

# SEP – Hauptaufgabe

## PROJEKTMAPPE DES PROJEKTES

<SEPMAN>

### Spezifikation des Projektes

*Gruppe : L*

Angelo Soltner

Bardia Asemi-Soloot

Bijan Shahbaz Nejad

Dilara Güler

Dominik Häckel

Sabiha Can

Tobias van den Boom

### Systemdesign des Projektes

*Gruppe :*

<Teilnehmer 1>

<Teilnehmer 2>

...

<Teilnehmer n>

# Einleitung

Dieses Dokument enthält alle nötigen Informationen zur Erstellung eines Software-Produktes. Die Erstellung eines Software-Produktes wird im Allgemeinen auch als Programmierung bezeichnet. Programmierung kann man aber auch dahingehend verstehen, dass ein Computers zur Durchführung eines bestimmten Verhaltens konfiguriert werden muss.<sup>1</sup> Vorher müssen jedoch alle nötigen Informationen über das „bestimmte Verhalten“ zusammengetragen und dokumentiert werden. Diese Informationen bestehen aus Anforderungen (zu neudeutsch *Requirements*), Architekturbeschreibungen, etc., die im Folgenden in diesem Dokument wiedergegeben werden.

Dieses Dokument soll euch durch den gesamten Verlauf der Hauptaufgabe begleiten und dazu dienen, euer Projekt zu dokumentieren. Jeder Abschnitt beschäftigt sich mit einem Teilaspekt eurer Aufgabenstellung (Anforderungen, Architektur, Projektplanung, Testfälle etc.). An vielen Stellen findet ihr im Dokument folgendes Kästchen:

<i>Dies ist eine Hilfestellung.</i>
-------------------------------------

Diese Kästen dienen dazu, euch kurze Informationen über Ziele und Inhalte der jeweiligen Abschnitte zu geben. Sie sind spätestens zur finalen Abgabe der Projektmappe vollständig zu entfernen. Betrachtet dieses Dokument bitte nicht als Aufgabe, die man von oben nach unten abarbeiten soll; es soll vielmehr als durchgängige Dokumentation eurer Projektarbeit dienen und fortlaufend erweitert bzw. angepasst werden, sodass am Ende des SEPs eine Software entstanden ist, die sich in diesem Dokument wieder findet.

## **Eine Anmerkung in eigener Sache**

Dieses Dokument soll keinen starren Rahmen vorgeben, sondern vielmehr eine Richtschnur für eure Arbeit sein. Wenn Ihr Abschnitte anders gestalten wollt, so könnt ihr dies gerne tun (grobe Änderungen bitte mit eurem Betreuer absprechen, außerdem nur strukturelle Änderungen auf den Ebenen unter der ersten Strukturierungsebene (1, 2, 3, ...) durchführen). Ferner ist dieses Dokument keineswegs vollständig oder erhebt Anspruch auf Perfektion. Wenn ihr Anmerkungen und/oder Verbesserungsvorschläge habt, dann könnt ihr diese gerne an euren Betreuer weitergeben. Wir werden sie dann in das Vorlagedokument übernehmen.

Das SEP-Team wünscht euch  
**viel Erfolg**  
bei der Bearbeitung der Hauptaufgabe!!!

---

<sup>1</sup>vgl. B. Kovitz: Practical Software Requirements: A Manual of Content & Style; Manning 1998

