

Diplomarbeit

PHLOWR

Unity Projectmanagement Software

Studiengang zum
Dipl. Informatiker HF Applikationsentwicklung
an der TEKO Bern

Klasse:
B-TIN-19-T-a

Eingereicht von
Sascha Dubois

17. September 2022

Prüfungsexperte:

Herr Jürg Hess

Inhaltsverzeichnis

1	Initialisierung	1
1.1	Lebenslauf	2
1.2	Vorwort	3
1.3	Planung	4
1.3.1	Initialisierung	4
1.3.2	Analyse	5
1.3.3	Konzept	6
1.3.4	Umsetzung	6
1.3.5	Deployment	7
2	Analyse	8
2.1	Ausgangslage	8
2.2	Stakeholder	9
2.3	Ziele	10
2.4	Anforderungen	11
2.5	Abgrenzung	12
2.6	Risikomanagement	13
2.7	Wirtschaftlichkeit	14
3	Konzept	15
3.1	Geschäftsnwendungsfälle	15
3.2	Kontextdiagramm	16
3.3	Use Case Diagramme	17
3.4	Sequenzdiagramme	18
3.5	State Diagramme	19
3.6	Systemarchitektur	20
3.7	Mockup	21
4	Realisierung	22
4.1	Entwicklungsumgebung	22
4.2	Codeconventions	22
5	Anhang	b
5.1	Themeneingabe	c

1 Initialisierung

1.1 Lebenslauf

1.2 Vorwort

1.3 Planung

Es wird mit einer abgewandelten Form des Wasserfallmodells geplant. Dabei wird das Projekt in 5 Phasen aufgeteilt:

1. Initialisierung
2. Analyse
3. Konzept
4. Umsetzung / Realisierung
5. Veröffentlichung

In den einzelnen Phasen werden Tasks geplant, welche eine geschätzte Dauer zur umsetzung haben. Anhand dieser Schätzung wird die Dauer der einzelnen Phasen geplant. Zusätzlich wird pro Phase ein Buffer eingeplant.

Um das Projekt zu planen habe ich mir zuerst die einzelnen Phasen in einer Gantt-Diagramm-form ausgelegt. Anschliessend bin ich Phase für Phase durchgegangen und habe mir Tasks erstellt, anhand der Schätzung der Dauer habe ich überprüft, ob die Phasenaufteilung passt.

Die Tasks sind in anzahl Tagen geplant, die länge der Tasks ist jedoch auch von meinem Zeitkontingent an den entsprechenden Tagen abhängig, so kann z.B. ein 6h Task 3 Tage andauern, da ich nur 2h pro Tag verplanen konnte. in der Woche vom 17.10.2022 habe ich eine Intensiv Woche geplant. In dieser Woche habe ich die Möglichkeit vollzeit an der Diplomarbeit zu arbeiten. Daher ist in dieser Woche auch der Hauptteil der Umsetzung geplant. Nach dem ich mir die Gantt-Diagramme gemacht und die einzelnen Tasks tabellarisch aufgelistet habe. Habe ich mir in einer Zeit-Tracking App (Toggle) alle Tasks angelegt. In Toggle erfasse ich somit die geleistete Arbeitszeit zu jeweiligem Task.

Dies soll mir ermöglichen verzögerungen schnell erkennen zu können und entsprechende Massnahmen einzuleiten.

