

SEP – Hauptaufgabe

Projektmappe des Projektes

SEP RP

Spezifikation des Projektes

Gruppe Gruppe K:

Kathi
Jo
Sami
Gizem
Thorben
Tristan
Mark Leifeld

Systemdesign des Projektes

Gruppe Gruppe L:

<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.¹ 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:

Dies ist eine Hilfestellung.

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 Vorlagendokument übernehmen.

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

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

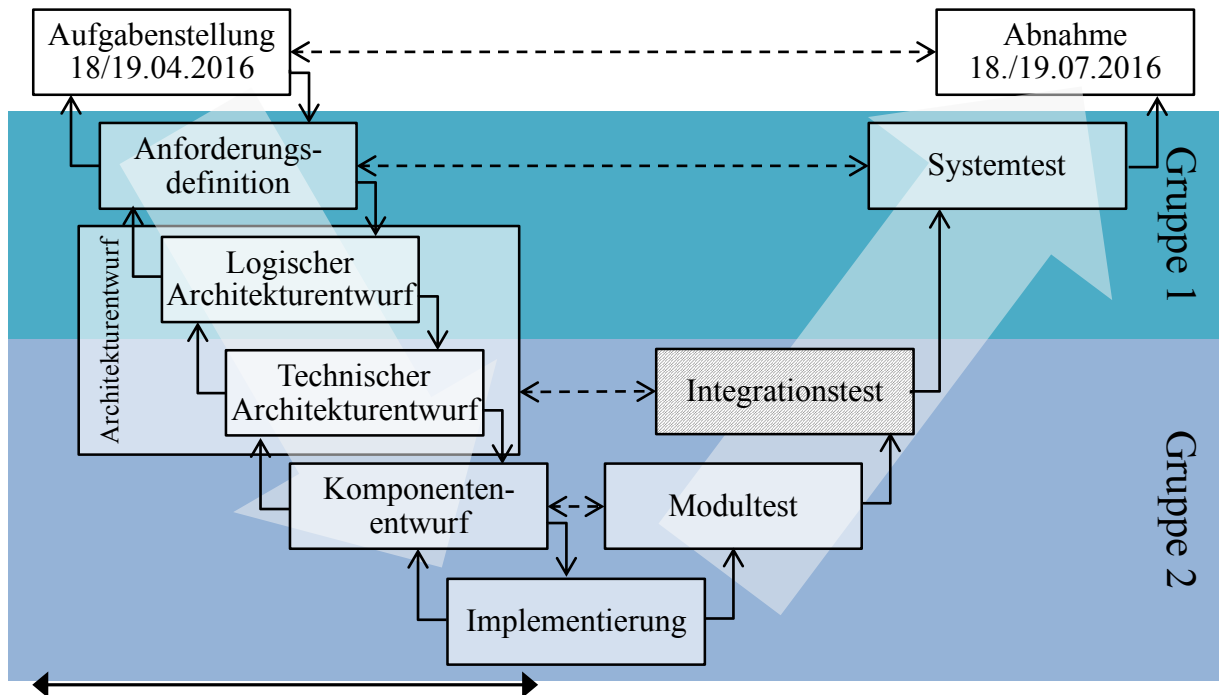
Inhaltsverzeichnis

1.	PROJEKTBESCHREIBUNG (4
2.	ANFORDERUNGSDEFINITION	6
2.1	ZIELMODELL	6
2.2	KONTEXTMODELL / SPIELMODELL	10
	<Akteur 1/Ext. System 1>	10
	<Akteur n/Ext. System n>	Fehler! Textmarke nicht definiert.
2.3	SZENARIEN	10
	<Name Szenario 1>	Fehler! Textmarke nicht definiert.
	<Name Szenario n>	Fehler! Textmarke nicht definiert.
3.	LOGISCHER ARCHITEKTURENTWURF	12
3.1	DATENFLUSSDIAGRAMM	12
3.2	MINI SPEZIFIKATION	12
3.2.1	<Name Prozess 1>	12
3.2.2	<Name Prozess n>	12
3.3	DATA DICTIONARY	12
3.4	MESSAGE SEQUENCE CHARTS	12
3.4.1	bMSCs	12
	bMSC-1: <Name des bMSC>	12
	bMSC-n: <Name des bMSC>	12
3.4.2	Abbildung der Szenarien auf Message Sequence Charts	13
3.4.3	hMSC	13
4.	TECHNISCHER ARCHITEKTURENTWURF	14
4.1	GUI-PAPIERPROTOTYP	14
4.1.1	Screen „<Name des Screens>“	14
4.2	TECHNISCHES KONZEPT	14
4.2.1	<Name Komponente 1>	14
4.2.2	<Name Komponente n>	14
4.3	KOMPONENTENDIAGRAMM	15
4.3.1	Komponentenbeschreibung	15
	<Name Komponente 1>	15
	<Name Komponente n>	15
4.3.2	Interfacebeschreibung	15
	<Name Interface 1>	15
	<Name Interface n>	15
5.	TESTARTEFAKTE	16
5.1	MODULTEST	16
5.1.1	Testspezifikation	16
	Modultestfall 1: <Kurzbezeichnung MTF-1>	16
	Modultestfall n: <Kurzbezeichnung MTF-n>	16
5.1.2	Testergebnisse	16
	Testprotokoll Modultestfall 1 (1. Testdurchführung)	16
