



---

# Project Manager Game weiterprogrammieren

---

## Bachelorarbeit

STUDIENGANG INFORMATIK  
OST-OSTSCHWEIZER FACHHOCHSCHULE  
CAPMUS RAPPERSWIL-JONA

---

**Herbstsemester 2020**

---

Autoren: Dario Caluzi, Martin Iten  
Betreuer: Claudio Fuchs  
Experte: Michael Spörri

# Änderungsverlauf

Datum	Version	Änderung	Autor
2020-09-21	1	Initiale Version des Dokumentes	Martin Iten
2020-12-04	2	Praktische Fragestellung	Martin Iten
2020-12-10	3	Anforderungen	Dario Caluzi
2020-12-15	4	Architektur & Deployment	Dario Caluzi
2020-12-16	5	Projektdokumentation	Martin Iten
2020-12-18	6	Testing	Dario Caluzi
2020-12-26	7	Installationsanleitung & Benutzerhandbuch	Dario Caluzi
2020-12-29	8	Abstract	Martin Iten & Dario Caluzi
2020-12-30	9	Offene Issues & Screenshots	Dario Caluzi
2021-01-03	10	Management Summary	Martin Iten
2021-01-15	11	Überarbeitung & Finalisierung	Dario Caluzi & Martin Iten

Tabelle 1: Changelog

## Abstract

„PM Game“ ist eine interaktive Lernsimulation, welche in einer Studienarbeit an der Ostschweizer Fachhochschule (OST) entwickelt wurde. In dieser Bachelorarbeit wurde die Software weiterentwickelt und optimiert. Die Applikation richtet sich in erster Linie an Studenten des Moduls Projekt- und Qualitätsmanagement (PmQm) an der OST und dient dazu, die in der Vorlesung erlernten Inhalte in einem simulierten IT-Projekt zu üben. Die Studierenden werden so spielerisch ins Thema Projektmanagement eingeführt und können die Auswirkungen ihrer Entscheide als Projektleiter mitverfolgen.

Bis auf eine User Story wurden alle Anforderungen des Projekts erfüllt. Die neuen Funktionen umfassen unter anderem ein umfangreicheres Risikomanagement sowie ein Messaging System. Auch die nicht-funktionalen Anforderungen wurden grundsätzlich erfüllt. So wurde die bestehende Applikation von Grund auf überarbeitet um eine bessere Performance zu ermöglichen und die Wartbarkeit der Software zu verbessern. Die Software wurde dann in mehreren Live Tests getestet und weiter verbessert.

Ebenfalls wurde untersucht, ob durch den Einsatz von „PM Game“ auch tatsächlich ein Wissensaufbau bei den Studierenden stattfand. Bei einer Befragung der am Live Test beteiligten Studenten gaben gut 60 % an, dass durch die Verwendung von „PM Game“ ein Lernvorteil erzielt werden konnte.

In einer Analyse wurden ausserdem Vorschläge ausgearbeitet, welche umgesetzt werden sollten, damit die Applikation auch für externe Trainingsdienstleitungen an Schulen und Unternehmen angeboten werden kann.

# Management Summary

## Ausgangslage

In dieser Bachelorarbeit wird die bestehende Applikation „PM Game“ weiterentwickelt. Der Betreuer der Arbeit hat hierzu eine Liste mit User Storys definiert, welche umgesetzt werden müssen. Bei „PM Game“ handelt es sich um eine webbasierte Applikation, mit der ein Softwareprojekt simuliert werden kann. Entwickelt wurde die erste Version dieser Applikation in einer Studienarbeit an der Ostschweizer Fachhochschule (OST) [6]. Sie wurde für den Einsatz in den Übungslektionen des Moduls Projekt- und Qualitätsmanagement (PmQm) konzipiert. Auch die Weiterentwicklung der Applikation ist auf die Verwendung in diesem Modul zugeschnitten.