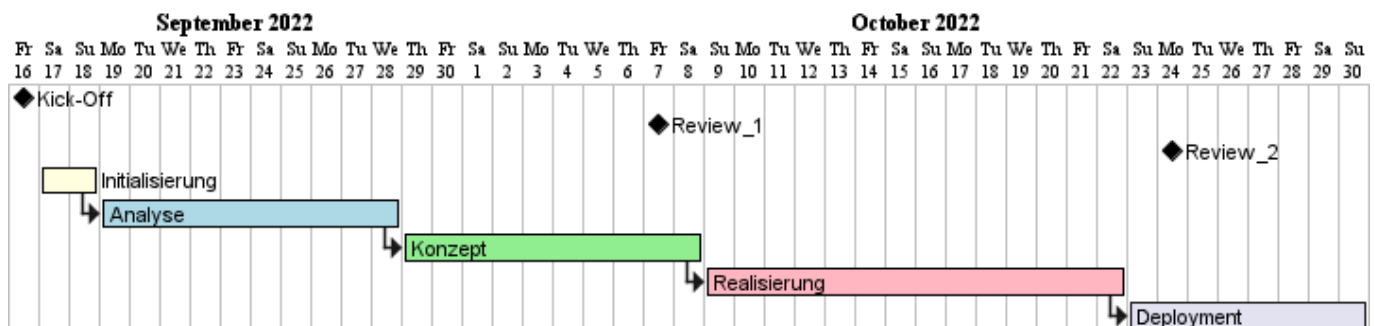


Abb. 1: GANTT Grobplanung

1.3.1 Initialisierung

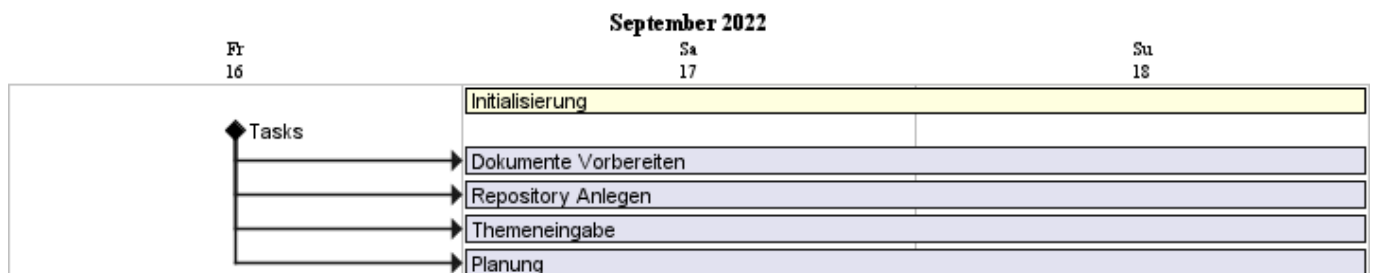


Abb. 2: GANTT Initialisierung

Task	Beschreibung	Geplant	Ist	Delta
Dokumente Vorbereiten	LaTeX-Dokument aufsetzen, PlantUML Diagrammvorlagen erstellen	4h	3h	-1h
Repository Anlegen	GitHub Repos für Dokumentation und Projekt anlegen	1h	1h	0h
Themeneingabe	Themeneingabe in Dokumentation einfügen	1h	0.5h	-0.5h
Planung	Projektplanung Durchführen	6h	3.5h	-2.5
Total		12h	8h	-4h

Tabelle 1: Initialisierungs Tasks

1.3.2 Analyse

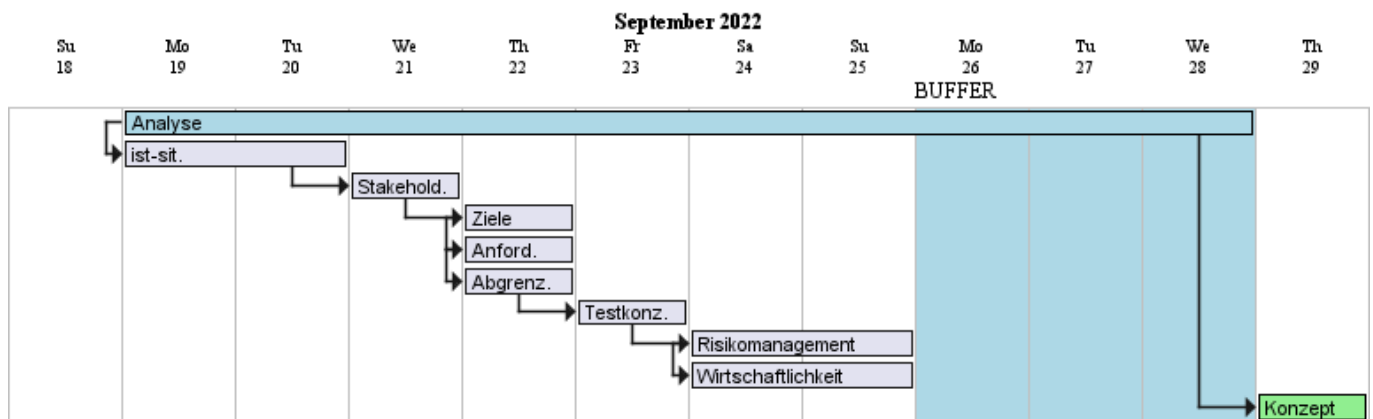


Abb. 3: GANTT Analyse

Task	Beschreibung	Geplant	Ist	Delta
Ist-Situation	Ausgangslage analysieren	4h	0h	
Stakeholderanalyse	Stakeholder erfassen und gewichten	3h	0h	
Ziele	Zielsetzungen aus Themeneingabe präzisieren	2h	0h	
Anforderungen	Anforderungen aus Themeneingabe präzisieren	2h	0h	
Abgrenzung	Abgrenzungen der Lösung definieren	3h	0h	
Testkonzept	Testkonzept erstellen	4h	0h	
Risikoanalyse	Risiken identifizieren und Massnahmen definieren	6h	0h	
Wirtschaftlichkeitsanalyse	Wirtschaftlichkeit der Lösung eruieren	4h	0h	
Total		28h	0h	0h