	Testprotokoll Modultestfall 1 (n. Testdurchführung)	16
	Testprotokoll Modultestfall n (1. Testdurchführung)	17
	Testprotokoll Modultestfall n (n. Testdurchführung)	17
5.2	SYSTEMTEST	17
5.2.1	Testspezifikation	17
	Systemtestfall 1: <Kurzbezeichnung STF-1>	17
	Systemtestfall n: <Kurzbezeichnung STF-n>	18
5.2.2	Testergebnisse	18
	Testprotokoll Systemtestfall 1 (<1. Testdurchführung>)	18
	Testprotokoll Systemtestfall 1 (<n. Testdurchführung>)	18
	Testprotokoll Systemtestfall n (Version <1. Testdurchführung>)	18
	Testprotokoll Systemtestfall n (Version <n. Testdurchführung>)	19

1. Projektbeschreibung SEP RP

Die Aufgabe besteht in der Entwicklung eines Action Rollenspiels. Der Spieler soll dabei in die Rolle einer fiktiven Person (im Folgenden: Charakter) schlüpfen und Gegner bekämpfen. Der Spieler steuert den Charakter mit Maus und Tastatur aus der Vogelperspektive. Zu Beginn des Spiels soll der Spieler die Möglichkeit haben den Charakter nach seinen Wünschen anzupassen. Dabei hat er die Möglichkeit sich einen Charakter aus mindestens vier Charaktertypen auszuwählen, die jeweils mindestens vier einzigartige Fähigkeiten erlernen können. Diese Fähigkeiten soll der Spieler bewusst einsetzen können, um das Spielgeschehen zu beeinflussen. Im Verlauf des Spiels kann der Spieler seinen Charakter verbessern, um sich in immer schwierigeren Kämpfen behaupten zu können. Gegner gibt es in mindestens drei verschiedenen Typen, die der Spieler bekämpfen kann, und die auf unterschiedliche Arten auf den Spieler reagieren. Die Geschichte des Spiels soll den Charakter des Spielers aktiv einbinden. Das heißt, dass die Entscheidungen, die der Spieler im Verlauf des Spiels trifft, sich auf den Verlauf der Geschichte auswirken. Insgesamt kann der Spieler die Geschichte des Spiels so zu mindestens drei verschiedenen Enden führen.

