

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:

## 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, standartmässig an. <a href="https://unity3d.com/manual/custom-editors">Unity - Manual: Custom Editors (unity3d.com)</a> |
| 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)

---

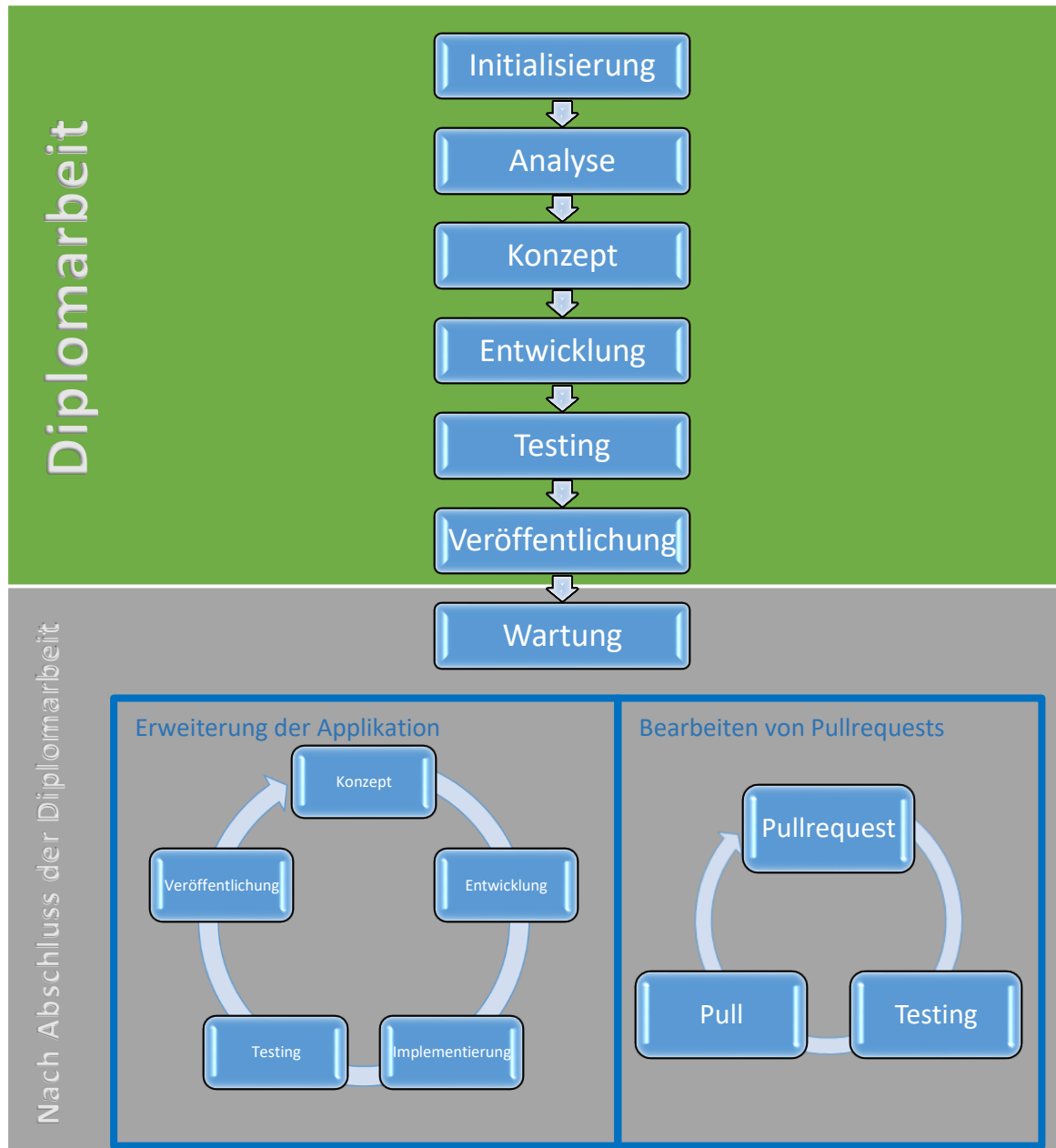
\* Nachträglich ergänzt 14.08.2022