Tabelle 2: Analyse Tasks

1.3.3 Konzept

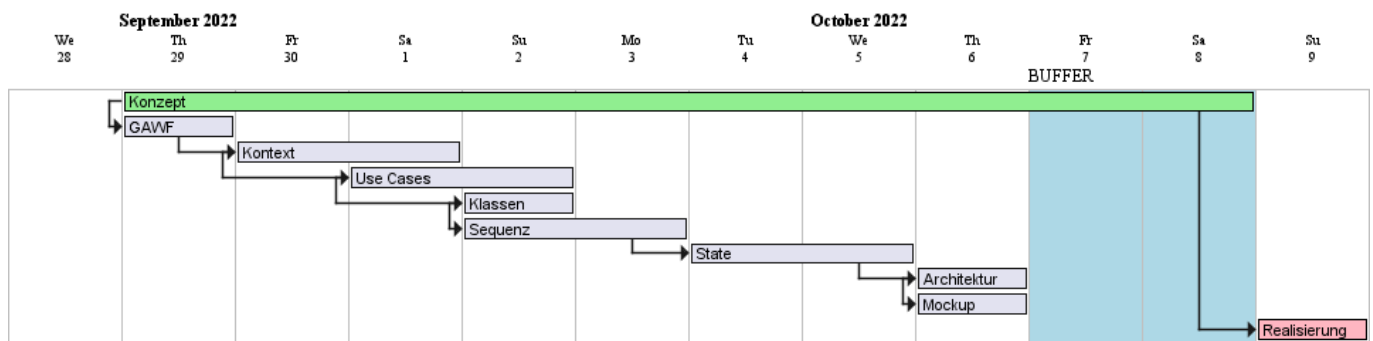


Abb. 4: GANTT Konzept

Task	Beschreibung	Geplant	Ist	Delta
GAWF	Geschäftsanwendungsfälle beschreiben	6h	0h	
Kontext	Kontextdiagramm erstellen	3h	0h	
Use Cases	Use-Case-Diagramme erstellen	5h	0h	
Klassen	Klassendiagramm erstellen	2h	0h	
Sequenz	Sequenzdiagramme erstellen	4h	0h	
State	Statediagramme erstellen	3h	0h	
Architektur	Planung der Software Architektur	3h	0h	
Mockup	Mockup der Applikationen erstellen	5h	0h	
Total		31h	0h	0h

Tabelle 3: Konzept Tasks

1.3.4 Umsetzung

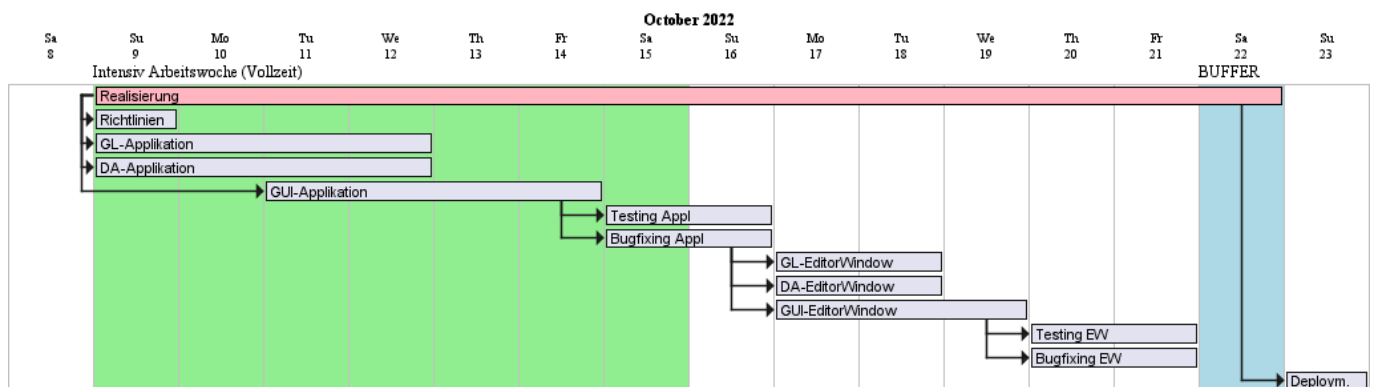


Abb. 5: GANTT Realisierung

Task	Beschreibung	Geplant	Ist	Delta
Richtlinien	Coding Richtlinien definieren	2h	0h	
GL-Applikation	Geschäftslogik der Applikation implementieren	11h	0h	
DA-Applikation	Data handling der Applikation implementieren	6h	0h	
GUI-Applikation	GUI der Applikation implementieren	25h	0h	
Testing Applikation	Testen der Applikation anhand Testkonzept	6h	0h	
Bugfixing Applikation	Allfällige Bugs fixen / Buffer	8h	0h	
GL-EditorWindow	Geschäftslogik des EditorWindows implementieren	4h	0h	
DA-EditorWindow	Data handling des EditorWindows implementieren	4h	0h	
GUI-EditorWindow	GUI des EditorWindows implementieren	8h	0h	
Testing EditorWindow	Testen des EditorWindows anhand Testkonzept	4h	0h	
Bugfixing EditorWindow	Allfällige Bugs fixen / Buffer	6h	0h	
Total		84h	0h	0h