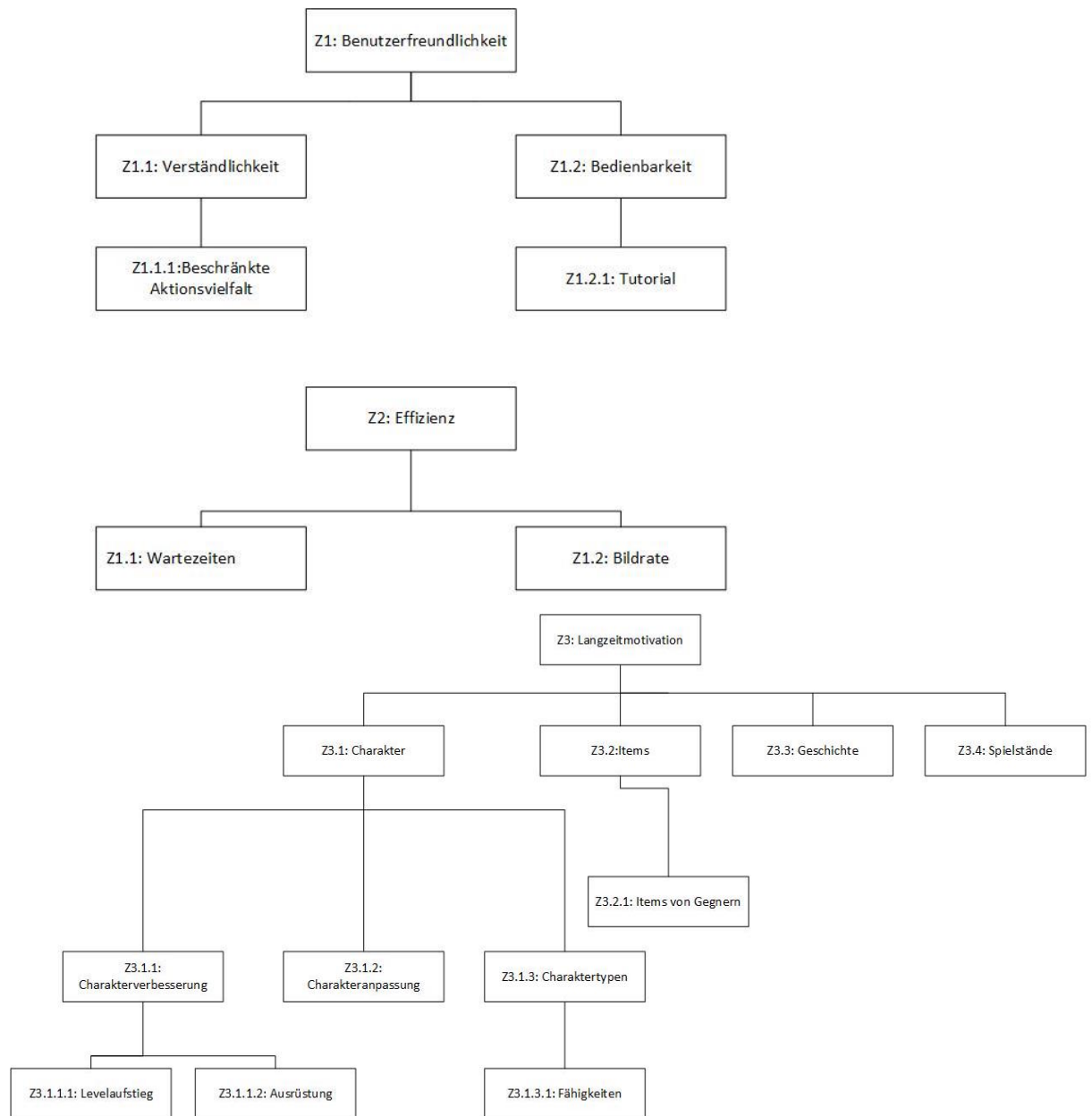
Der Charakter soll darüber hinaus in der Lage sein während des Spielens Items zu erlangen, die sein Charakter benutzen kann, um Vorteile zu erhalten. Es gibt mindestens fünf verschiedene Kategorien von Items, die sich in ihrer Nutzung unterscheiden, wobei es von jeder Kategorie mindestens vier verschiedene Items geben soll. Dabei können selten auftretende Items dem Charakter einen größeren Vorteil bringen als häufig auftretende Items. Außerdem beinhaltet das Spiel ein Währungssystem, um es dem Spieler zu ermöglichen, Items in einem Auktionshaus anderen Spielern zum Verkauf anzubieten. Neben dem Verkauf im Auktionshaus soll der Spieler auch in der Lage sein durch das Bekämpfen von Gegnern Geld des Währungssystems zu erhalten. Das Auktionshaus soll über eine Netzwerkverbindung erreicht werden können. Das Spiel soll für Bildschirme mit einer Auflösung von 1920x1080 Pixel entwickelt werden und der Charakter darf auf diesen Bildschirmen maximal 150 Pixel hoch und 150 Pixel breit sein. Das Spiel beinhaltet mindestens drei separierte Bereiche, die der Spieler erkunden kann. Der erste dieser Bereiche soll mindestens 15.000.000 Quadratpixel groß sein, der zweite Bereich mindestens 25.000.000 Quadratpixel groß sein und der dritte Bereich mindestens 35.000.000 Quadratpixel groß sein. In jedem dieser drei Bereiche hält sich ein jeweils einzigartiger Gegner auf, der deutlich schwerer zu besiegen ist als die normalen Gegner. Außerdem soll der Charakter, am Ende des Spiels, einen weiteren einzigartigen Gegner besiegen müssen, der schwieriger zu besiegen ist als alle anderen Gegner. Dem Spieler soll mindestens nach dem Durchqueren jeder der drei Bereiche angeboten werden seinen Spielstand zu speichern. Gespeicherte Spielstände sollen auch nach Beenden und Neustarten des Spiels bestehen bleiben. Der Spieler kann so eine beliebige Anzahl Spielstände speichern und laden.