# Inhaltsverzeichnis

<b>1.PROJEKTBE SCHREIBUNG (&lt;PROJEKTNAME&gt;)</b>	<b>5</b>
<b>2.ANFORDERUNGSDEFINITION</b>	<b>7</b>
2.1ZIELMODELL	7
2.2KONTEXTMODELL / SPIELMODELL	8
2.3SZENARIEN	8
<b>3.LOGISCHER ARCHITEKTURENTWURF</b>	<b>10</b>
3.1DATENFLUSSDIAGRAMM	10
3.2MINI SPEZIFIKATION	10
<b>3.2.1&lt;NAME PROZESS 1&gt;</b>	<b>10</b>
<b>3.2.2&lt;NAME PROZESS N&gt;</b>	<b>10</b>
3.3DATA DICTIONARY	10
3.4MESSAGE SEQUENCE CHARTS	10
<b>3.4.1BMSCS</b>	<b>10</b>
bMSC-1: <Name des bMSC>	10
bMSC-n: <Name des bMSC>	10
<b>3.4.2ABBILDUNG DER SZENARIEN AUF MESSAGE SEQUENCE CHARTS</b>	<b>11</b>
<b>3.4.3HMSC</b>	<b>11</b>
<b>4.TECHNISCHER ARCHITEKTURENTWURF</b>	<b>12</b>
4.1GUI-PAPIERPROTOTYP	12
<b>4.1.1SCREEN „&lt;NAME DES SCREENS&gt;“</b>	<b>12</b>
4.2TECHNISCHES KONZEPT	12
<b>4.2.1&lt;NAME KOMPONENTE 1&gt;</b>	<b>12</b>
<b>4.2.2&lt;NAME KOMPONENTE N&gt;</b>	<b>12</b>
4.3KOMPONENTENDIAGRAMM	13
<b>4.3.1KOMPONENTENBESCHREIBUNG</b>	<b>13</b>
<Name Komponente 1>	13
<Name Komponente n>	13
<b>4.3.2INTERFACEBESCHREIBUNG</b>	<b>13</b>
<Name Interface 1>	13
<Name Interface n>	13
<b>5.TESTARTEFAKTE</b>	<b>14</b>
5.1MODULTEST	14
<b>5.1.1TESTSPEZIFIKATION</b>	<b>14</b>
Modultestfall 1: <Kurzbezeichnung MTF-1>	14
Modultestfall n: <Kurzbezeichnung MTF-n>	14
<b>5.1.2TESTERGEBNISSE</b>	<b>14</b>
Testprotokoll Modultestfall 1 (1. Testdurchführung)	14
Testprotokoll Modultestfall 1 (n. Testdurchführung)	14
Testprotokoll Modultestfall n (1. Testdurchführung)	14
Testprotokoll Modultestfall n (n. Testdurchführung)	15
5.2SYSTEMTEST	15
<b>5.2.1TESTSPEZIFIKATION</b>	<b>15</b>
Systemtestfall 1: <Kurzbezeichnung STF-1>	15
Systemtestfall n: <Kurzbezeichnung STF-n>	15

<b>5.2.2TESTERGEBNISSE .....</b>	<b>16</b>
Testprotokoll Systemtestfall 1 (<1. Testdurchführung>).....	16
Testprotokoll Systemtestfall 1 (<n. Testdurchführung>).....	16
Testprotokoll Systemtestfall n (Version <1. Testdurchführung>).....	16
Testprotokoll Systemtestfall n (Version <n. Testdurchführung>).....	16

## Projektbeschreibung (<Projektname>)

*In diesem Abschnitt soll die Projektbeschreibung abgedruckt werden, die ihr als Aufgabenbeschreibung von eurem Betreuer erhalten habt. Sie dient als initiales Anforderungsdokument für eure Spezifikationsaktivitäten.*

<Eure Projektbeschreibung>

Der Softwareentwicklungsprozess basiert im Rahmen des SEP auf dem angepassten V-Modell. Die Projektmappe ist entsprechend den Phasen des V-Modells aufgebaut. Jede Phase wird Schritt für Schritt im Verlaufe der Veranstaltung bearbeitet und dokumentiert.

# Anforderungsdefinition

Dieser Abschnitt soll jeweils von der Gruppe, die für die Spezifikation des Projekts zuständig ist, ausgefüllt werden.

Auf dieser Ebene wird das System als Ganzes betrachtet. Jedoch gibt es kein Wissen über die Abläufe im System oder über die genauen Funktionalitäten.