Tabelle 4: Umsetzung Tasks

1.3.5 Deployment

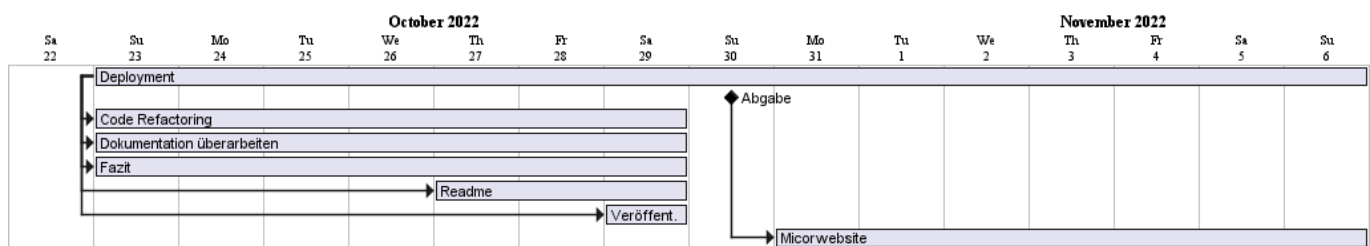


Abb. 6: GANTT Deployment

Task	Beschreibung	Geplant	Ist	Delta
Code Refactoring	Code cleanup / Optimisierungen	5h	0h	
Dokumentation Überarbeiten	Fehlerkorrektur, Format	5h	0h	
Readme	Readme für GitHub-Page erstellen	3h	0h	
Veröffentlichen	GitHub Projekt veröffentlichen	2h	0h	
Total		15h	0h	0h

Tabelle 5: Deployment Tasks

2 Analyse

In den folgenden Kapiteln, werden die punkte, welche bereits in der Themeneingabe definiert wurden, präzisiert.

2.1 Ausgangslage

Es gibt bereits etliche Applikationen zum managen von Projekten

2.2 Stakeholder

- TEKO
- Indieentwickler
- Konsumenten von Indiespielen

2.3 Ziele

2.4 Anforderungen

2.5 Abgrenzung

2.6 Risikomanagement

2.7 Wirtschaftlichkeit

3 Konzept

3.1 Geschäftsnwendungsfälle

3.2 Kontextdiagramm

3.3 Use Case Diagramme

3.4 Sequenzdiagramme

3.5 State Diagramme

3.6 Systemarchitektur

3.7 Mockup

4 Realisierung

4.1 Entwicklungsumgebung

4.2 Codeconventions

Abbildungsverzeichnis

1	GANTT Grobplanung	4
2	GANTT Initialisierung	4
3	GANTT Analyse	5
4	GANTT Konzept	6
5	GANTT Realisierung	6
6	GANTT Deployment	7

Tabellenverzeichnis

1	Initialisierungs Tasks	5
2	Analyse Tasks	5
3	Konzept Tasks	6
4	Umsetzung Tasks	7
5	Deployment Tasks	7

5 Anhang

5.1 Themeneingabe



Themeneingabe zur Diplomarbeit

Projektmanagement-Applikation für Unity3D

B-TIN-19-T-a

Standort: Bern

Eingereicht von

Dubois Sascha

Bergernschachenweg3, 3436 Zollbrück

saschadub@gmail.com

0794615088

Eingereicht am: 29.05.2022

Bewilligt: ☐ Ja / ☐ Nein

Datum: _____ / Unterschrift: _____

Begründung:

A large, solid gray rectangular box intended for the student to provide a justification for their submission.

Änderungsverzeichnis

Datum	Änderung	Name
14.08.2022	Präzisierungen	Sascha Dubois

Management Summary

Themenbeschreibung

Entwicklung einer Open-Source Projektmanagement-Applikation für Videospiel-Einzelentwickler und kleine Entwicklerteams welche mit Unity3D arbeiten. Dabei sollen die Wichtigsten Funktionen direkt in Unity selber verfügbar sein

Kunde

Potenzielle Kunden sind alle Einzelentwickler / Entwicklerteams, welche mit Unity3D arbeiten

Erfolgskriterien

- Eine WPF-Applikation wurde den Anforderungen (siehe: [Ziele und Anforderungen](#)) entsprechend entwickelt.
- Ein in Unity integriertes Benutzerinterface (Custom Editor) in welchem Aktivitäten zum aktuellen Projekt angesehen/bearbeitet werden können wurde den Anforderungen (siehe: [Ziele und Anforderungen](#)) entsprechend entwickelt.
- Es Liegt eine vollständige Dokumentation des Projektes vor
- Die Applikation ist als Open-Source Software lizenziert und zum Download verfügbar