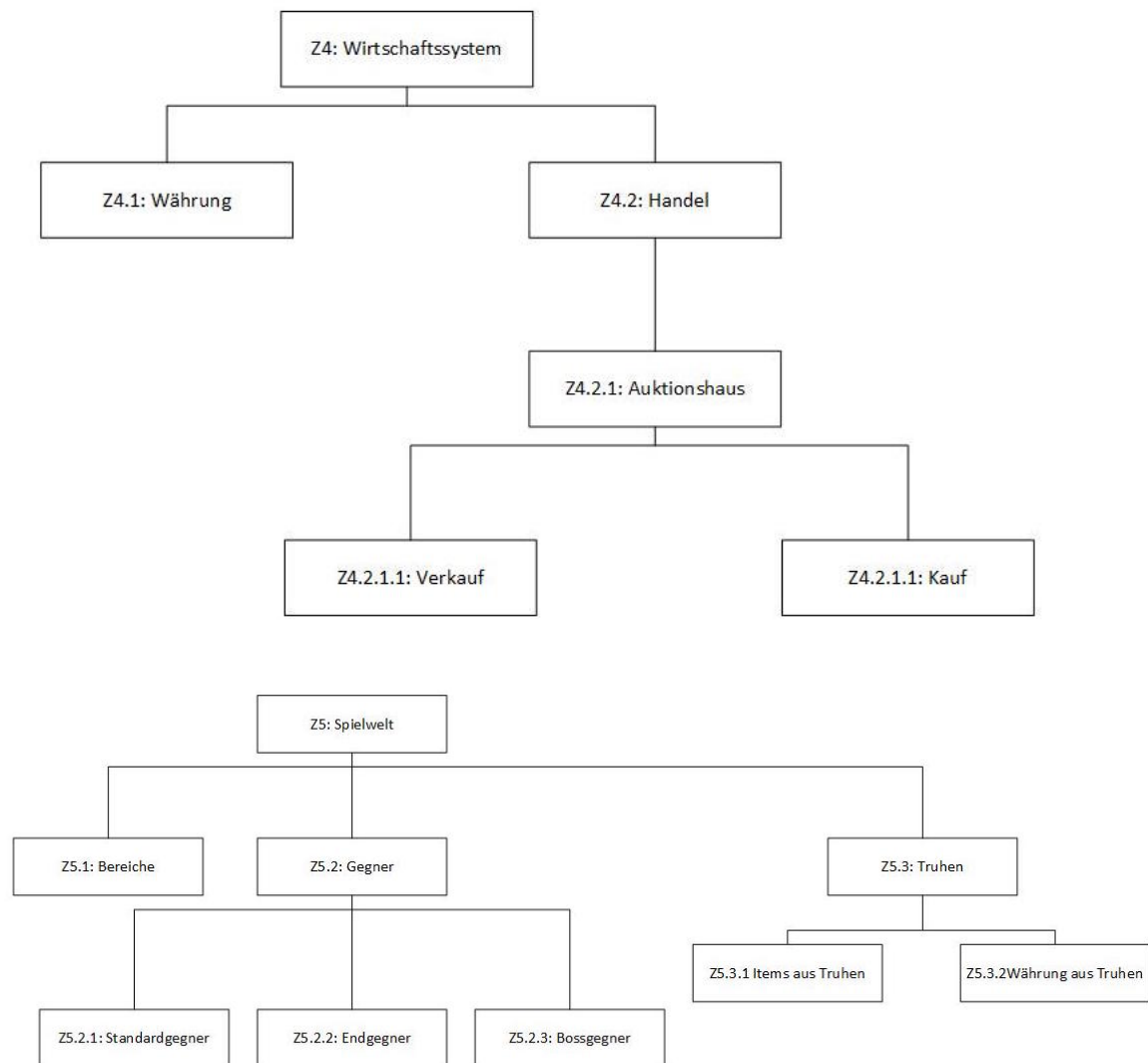


Der Softwareentwicklungsprozess basiert im Rahmen des SEP auf dem angepassten V-Modell. Die Projektmappe ist entsprechend den Phasen des V-Modells aufgebaut. Jede Phase wird Schritt für Schritt im Verlaufe der Veranstaltung bearbeitet und dokumentiert.