## Zielmodell

In diesem Abschnitt sollen die Ziele des Systems beschrieben werden. Ein Zielbaum (oder Zielgraph) stellt dabei die geeignetste Methode zur Darstellung dar. Eine textuelle Beschreibung aller Ziele detailliert das Modell entsprechend.

Der Zielbaum stellt ein Artefakt dar, das die Spezifikation des logischen und des technischen Systemdesigns überspannt. Von daher muss er zweimal während der Projektlaufzeit bearbeitet werden. Zuerst wird er während der Spezifikation des logischen Systemdesigns erstellt, und dann während der Spezifikation des technischen Systemdesigns verfeinert, erweitert und aktualisiert. Das Zielmodell enthält damit Informationen unterschiedlicher Detaillierung und stellt die enthaltenen Ziele in ihren Beziehungen dar.

Aufgrund der Trennung von Anforderungsspezifikation und technischem Systemdesign wird folgende Dokumentationsrichtlinie verwendet: Das Zielmodell wird in der Spezifikationsphase in diesem Abschnitt **Fehler! Verweisquelle konnte nicht gefunden werden.** erstellt und fortwährend aktualisiert; nach der Übergabe der Anforderungsspezifikation wird im Rahmen des technischen Systemdesigns das Zielmodell in diesem Abschnitt von der Partner-Gruppe, die das System implementiert, vervollständigt.

Zuletzt noch einige inhaltliche und strukturelle Anmerkungen:

- 1) Ziele beschreiben die Intention eines Akteurs mit einem zu bauenden System.
- 2) Die Akteure der Ziele müssen sich im Kontextmodell des Systems wiederfinden.
- 3) Man unterscheidet zwischen logischen und technischen Zielen. Logische Ziele sollen immer lösungsunabhängig sein (z.B. „Der Nutzer möchte seine Termine verwalten“), während technische Ziele explizit lösungsabhängig sein sollen (z.B. „Die Termine des Nutzers werden mit dem Google-Kalender des Nutzer synchronisiert.“).
- 4) Logische Ziele werden nur während der Spezifikation des logischen Systemdesigns erstellt und technische Ziele nur während der Spezifikation des technischen Systemdesigns erstellt. Das bedeutet auch, dass die logischen und technischen Ziele von zwei verschiedenen Gruppen erstellt werden.
- 5) Man unterscheidet außerdem zwischen Softgoals und Hardgoals. Softgoals sind Ziele, die man nicht objektiv überprüfen kann (z.B. „Die Termine des Nutzers werden übersichtlich dargestellt.“), während Hardgoals Ziele sind, die man explizit objektiv überprüfen kann (z.B. „Es können mindestens 10 Termine des Nutzers gleichzeitig dargestellt werden.“).
- 6) Es gibt also vier verschiedene Arten von Zielen. 1. Logische Ziele, die Hardgoals sind; 2. Logische Ziele, die Softgoals sind; 3. Technische Ziele, die Hardgoals sind; und 4. Technische Ziele, die Softgoals sind.
- 7) Ziele der Ebene 1 des Zielbaums werden in der Form **Z-<x>** nummeriert, wobei x die Nummer des Zielbaums ist.
- 8) Ziele aller weiteren Ebenen werden in der Form **Z-<L/T>-<HG/SG>-<x>.<y>** nummeriert. Es wird angegeben ob es sich um ein logisches (L) oder technisches (T) Ziel, ein Hardgoal (HG) oder ein Softgoal (SG) handelt und welche Nummer (x,y) das Ziel hat.

- 9) Die Eltern-Kind-Beziehung zwischen Zielen wird durch die Punktnotation am Ende der Zielnummer angegeben. So ist das Ziel 3.4.2 das zweite Unterziel des Ziels 3.4, das wiederum ein Unterziel von Ziel 3 ist.
- 10) Wichtig: der Zielbaum ist in jedem Ast nicht auf eine Anzahl Ebenen beschränkt. D.h. der Zielbaum kann in jedem Ast beliebig viele Ebenen aufweisen. Die Nummerierung muss dementsprechend angepasst werden.