Inhaltsverzeichnis

Änderungsverzeichnis	2
1 Zweck des Dokuments.....	6
2 Ausgangslage.....	6
2.1 Unity3D	6
2.2 Beschreibung	6
3 Aufgabenstellung.....	6
4 IST-Zustand.....	6
5 Abgrenzung	7
5.1 Generelle Abgrenzung der Arbeit	7
5.2 Geleistete Vorarbeiten.....	7
5.3 Verwendung Erzeugnisse Dritter	7
6 Rahmenbedingungen	7
6.1 Prozessbezogene Rahmenbedingungen	7
6.2 Produktbezogene Rahmenbedingungen	7
7 Ziele und Anforderungen	8
7.1 Ziele.....	8
7.2 Funktionale Anforderungen.....	8
7.2.1 Applikation	8
7.2.2 In Unity integriertes Benutzerinterface (Custom Editor)	8
7.3 Nicht-funktionale Anforderungen	8
8 Projektmanagement.....	9
8.1 Vorgehensmethodik.....	9
8.2 Grobe Zeitplanung	10
9 Nutzen	11
10 Skizze	12
11 Nachträgliche Präzisierungen.....	13
11.1 Umfang des Projektmanagements	13
11.2 Features von Unity3D	13

Glossar

Begriff	Bedeutung
Unity3D / Unity	Entwicklungsumgebung / Engine für Videospiele
Unity-Collaborate	Angebot von Unity zur verteilten Versionsverwaltung von Dateien / Projekten
Custom Editor	Mit «Custom Editor» ist eine Erweiterung des Unity-Editors gemeint. Unity bietet die Möglichkeit, solche Erweiterungen erstellen zu können, standardmässig an. Unity - Manual: Custom Editors (unity3d.com)
Git	Freie Software zur verteilten Versionsverwaltung von Dateien
GitHub	Netzbasierter Dienst zur Versionsverwaltung für Software-Entwicklungsprojekte
Open Source	Als Open Source wird Software bezeichnet, deren Quelltext öffentlich und von Dritten eingesehen, geändert und genutzt werden kann.

1 Zweck des Dokuments

Genaue Beschreibung des zu Bearbeitenden Themas für die Diplomarbeit.

2 Ausgangslage

2.1 Unity3D

«Unity ist eine Laufzeit- und Entwicklungsumgebung für Spiele (Spiel-Engine) des Unternehmens Unity Technologies mit Hauptsitz in San Francisco. Ziel-Spieleplattformen sind neben PCs (PC und Mac) auch Spielkonsolen, mobile Geräte und Webbrowser. Die Entwicklungsumgebung, die die Entwicklung von Computerspielen und anderer interaktiver 3D-Grafik-Anwendungen ermöglicht, gibt es für Windows, Linux (nur Beta) und macOS.» (Wikipedia Unity_(Spiel-Engine))

2.2 Beschreibung

Gamedesign unterscheidet sich in einigen Punkten von der herkömmlichen Softwareentwicklung und wird eher als ein kreativer Prozess angesehen. Vor allem bei kleinen Entwicklerteams und Einzelentwickler manifestiert sich dies besonders: Erfahrungsgemäss wird nur wenig Projektmanagement betrieben und vieles wird spontan entschieden, der Vergleich zum Künstler, der sein Bild nach Gefühl mahlt, ist hier sehr passend. Leider hat dieses Vorgehen einige Nachteile, so werden viele Projekte bekanntermassen nicht beendet oder unfertig publiziert, schwerwiegende Bugs liegen ebenfalls an der Tagesordnung. Zumindest ein grundlegendes Projektmanagement sollte jeder Entwickler führen, denn Videospiele können sehr schnell zur sehr komplexen Code-Suppe werden, bei welcher niemand mehr den Überblick behalten kann.

Ein Grund für das fehlende Projektmanagement ist vermutlich das fehlende Tool dafür. Es gibt zwar viele Projektmanagement-Tools, diese sind jedoch meist für grössere Teams ausgelegt und müssen je nachdem auch bezahlt werden. Der grösste Faktor ist, meiner Meinung nach, jedoch der Mehraufwand den ein solches Tool generiert, denn diese Tools müssen separat gestartet und verwaltet werden. Es wäre doch viel einfacher, wenn ich als Entwickler einfach Unity starten kann und ich alle meine Tasks zum geöffneten Projekt direkt in Unity selber verwalten könnte. Genau dies ist der Ansatz, den ich für meine Diplomarbeit verfolgen möchte.

3 Aufgabenstellung

Es wird eine eigenständige Applikation entwickelt, in welcher Unity-Projekte geplant und verwaltet werden können. Die Applikation soll als Open-Source Software lizenziert und auf GitHub veröffentlicht werden. Die Applikation soll dem Nutzer die Möglichkeit bieten, Projekte nach eigens ausgewähltem Vorgehensmodell zu planen, dass heisst, es wird ein generischer Aufbau der Projekte verlangt.

Ein in Unity integriertes Benutzerinterface (Custom-Editor), in welchem das geöffnete Projekt verwaltet werden kann, ist ebenfalls Teil der Aufgabenstellung.