2. Anforderungsdefinition

2.1 Zielmodell



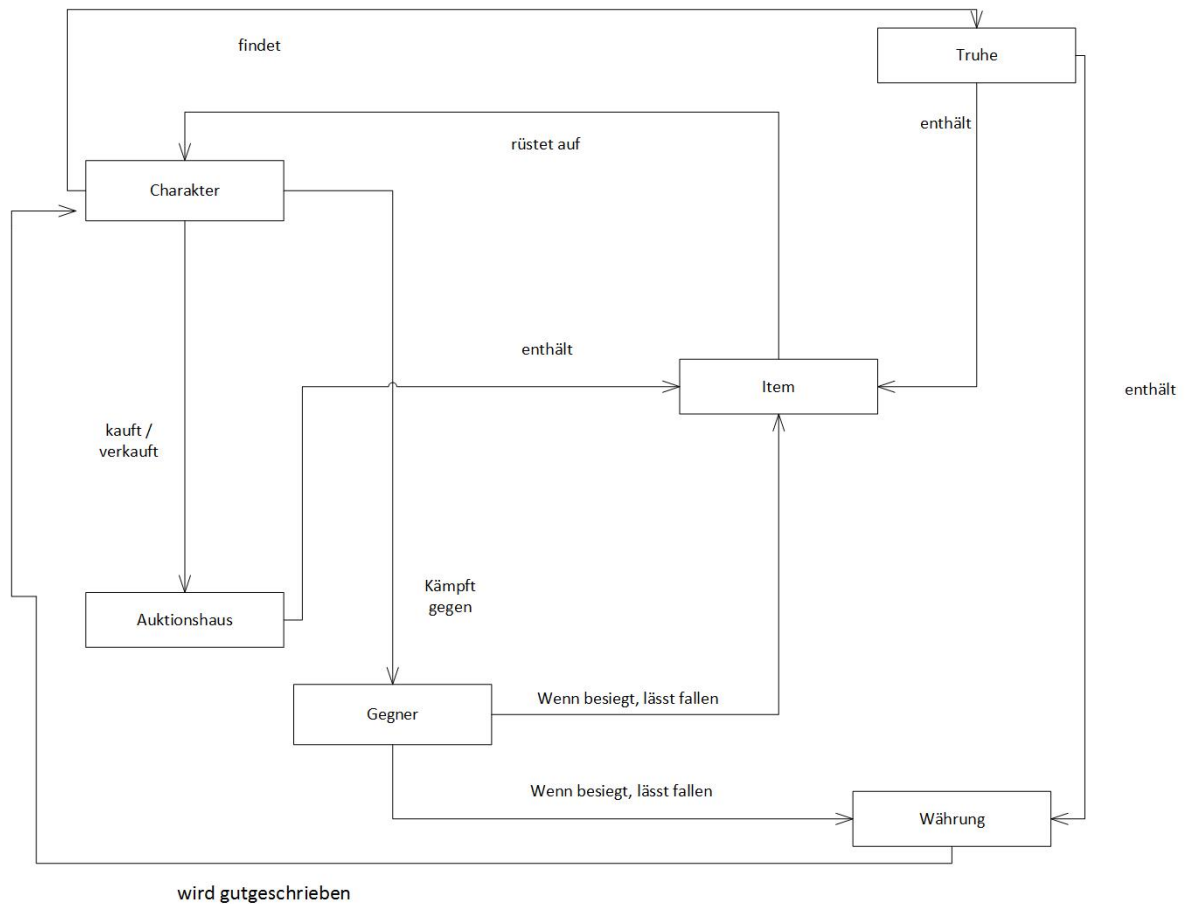


- **Z-1: Benutzerfreundlichkeit**
Das Spiel soll benutzerfreundlich sein.
 - **Z-<L>-<SG>-1.1: Verständlichkeit**
Das Spiel soll selbsterklärend sein.
 - **Z-<L>-<HG>1.1.1 Beschränkte Aktionsvielfalt**
Die Spieler kann 10 verschieden Aktionen ausführen.
 - **Z-<L>-<SG>-1.2: Bedienbarkeit**
Die Bedienung des Spiels soll einfach sein.
 - **Z-<L>-<HG>-1.2.1 Tutorial**
Der Spieler wird in einem Tutorial mit der Bedienung vertraut Gemacht.
- **Z-2: Effizienz**
Das Spiel soll effizient sein.
 - **Z-<L>-<HG>-2.1: Wartezeit**
Das Spiel soll Ladzeiten unter 1 Minute haben.
 - **Z-<L>-<HG>-2.1 Bildrate**
Das Spiel soll mindestens 30 Fps haben.
- **Z-3: Langzeitmotivation**
Das Spiel soll über längere Zeit den Spieler zum Spielen motivieren.

