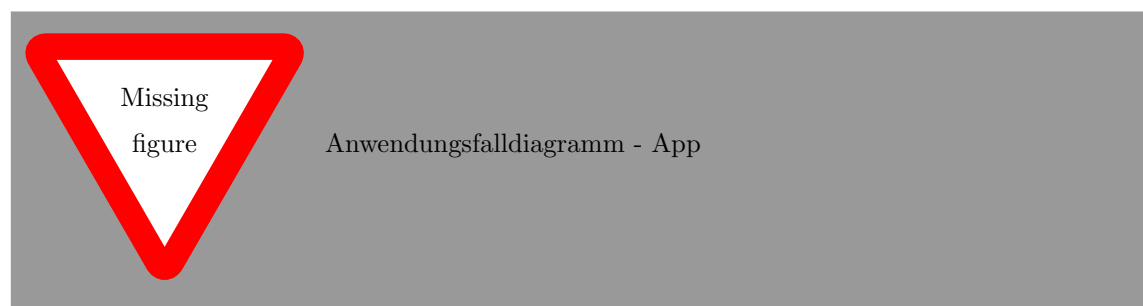


Abbildung 5.2: Anwendungsfalldiagramm - App



<b>Anwendungsfall ID</b>	XX-1
<b>Anwendungsfallname</b>	Hier steht ein Name.
<b>Initiierender Akteur</b>	Informatiker
<b>Weitere Akteure</b>	Designer, Techniker
<b>Kurzbeschreibung</b>	Hier steht eine Kurzbeschreibung.
<b>Vorbedingungen</b>	-
<b>Nachbedingungen</b>	Y trifft zu.
<b>Ablauf</b>	<ol style="list-style-type: none"> <li>1. Erster ganzer Satz.</li> <li>2. Zweiter ganzer Satz.</li> </ol>
<b>Alternative</b>	<ol style="list-style-type: none"> <li>1. Erster ganzer Satz.</li> <li>2. Zweiter ganzer Satz.</li> </ol>
<b>Ausnahme</b>	<ol style="list-style-type: none"> <li>1. Erster ganzer Satz.</li> <li>2. Zweiter ganzer Satz.</li> </ol>
<b>Benutzte Anwendungsfälle</b>	YY-1 (oder Name)
<b>Spezielle Anforderungen</b>	-
<b>Annahmen</b>	-

Abbildung 5.3: Anwendungsfall XX-1

## 5.2 Anwendungsfalldiagramm - Server

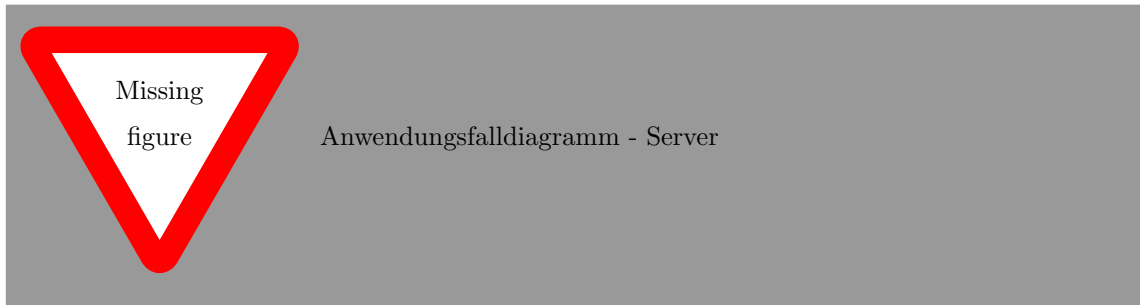


Abbildung 5.4: Anwendungsfalldiagramm - Server

<b>Anwendungsfall ID</b>	XX-1
<b>Anwendungsfallname</b>	Hier steht ein Name.
<b>Initiierender Akteur</b>	Informatiker
<b>Weitere Akteure</b>	Designer, Techniker
<b>Kurzbeschreibung</b>	Hier steht eine Kurzbeschreibung.
<b>Vorbedingungen</b>	-
<b>Nachbedingungen</b>	Y trifft zu.
<b>Ablauf</b>	<ol style="list-style-type: none"> <li>1. Erster ganzer Satz.</li> <li>2. Zweiter ganzer Satz.</li> </ol>
<b>Alternative</b>	<ol style="list-style-type: none"> <li>1. Erster ganzer Satz.</li> <li>2. Zweiter ganzer Satz.</li> </ol>
<b>Ausnahme</b>	<ol style="list-style-type: none"> <li>1. Erster ganzer Satz.</li> <li>2. Zweiter ganzer Satz.</li> </ol>
<b>Benutzte Anwendungsfälle</b>	YY-1 (oder Name)
<b>Spezielle Anforderungen</b>	-
<b>Annahmen</b>	-