Projektinformationen, welche von der Applikation erstellt werden, sollen direkt im Projektordner selbst abgespeichert werden, damit diese via Git oder Unity-Collaborate synchronisiert werden können.

Ebenfalls soll es eine Möglichkeit geben, Aufgaben bestimmten Benutzern zuweisen zu können und Meilensteine zu definieren.

4 IST-Zustand

Einzelentwickler und kleine Entwicklerteams verwenden zurzeit (wenn überhaupt) die herkömmlich verfügbaren Tools, um ihre Projekte zu verwalten.

5 Abgrenzung

5.1 Generelle Abgrenzung der Arbeit

Im Rahmen der Diplomarbeit, gibt es keine Möglichkeit Daten auf einer Datenbank zu persistieren. Daten werden lediglich serialisiert im entsprechenden Unity-Projektordner abgespeichert. Die Softwarelösung wird ausschliesslich für Windows entwickelt.

5.2 Geleistete Vorarbeiten

Vorab wurden Tests zwecks der Machbarkeitsüberprüfung durchgeführt. Für das Endprodukt selbst oder dessen Dokumentation werden jedoch keine Vorarbeiten durchgeführt.

5.3 Verwendung Erzeugnisse Dritter

Für die Integration in Unity, werden die von Unity bereitgestellten Bibliotheken verwendet. Verwendung von frei verfügbaren Bibliotheken (z.B. NuGets) für Darstellung und Design wird in der Projektdokumentation beschrieben.

6 Rahmenbedingungen

6.1 Prozessbezogene Rahmenbedingungen

- Das Projekt wird als Einzelarbeit durchgeführt
- Es wird eine vollumfängliche Dokumentation erstellt
- Da das Produkt anschliessend als Open-Source Software lizenziert und veröffentlicht werden soll, wird zudem eine englische Dokumentation des Quellcodes erstellt, diese Dokumentation soll anschliessend als Wiki in GitHub verfügbar sein.

6.2 Produktbezogene Rahmenbedingungen

- Die Softwarelösung wird als WPF-Applikation umgesetzt
- Als Entwicklungsumgebung wird VisualStudio verwendet

7 Ziele und Anforderungen

7.1 Ziele

- Die Lösung ist als Open-Source Software gratis verfügbar
- Die Applikation kann von Einzelentwicklern als auch Entwicklerteams verwendet werden
- Die Applikation kann lokal installiert werden, es wird kein Server benötigt

7.2 Funktionale Anforderungen

7.2.1 Applikation

- Unity-Projekte können in der Applikation geplant werden
- Aktivitäten können in der Applikation erstellt, bearbeitet und gelöscht werden
- Aktivitäten können terminiert werden (Start- und Enddatum) *
- Aktivitäten können kommentiert werden*
- Aktivitäten können priorisiert werden*
- Aktivitäten können punkte zugewiesen werden (z.B. Story Points oder Stunden) *
- Aktivitäten können gruppiert werden (z.B. zu Phasen)
- Gruppen von Aktivitäten können kommentiert werden*
- Gruppen können terminiert werden (Start- und Enddatum) *
- Gruppen können priorisiert werden *
- Gruppen können punkte zugewiesen werden (z.B. Story Points oder Stunden) *
- Bestehende Unity-Projekte können der Applikation hinzugefügt werden
- Meilensteine können in der Applikation geplant werden
- Aktivitäten können Benutzern zugewiesen werden
- Wird ein Unity-Projekt der Applikation hinzugefügt, so wird das entsprechende Custom-Editor-Script dem Projektordner hinzugefügt

7.2.2 In Unity integriertes Benutzerinterface (Custom Editor)

- Aktivitäten zum aktuell geöffneten Projekt können erstellt werden
- Aktivitäten zum aktuell geöffneten Projekt können bearbeitet werden
- Projektinformationen zum aktuell geöffneten Projekt (Beschreibung, Meilensteine usw.) können angesehen werden