- **Z-<L>-<SG>-3.1: Charakter**
Der Charakter ist die fiktive Spielfigur, die vom Spieler gesteuert wird.
 - **Z-<L>-<HG>-3.1.1: Charakterverbesserung**
Der Spieler soll seinen Charakter ausrüsten können.
 - **Z-<L>-<HG>-3.1.1.1: Levelaufstieg**
Der Charakter erhält für das Besiegen von Gegnern Erfahrungspunkte wodurch sein Level steigt.
 - **Z-<L>-<HG>-3.1.1.2: Ausrüstung**
Der Charakter kann durch Items verbessert werden.
 - **Z-<L>-<HG>-3.1.2: Charakteranpassung**
Der Spieler soll das Aussehen seines Charakters anpassen können.
 - **Z-<L>-<SG>-3.1.3: Charaktertypen**
Der Spieler hat die Wahl zwischen verschiedenen, einzigartigen Charaktertypen.
 - **Z-<L>-<HG>-3.1.3.1: Fähigkeiten**
Jeder Charaktertyp hat einzigartige, erlernbare Fähigkeiten.
 - **Z-<L>-<SG>-3.2: Items**
Der Spieler soll Items sammeln können.
 - **Z-<L>-<HG>-3.2.1: Items von Gegnern**
Der Spieler soll Items erhalten können, wenn er einen Gegner besiegt.
- **Z-<L>-<HG>-3.3: Geschichte**
Das Spiel beinhaltet eine begleitende Geschichte.
- **Z-<L>-<HG>-3.4: Spielstände:**
Der Spieler soll die Möglichkeit haben, seinen Fortschritt in beliebig vielen Spielständen abzuspeichern.
- **Z-4: Wirtschaftssystem**
Das Spiel beinhaltet ein eigenes Wirtschaftssystem.
 - **Z-<L>-<HG>-4.1: Währung**
Das Spiel soll eine eigene Währung haben.
 - **Z-<L>-<HG>-4.2: Handel**
Der Spieler soll Handel mit anderen Spieler treiben können.
 - **Z-<L>-<HG>-4.2.1 Auktionshaus**
Der Spieler soll die Möglichkeit haben, Items im Auktionshaus an andere Spieler zu verkaufen.
 - **Z-<L>-<HG>-4.2.1.1: Verkauf**
Der Spieler soll die Möglichkeit haben, Items im Auktionshaus an andere Spieler zu verkaufen.
 - **Z-<L>-<HG>-4.2.1.2: Kauf**
Der Spieler soll die Möglichkeit haben, Items im Auktionshaus von anderen Spielern zu erwerben.
- **Z-5: Spielwelt**
Das Spiel hat eine Spielwelt, in der der Spieler den Charakter bewegen kann.
 - **Z-<L>-<HG>-5.1: Bereiche**
Das Spiel hat mindestens drei verschiedene Bereiche, die der Spieler durchqueren muss.
 - **Z-<L>-<SG>-5.2: Gegner**
Im Bereich gibt es verschiedene, zu bekämpfende Gegner.
 - **Z-<L>-<HG>-5.2.1: Standardgegner**
Die Bereiche sollen drei verschiedenen Arten von Gegnern

- enthalten.
- **Z-<L>-<HG>-5.2.2: Endgegner**
Am Ende jedes Bereichs soll ein stärkerer Gegner sein.
- **Z-<L>-<HG>-5.2.3: Bossgegner**
Nach Besiegen des dritten Endgegners soll ein besonders starker Gegner erscheinen.
- **Z-<L>-<SG>-5.3: Truhen:**
Der Spieler findet während des Spielens Truhen.
 - **Z-<L>-<HG>-5.3.1: Items aus Truhen**
Der Spieler kann Items aus Truhen erhalten.
 - **Z-<L>-<HG>-5.3.2: Währung aus Truhen**
Der Spieler kann Währung aus Truhen erhalten.

2.2 Spielmodell



2.3 Szenarien

Szenario 1 (Erfüllt Ziel 3.1.1.1)

1. Der Spieler besiegt einen Gegner.
2. Das System zeigt dem Spieler erhaltene Erfahrungspunkte an.
3. Das System zeigt dem Spieler an, dass er ein Level aufgestiegen ist.
4. Das System zeigt dem Spieler seine neuen Statuspunkte an.
5. Der Spieler schließt die Meldung mit einem Klick auf „weiter“.
6. Das System setzt das Spiel fort.

Szenario 2 (Erfüllt Ziel 3.1.3)

1. Der Nutzer weist das System an, einen neuen Charakter zu erstellen.
2. Das System fordert den Nutzer auf, eine von vier Klassen zu wählen.
3. Der Nutzer wählt eine der vier zur Verfügung stehenden Klassen.
4. Das System fordert den Nutzer auf, den Charakter zu benennen.
5. Der Nutzer gibt den Namen des Charakters ein.
6. Das System fordert der Nutzer auf, seine Eingabe zu bestätigen.
7. Der Nutzer bestätigt seine Eingabe.