Der fertige Zielbaum enthält lediglich Hardgoals als Blätter, um die Überprüfbarkeit zu gewährleisten. <ggf. grafische Repräsentation des Zielmodells>

Erinnerung: Struktur der Ziele: Z<X.Y.Z>-<L|T>-<SG|HG>: <Name der Ziels>

- **Z-1: Spielfiguren**  
Roboter dienen als Spielfiguren
  - **Z-L-HG-1.1: Geist**  
Der Roboter kann als Geist fungieren
    - **Z-L-HG-1.1.1: Autonomie**  
Die Roboter arbeiten autonom
    - **Z-L-HG-1.1.2: Modus Auswahl**  
Der Roboter kann sich in einem der u.g. Modi befinden
      - **Z-L-HG-1.1.2.1: Verfolgung**  
Die Roboter verfolgen SEPMAN
      - **Z-L-HG-1.1.2.2: Zufall**  
Die Roboter bewegen sich zufällig
      - **Z-L-HG-1.1.2.3: Verteidigung**  
Die Roboter verteidigen die Power-Ups
  - **Z-L-HG-1.2: SEPMAN**  
Der Roboter kann als SEPMAN fungieren
    - **Z-L-HG-1.2.1: Spielersteuerung**  
Der Nutzer kann durch Aktionen den SEPMAN steuern.
- **Z-2: Spielfeld**  
Das System setzt ein Spielfeld um, auf dem sich die Roboter bewegen
  - **Z-L-HG-2.1: Knotenpunkte**  
Roboter kommen auf den Knotenpunkten zum stehen
    - **Z-L-HG-2.1.1: Position der Roboter**  
Zeigt an ob sich Roboter auf Knotenpunkt befindet
    - **Z-L-HG-2.1.2: Zustand**  
Zeigt Zustand des Knotens an (Normal / Power-Up)
  - **Z-L-HG-2.2: Kanten**  
Roboter sollen nur zwischen Knotenpunkten wechseln, die mit Kanten verbunden sind
    - **Z-L-HG-2.2.1: (Besuch-)Zustand**  
Zeigt an ob Kante besucht ist oder nicht
    - **Z-L-HG-2.2.2:**



- **Z-3: Spielregeln**  
Das System folgt den u.g. Regeln
  - **Z-L-HG-3.1: Startposition**  
Die Roboter starten auf bestimmten Positionen.
  - **Z-L-HG-3.2: Spielgeschehen**  
Die Roboter handeln nach gewissen Regeln
    - **Z-L-HG-3.2.1: Kollision**  
Bei Kollision zwischen SEPMAN und Geist tritt eine Aktion auf.
      - **Z-L-HG-3.2.1.1: Power-Up**  
Power-Ups können vom SEPMAN aufgesammelt werden  
Die Geister flüchten vom SEPMAN. Bei Kollision wird der Geist deaktiviert; Nach einer bestimmten Zeit wird der Geist zurückgesetzt.
      - **Z-L-HG-3.2.1.2: Kein Power-Up**  
„Normales“ Spielerlebnis  
Bei Kollision wird das Spiel pausiert und der SEPMAN verliert ein Leben; Es werden alle Roboter auf ihre Startposition zurückgesetzt. (Kanten bleiben erhalten)
    - **Z-L-HG-3.2.2: Anzeige**  
Dem Spieler wird angezeigt, wenn ein Roboter deaktiviert ist.  
Dem Spieler wird sein Leben angezeigt.
    - **Z-L-HG-3.2.3: Spielende**  
Das Spiel endet unter einer der beiden Bedingungen.
      - **Z-L-HG-3.2.3.1 Sieg**  
Der Spieler hat gewonnen, wenn alle Kanten mind. 1 Mal begangen wurden.
      - **Z-L-HG-3.2.3.2 Niederlage**  
Der Spieler hat verloren, wenn er kein Leben mehr hat.