7.3 Nicht-funktionale Anforderungen

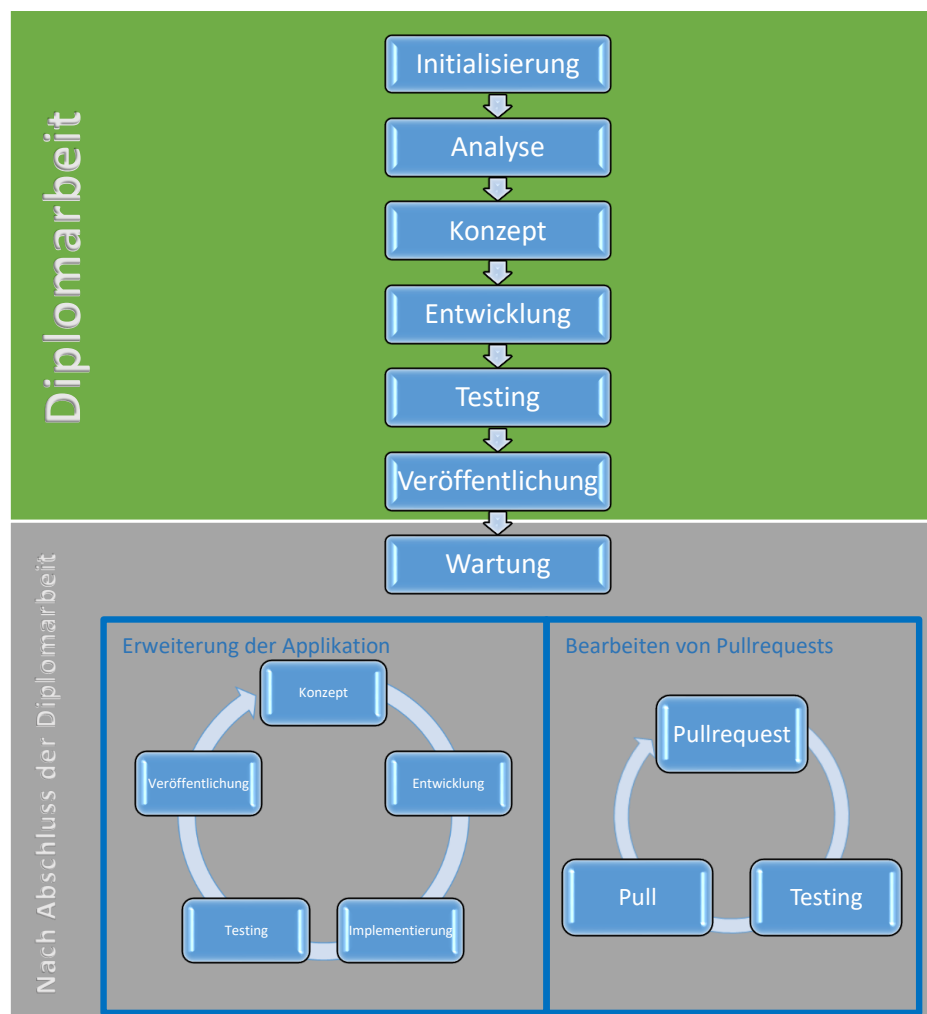
- Projektdaten werden mit dem Projekt zusammen abgespeichert und synchronisiert (z.B. via Git)

* Nachträglich ergänzt 14.08.2022

8 Projektmanagement

8.1 Vorgehensmethodik

Da ich schon einige Erfahrung mit der Entwicklung von WPF-Applikationen sowie mit dem Arbeiten mit Unity habe, entscheide ich mich für das Wasserfallmodell als Vorgehensmodell für das Entwickeln der Applikation im Rahmen der Diplomarbeit. Nach Abschluss der Diplomarbeit werden weitere Features und Verbesserungen in einem Agilen verfahren hinzugefügt.



8.2 Grobe Zeitplanung

Phase	Aktivitäten (Stichpunkte)	Zeitraum	Aufwand
Initialisierung	<ul style="list-style-type: none"> - Detaillierte Projektplanung 	16.09.2022 – 18.09.2022	8-10h
Analyse	<ul style="list-style-type: none"> - Ziele spezifizieren - Anforderungen erfassen - Risikoanalyse - Dokumentation 	19.09.2022 – 25.09.2022	24-28h
Konzept	<ul style="list-style-type: none"> - Modellierung mittels UML - GUI-Mockup für Applikation und Editor - Testkonzept erstellen - Dokumentation 	26.09.2022- 05.10.2022	28-32h
Entwicklung	<ul style="list-style-type: none"> - Entwickeln der WPF-Applikation - Entwickeln des Custom-Editors - Dokumentation 	06.10.2022- 23.10.2022	52-56h
Testing	<ul style="list-style-type: none"> - Testen nach Testkonzept - Bugfixing - Dokumentation 	24.10.2022- 29.10.2022	24-28h
Veröffentlichung	<ul style="list-style-type: none"> - Readme-Datei erstellen - Open-Source-Lizenz importieren - GitHub Repo veröffentlichen - Dokumentation 	30-31.10.2022	8-10h
Präsentation		31.10.2022 – 11.11.2022	