Abbildung 5.5: Anwendungsfall XX-1

## Kapitel 6

# Testfälle

In diesem Abschnitt werden Testfälle für die Anwendungsfälle der Produktfunktionen definiert. Diese sollen später ebenfalls als **reale Tests** implementiert werden. Abbildung 6.1 stellt eine exemplarische Tabelle für die Beschreibung der zu testenden Anwendungsfälle dar. Stil und Formatierung sind variabel.

Nr.	Anwendungsfall ID	Szenario	Erwartetes Verhalten
1	XX-1	Der Informatiker programmiert ...	Der Quellcode ist schön.
2	XX-2	Der Informatiker trinkt Kaffee ...	Die Arbeitsleistung steigt.

Abbildung 6.1: Beschreibung der Akteure

## Kapitel 7

# Produkt Daten

Die Produkt Daten beschreiben die gespeicherten Daten des Produkts. Hier werden alle verarbeiteten Daten mit allen Attributen so genau wie jetzt schon möglich aufgeschrieben. So kann etwa ein Auto mit Hersteller, Modell, Farbe, Hubraum usw. langfristig gespeichert werden. Wichtig ist, dass nur tatsächlich benötigte Daten gespeichert werden, und dass Redundanzen vermieden werden. Form und Stil des Aufschrieb sind variabel, sollten jedoch sehr klar strukturiert sein.

## Kapitel 8

# Benutzeroberfläche

In diesem Kapitel werden erste Skizzen (Mockups) der Benutzeroberflächen dargestellt. Diese sollen in erster Linie dazu dienen, dem Kunden einen Überblick über die zu erstellenden UIs zu geben und ggf. Änderungen frühzeitig durchführen zu können. Dafür eignen sich spezielle Tools, wie z.B. Balsamiq Mockups<sup>1</sup>.

<sup>1</sup><https://balsamiq.com/products/mockups>



Abbildung 8.1: Startbildschirm

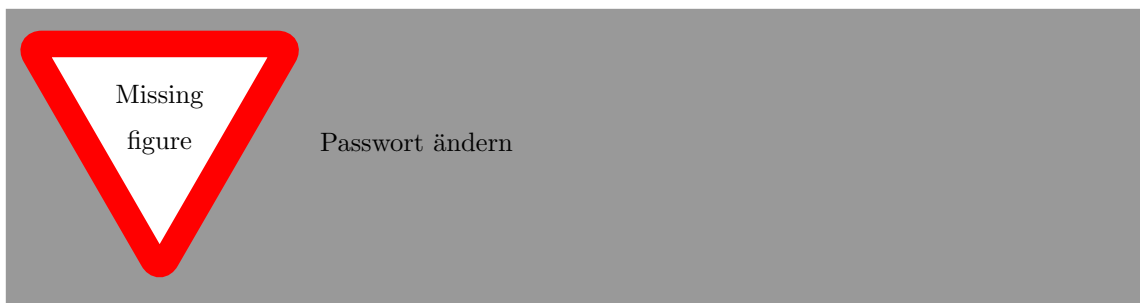


Abbildung 8.2: Passwort ändern

## Kapitel 9

# Glossar

In diesem Glossar können Akronyme und abkürzende Schreibweisen aufgelistet werden. Alle verwendeten Abkürzungen innerhalb des Projekts müssen hier erläutert werden.

Abkürzung	Beschreibung
Abk. A	Beschreibung A
Abk. B	Beschreibung B
Abk. C	Beschreibung C
Abk. D	Beschreibung D
Abk. E	Beschreibung E
Abk. F	Beschreibung F
Abk. G	Beschreibung G

Tabelle 9.1: Glossar

# Literaturverzeichnis

- [1] Mary Shaw. Writing good software engineering research papers: minitutorial. In *Proceedings of the 25th International Conference on Software Engineering (ICSE 2003)*, pages 726–736, Washington, DC, USA, 2003. IEEE Computer Society.