## Kontextmodell / Spielmodell

*Durch das Kontextmodell wird die Umgebung des Systems modelliert. Es wird definiert womit das System interagiert. Alle Elemente des Kontextmodells sollen auch beschrieben werden.*

*Für Gruppen, die ein Spiel implementieren, tritt an die Stelle des Kontextmodells das Spielmodell. Es enthält alle Elemente des Spiels und beschreibt ihre Interaktion mit dem Spieler. Die Elemente des Spiels sollen außerdem beschrieben werden.*

<Bild des Kontextmodells>

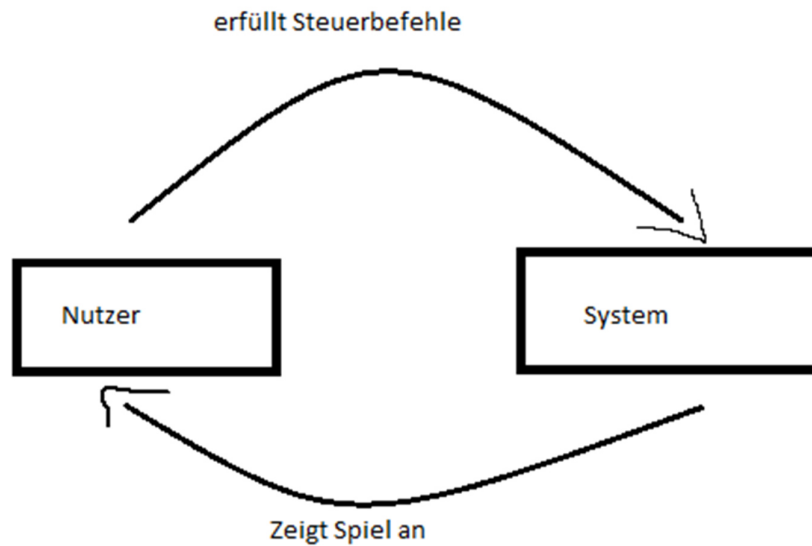
<Akteur 1/Ext. System 1>

*Zu jedem Akteur/Ext. System gehört eine kurze Beschreibung. Die Ziele des Akteurs sollten sich in Abschnitt **Fehler! Verweisquelle konnte nicht gefunden werden.** wiederfinden und entsprechend verwiesen werden. Zur Beschreibung gehört auch eine textuelle Definition der Daten, die mit dem zu modellierenden System ausgetauscht werden.*

<Beschreibung des Akteurs>

<Akteur n/Ext. System n>

<Beschreibung des Akteurs>



**Nutzer:**

Der Nutzer interagiert mit dem System, indem er Steuerbefehle sendet.

**System:**

Das entwickelte System zeigt dem Nutzer das Spielgeschehen an.

## Szenarien

*Szenarien verbinden die Artefakte miteinander. Jedes Szenario beschreibt eine konkrete Interaktion mit dem zu bauenden System. In diesen Szenarien soll beschrieben werden, wie die Ziele der Akteure durch eine Interaktion mit dem System erfüllt werden.*

**<Name Szenario 1>**

*Natürlich sprachliche Dokumentation eines Szenarios mit Referenz auf die beteiligten Akteure/Systeme aus Abschnitt Fehler! Verweisquelle konnte nicht gefunden werden. und der erreichten Ziele aus Abschnitt 2.1*

**<Beschreibung Szenario 1>**

**<Name Szenario n>**

**<Beschreibung Szenario n>**

# Logischer Architekturentwurf

## Datenflussdiagramm

*Durch ein Datenflussdiagramm werden die Datenflüsse und Verarbeitungsprozesse der Daten innerhalb eines Systems modelliert. Daten kommen von externen Systemen oder Akteuren (Abschnitt **Fehler! Verweisquelle konnte nicht gefunden werden.**) in das System und werden verarbeitet.*