9 Nutzen

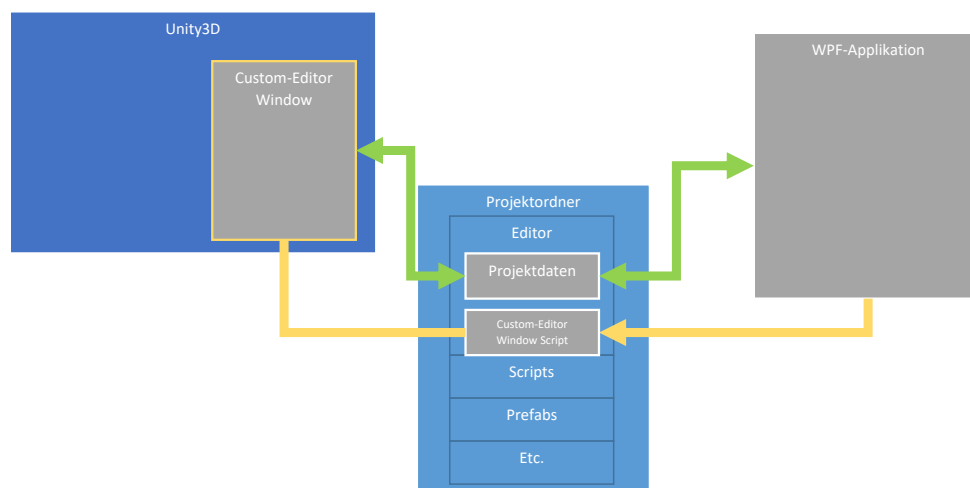
Persönlich:

In meiner Freizeit entwickle ich selbst Videospiele, leider ist das Management dieser Projekte immer sehr frustrierend, da es einfach keine gute Lösung dafür gibt. Ich erhoffe mir persönlich daher einen grossen Nutzen der Lösung und somit auch mehr Spass und Erfolg beim Entwickeln von Videospielen.

Allgemein:

In Online-Foren liest man immer wieder dasselbe Problem: Ein neues Projekt wird aufgrund einer eigentlich guten Idee begonnen aber nach einiger Zeit wird das Spiel so komplex, dass man die Übersicht verliert und das ganze Projekt begräbt. Das ist wohl die Geschichte, die die meisten Einzelentwickler kennen und somit erhoffe ich mir, dass meine Lösung zu einigen vollendeten Spielen und somit auch zu einigen Spielstunden beitragen wird.

10 Skizze



Die Applikation speichert alle Daten zum jeweiligen Projekt in dem entsprechenden Projektordner auf dem Dateisystem, in einem Custom-Editor in Unity können diese Daten ebenfalls angezeigt und bearbeitet werden. Das Custom-Editor-Script wird, beim Importieren des Projektes, von der Applikation im Ordner des Projektes (/Projektname/Assets/Editor/CustomEditorScript) abgelegt. Alle Scripts die in dem /Editor-Ordner abgelegt sind werden von Unity vom Build automatisch exkludiert.

11 Nachträgliche Präzisierungen

11.1 Umfang des Projektmanagements

«Was beinhaltet das Projektmanagement ausser Aktivitäten (und Gruppierung in Phasen), Meilensteine und Ressourcenallokation noch?»

Siehe Änderungen in [«Funktionale Anforderungen»](#)

- Aktivitäten können terminiert werden (Start -und Enddatum)
- Aktivitäten können kommentiert werden
- Aktivitäten können priorisiert werden
- Aktivitäten können punkte zugewiesen werden (z.B. Story Points oder Stunden)
- Gruppen von Aktivitäten können kommentiert werden
- Gruppen können terminiert werden (Start- und Enddatum)
- Gruppen können priorisiert werden
- Gruppen können punkte zugewiesen werden (z.B. Story Points oder Stunden)

11.2 Features von Unity3D

«Welche Features stellt Unity3D für die Integration der Applikation zur Verfügung?»

Unity bietet die Möglichkeit sogenannte «Custom Editors» und «Editor Windows» zu erstellen. «Custom Editors» können als Art Erweiterungen der normalen Editoroberfläche angesehen werden und werden komplett in C# geschrieben. Unity nennt dieses Verfahren zum Erstellen von GUI-Elementen «IMGUI» (Immediate Mode GUI system).

«Editor Windows» sind, wie der Name bereits sagt, eigene Editor Fenster welche innerhalb des Unity Editors geöffnet und angedockt werden können.

Unity stellt die üblichen GUI-Elemente wie Buttons, Labels, TextBoxen usw. zu Verfügung. Jedes GUI-Element verfügt über einen GUI-Style welcher individuell angepasst oder erstellt werden kann, so sind die Möglichkeiten zur Erzeugung eigener Editor-Erweiterungen recht umfangreich.

Details zu IMGUI: [Unity - Manual: Immediate Mode GUI \(IMGUI\) \(unity3d.com\)](#)
[Unity - Manual: IMGUI Basics \(unity3d.com\)](#)

Details zu «Custom Editors»: [Unity - Manual: Custom Editors \(unity3d.com\)](#)

Details zu «Editor Windows»: [Unity - Manual: Editor Windows \(unity3d.com\)](#)

Details zu GUI-Styles: [Unity - Manual: GUI Style \(IMGUI System\) \(unity3d.com\)](#)

Des Weiteren verfügt Unity über recht umfangreiche Serialisations-Verfahren, welche das effiziente Speichern und Laden von serialisierten Daten zur Laufzeit ermöglicht.

Details zur Serialisation: [Unity - Manual: Script serialization \(unity3d.com\)](#)