Szenario 3 (Erfüllt Ziel 3.2)

1. Der Nutzer befiehlt dem System Items zu benutzen.
2. Das System zeigt dem Nutzer eine Liste der Items an.

3. Der Nutzer wählt im System Items aus der Liste aus.
4. Das System weist dem Spieler Items zu.
5. Das System zeigt dem Spieler eine aktualisierte Items Liste an.

Szenario 4 (Erfüllt Ziel 3.2.1)

1. Der Spieler besiegt den Gegner.
2. Das System weist dem Spieler Betrag X Währung zu.
3. Das System zeigt den aktuellen Kontostand an.
4. Der Spieler drückt auf „weiter“.
5. Das System speichert den Spielstand.
6. Das System setzt das Spiel fort.

Szenario 5 (Erfüllt Ziel 3.4)

1. Der Nutzer hat den ersten Bereich beendet.
2. Das System fordert den Nutzer dazu auf weiter zu spielen oder zu beenden.
3. Der Nutzer beendet das Spiel.
4. Das System speichert den aktuellen Spielstand.
5. Der Nutzer verlässt das Spiel.

Szenario 6 (Erfüllt Ziel 4.2.1.1)

1. Der Nutzer weist das System an, das Auktionshaus zu öffnen.
2. Das System zeigt eine Übersicht des Auktionshauses an.
3. Der Nutzer stellt ein Item zum Verkauf bereit.
4. Das System fordert den Nutzer auf, einen Preis festzulegen.
5. Der Nutzer gibt einen Betrag ein.
6. Das System fordert den Nutzer auf, seine Eingabe zu bestätigen.
7. Der Nutzer bestätigt seine Eingabe.
8. Das System stellt das Item im Auktionshaus zum Kauf zur Verfügung

Szenario 7 (Erfüllt Ziel 5.3)

1. Das System zeigt dem Spieler eine Truhe an.
2. Der Spieler weist das System an die Truhe zu öffnen.
3. Das System zeigt dem Spieler den Inhalt der Truhe an.
4. Das System weist dem Charakter des Spielers den Inhalt zu.
5. Das System führt das Spiel fort.

3. Logischer Architekturentwurf

3.1 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>

3.2 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.

3.2.1 <Name Prozess 1>

<Beschreibung Prozess1>

3.2.2 <Name Prozess n>

<Beschreibung Prozess n>

3.3 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.

3.4 Message Sequence Charts

Mit Hilfe von MSC werden Interaktionen zwischen den Elementen des DFDs aus Abschnitt 3.1 modelliert. Zu jedem Szenario aus Abschnitt 2.3 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.

3.4.1 bMSCs

bMSC-1: <Name des bMSC>

<Grafik des bMSC>

bMSC-n: <Name des bMSC>

<Grafik des bMSC>

3.4.2 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>	

3.4.3 hMSC

<Grafik des hMSC>

4. 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.

4.1 GUI-Papierprototyp

4.1.1 Screen „<Name des Screens>“

<Scan des Screen-Papierprototypen>

4.2 Technisches Konzept

<Grafik des technischen Konzepts>

4.2.1 <Name Komponente 1>

<Beschreibung zu Komponente 1>

4.2.2 <Name Komponente n>

<Beschreibung zu Komponente n>

4.3 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>

4.3.1 Komponentenbeschreibung

<Name Komponente 1>

<Beschreibung zu Komponente 1>

<Name Komponente n>

<Beschreibung zu Komponente n>

4.3.2 Interfacebeschreibung

<Name Interface 1>

<Beschreibung zu Interface 1>

<Name Interface n>

<Beschreibung zu Interface n>

5. Testartefakte

5.1 Modultest

5.1.1 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	

5.1.2 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	

5.2 Systemtest

5.2.1 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		
...		

5.2.2 Testergebnisse

Testprotokoll Systemtestfall 1 (<1. Testdurchführung>)

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