<Grafik des Datenflussdiagramms>

## Mini Spezifikation

*Die Mini Spezifikation gibt Einblick in die Prozesse des DFD. Sie beschreibt wie der Prozess Eingabedaten in die entsprechenden Ausgabedaten transformiert Dabei geht es nicht darum bereits entsprechende Algorithmen zu entwickeln, sondern natürlich sprachlich festzuhalten aus welchen Informationen der Eingabedaten die Ausgabedaten ermittelt werden.*

<Name Prozess 1>

<Beschreibung Prozess1>

<Name Prozess n>

<Beschreibung Prozess n>

## Data Dictionary

*Das Data Dictionary schlüsselt die Datenflüsse des DFD in atomare Datentypen auf. Jeder Datenfluss muss dabei einem eindeutigen atomaren Datentyp zugeordnet werden. Die Anzahl der Ebenen in die ein Datenfluss zerlegt werden kann, variiert je nach Datentyp. Ein Datentyp gilt als atomar, wenn er sich nicht in weitere Datentypen zerlegen lässt und einem fest definierten Wertebereich zuzuordnen ist.*

## Message Sequence Charts

*Mit Hilfe von MSC werden Interaktionen zwischen den Elementen des DFDs aus Abschnitt Datenflussdiagramm modelliert. Zu jedem Szenario aus Abschnitt Szenarien wird dazu ein oder mehrere zusammenhängende basic MSC (bMSC) modelliert, dass den Datenaustausch zwischen den Elementen des DFDs zeigt. Durch das hMSC werden die bMSC in einen Zusammenhang gesetzt.*

### bMSCs

bMSC-1: <Name des bMSC>

<Grafik des bMSC>

bMSC-n: <Name des bMSC>

<Grafik des bMSC>

### Abbildung der Szenarien auf Message Sequence Charts

*Es muss dokumentiert werden, welche Szenarien in welchen bMSCs (oder in welcher Reihenfolge) umgesetzt wurden.*

<Name Szenario 1>	bMSC-1: <Name des bMSC> bMSC-2: <Name des bMSC> bMSC-1: <Name des bMSC>
...	...
<Name Szenario n>	

## **hMSC**

<Grafik des hMSC>

# Technischer Architekturentwurf

*Dieser Abschnitt wird von der Partner-Gruppe ausgefüllt, die das Projekt auch am Ende implementieren wird. Vor der Bearbeitung dieses Abschnitts wird das Dokument an die Partner-Gruppe übergeben.*

*Auf der technischen Ebene erfolgt der kreative Schritt der Konstruktion des technischen Systems. Hierbei liegt der kreative Schritt besonders in der Umsetzung der logischen Architektur der DFDs in ein technisches System mit „echten“ Komponenten.*

## GUI-Papierprototyp

**Screen „<Name des Screens>“**

<Scan des Screen-Papierprototypen>

## Technisches Konzept

<Grafik des technischen Konzepts>

**<Name Komponente 1>**

<Beschreibung zu Komponente 1>

**<Name Komponente n>**

<Beschreibung zu Komponente n>

## Komponentendiagramm

*Die technischen Komponenten zeigen die Realisierung des Systems. Dazu wird hier nun beschrieben, welche echten Komponenten später im System zu finden sind und damit implementiert werden. Sowohl zu jeder technischen Komponente als auch zu jedem Interface soll es eine kurze Beschreibung geben. Zu jeder Komponente soll angegeben werden, welche Funktionen umgesetzt werden. Zur Beschreibung eines Interfaces gehören die Zuordnung zu anbietenden und nutzenden Komponenten sowie die Auflistung aller Methodenköpfe inklusive ihrer Übergabeparameter und Rückgabewerte.*