## 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 Agileren verfahren hinzugefügt.



## 8.2 Grobe Zeitplanung

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

## 9 Nutzen

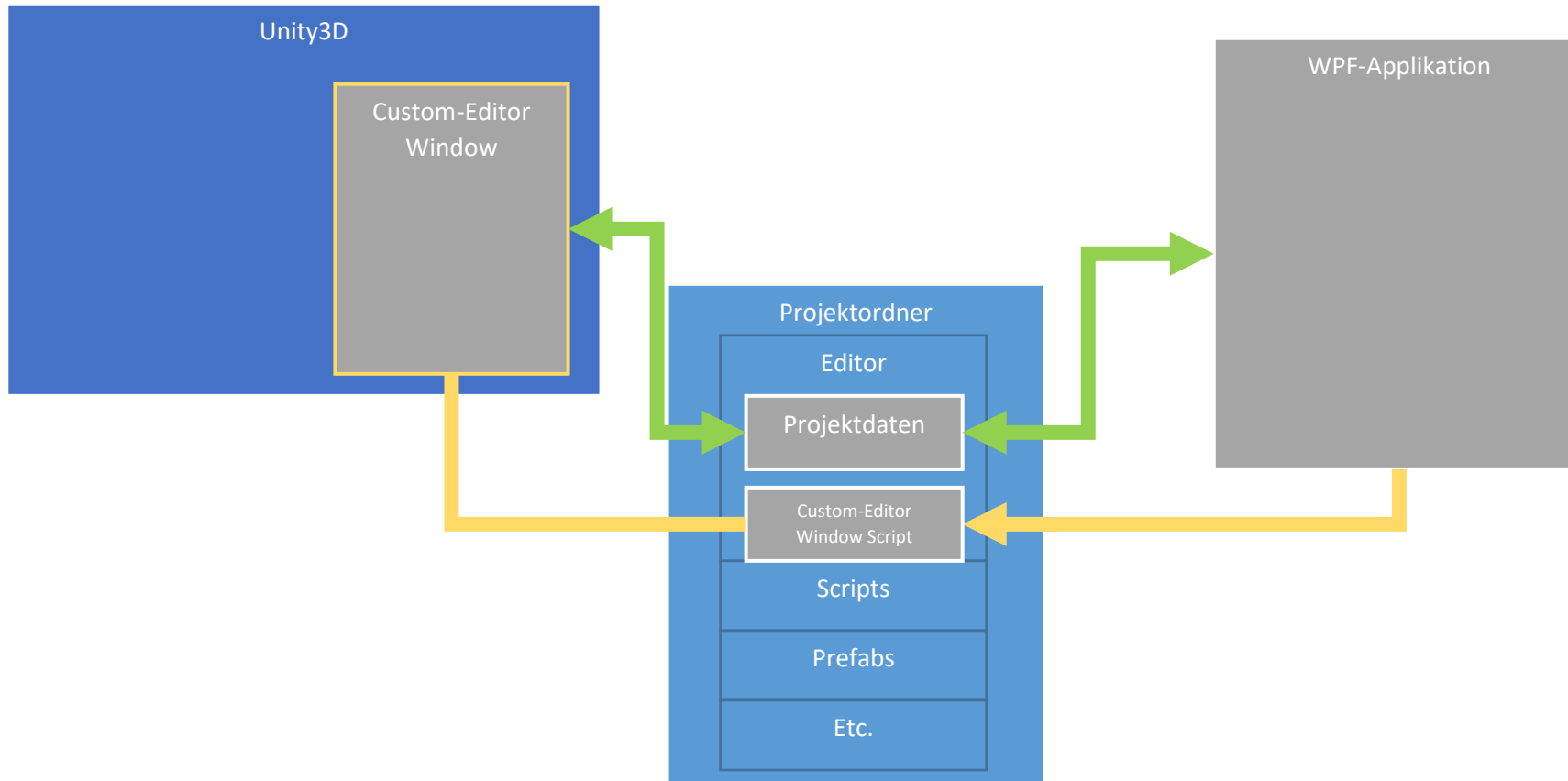
### **Persönlich:**

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-Skript 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 andockt 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 \(unity3d.com\)](#)