<Grafik des Komponentendiagramms>

### Komponentenbeschreibung

<Name Komponente 1>

<Beschreibung zu Komponente 1>

<Name Komponente n>

<Beschreibung zu Komponente n>

### Interfacebeschreibung

<Name Interface 1>

<Beschreibung zu Interface 1>

<Name Interface n>

<Beschreibung zu Interface n>

# Testartefakte

## Modultest

### Testspezifikation

#### Modultestfall 1: <Kurzbezeichnung MTF-1>

Testziel	
Schnittstelle/Klasse	
Vorbedingung	
Nachbedingung	
Bestehens Kriterien	

#### Modultestfall n: <Kurzbezeichnung MTF-n>

Testziel	
Schnittstelle/Klasse	
Vorbedingung	
Nachbedingung	
Bestehens Kriterien	

### Testergebnisse

#### Testprotokoll Modultestfall 1 (1. Testdurchführung)

Testziel	
Schnittstelle/Klasse	
Vorbedingung	
Nachbedingung	
Bestehens Kriterien	
Datum	
Tester	
Version der Software	
Testtreiber	
Testsystem & -umgebung	
Testurteil	

#### Testprotokoll Modultestfall 1 (n. Testdurchführung)

Testziel	
Schnittstelle/Klasse	
Vorbedingung	
Nachbedingung	
Bestehens Kriterien	
Datum	
Tester	
Version der Software	
Testtreiber	
Testsystem & -umgebung	
Testurteil	

#### Testprotokoll Modultestfall n (1. Testdurchführung)

Testziel	
Schnittstelle/Klasse	

Vorbedingung	
Nachbedingung	
Bestehens Kriterien	
Datum	
Tester	
Version der Software	
Testtreiber	
Testsystem & -umgebung	
Testurteil	

### **Testprotokoll Modultestfall n (n. Testdurchführung)**

Testziel	
Schnittstelle/Klasse	
Vorbedingung	
Nachbedingung	
Bestehens Kriterien	
Datum	
Tester	
Version der Software	
Testtreiber	
Testsystem & -umgebung	
Testurteil	

## **Systemtest**

### **Testspezifikation**

#### **Systemtestfall 1: <Kurzbezeichnung STF-1>**

Szenario		
Schritt	Aktion (User)	Erwartete Reaktion (System)
1		
2		
3		
4		
...		

#### **Systemtestfall n: <Kurzbezeichnung STF-n>**

Szenario		
Schritt	Aktion (User)	Erwartete Reaktion (System)
1		
2		
3		
4		
...		



## **Testergebnisse**

### **Testprotokoll Systemtestfall 1 (<1. Testdurchführung>)**

Datum			
Tester			
Version der Software			
Szenario			
Schritt	Aktion (User)	Erwartete Reaktion (System)	Tatsächliche Reaktion (System)
1			
2			
3			
4			
...			
Testurteil			

### **Testprotokoll Systemtestfall 1 (<n. Testdurchführung>)**

Datum			
Tester			
Version der Software			
Szenario			
Schritt	Aktion (User)	Erwartete Reaktion (System)	Tatsächliche Reaktion (System)
1			
2			
3			
4			
...			
Testurteil			

### **Testprotokoll Systemtestfall n (Version <1. Testdurchführung>)**

Datum			
Tester			
Version der Software			
Szenario			
Schritt	Aktion (User)	Erwartete Reaktion (System)	Tatsächliche Reaktion (System)
1			
2			
3			
4			
...			
Testurteil			

### **Testprotokoll Systemtestfall n (Version <n. Testdurchführung>)**

Datum			
Tester			
Version der Software			
Szenario			
Schritt	Aktion (User)	Erwartete Reaktion (System)	Tatsächliche Reaktion (System)

Software-Entwicklung und Programmierung Wintersemester 2015/2016

1			
2			
3			
4			
...			
Testurteil			