Der Hauptteil dieser Arbeit befasst sich mit der Umsetzung der neuen funktionalen sowie nicht-funktionalen Anforderungen an die Applikation. Neben dieser praktischen Umsetzung wurde auch eine theoretische sowie betriebswirtschaftliche Fragestellung beantwortet. So wurde in dieser Arbeit untersucht, ob durch den Einsatz der Applikation im Modul PmQm die Studenten einen Lernvorteil erzielen konnten. Ausserdem wurde die fertige Software analysiert, um zu eruieren, was nötig wäre, damit die Software als Trainingsdienstleistung für Projektleiter angeboten werden könnte.

## Vorgehen und Technologie

In dem praktischen Teil dieser Arbeit haben wir die funktionalen und nicht-funktionalen Anforderungen umgesetzt. Wir haben uns entschieden, das Projekt nach dem Wasserfall Modell zu organisieren. Dieses Vorgehensmodell haben wir gewählt, da es uns für die kurze Projektdauer und den eng abgesteckten Funktionsumfang am effizientesten erschien.

Nach unserer Planung sollten die funktionalen sowie nicht-funktionalen Anforderungen der Software nach gut der Hälfte der Projektdauer umgesetzt sein. Anschliessend hatten wir eine dreiwöchige Testphase angesetzt, in der die Applikation durch einen Einsatz in den Übungslektionen ausgiebig getestet wurde. Die Bugs und Feedbacks, welche in diesen Tests gesammelt wurden, haben wir in dieser Phase direkt behoben.

Um die theoretische Fragestellung beantworten zu können, haben wir die Studenten, welche an den Live Tests der Applikation teilgenommen haben, befragt. Durch die Auswertung dieser Befragung haben wir die Frage beantwortet, ob die Studenten durch die Verwendung von „PM Game“ einen Lernvorteil erzielen konnten.

Zu guter Letzt haben wir die Software analysiert und mit bestehenden Lösungen auf dem Markt verglichen. Aus dieser Analyse haben wir Vorschläge erarbeitet, welche umgesetzt werden sollten, damit die Software als Trainingsdienstleistung für Externe angeboten werden könnte.

Für die Umsetzung der Software haben wir die bestehenden Technologien der Applika-

tion übernommen und entsprechend erweitert, damit diese den neuen funktionalen und nicht-funktionalen Anforderungen genügen. Für das Backend wurde wie bisher Node.js<sup>1</sup> mit TypeScript<sup>2</sup> verwendet. Da die Applikation neu in einer Server Client Architektur aufgebaut ist, wurde das Backend mit Express.js<sup>3</sup> so erweitert, dass API Aufrufe verarbeitet werden können. Die Datenbankschicht wurde auf einen neuen OR-Mapper migriert. Das Frontend der Applikation wurde komplett neu mit VUE.js<sup>4</sup> umgesetzt. Da der alte Code generell komplex und unsauber umgesetzt war, wurde die neue Version der Applikation von Grund auf neu geschrieben. Dadurch ist die neue Version der Applikation wesentlich wartungsfreundlicher und kann künftig einfacher erweitert werden.

## Ergebnisse und Ausblick

Als Ergebnis des praktischen Teils der Arbeit entstand eine voll funktionsfähige Software inklusive Dokumentation. Wir konnten bis auf eine Ausnahme alle funktionalen Anforderungen an die Software umsetzen. Zusätzlich wurden noch zahlreiche weitere Verbesserungen aufgrund der Feedbacks aus der Testphase umgesetzt. Die funktionale Anforderung, welche nicht umgesetzt werden konnten, hatte in der Planung des Projekts die niedrigste Priorität erhalten. Wir haben daher mittels Change Request beantragt, diese Anforderung zugunsten anderer Verbesserungsmöglichkeiten, welche wir in der Testphase identifizieren konnten, zurückzustellen. Dieser Change Request wurde vom Betreuer der Arbeit genehmigt.

Bei der Befragung der Studierenden nach den Live Tests gaben 60 % die Einschätzung ab, dass Sie mit der Verwendung von „PM Game“ einen Lernvorteil erzielen konnten. Durch die Auswertung der Umfrage stellte sich ebenfalls heraus, dass die meisten Studierenden eine Applikation wie „PM Game“ als Ergänzung zu den bestehenden Übungen sehen und bei der Verwendung sehr motiviert sind.

Durch den Vergleich der Applikation „PM Game“ mit bereits am Markt etablierten Applikationen konnten wir eine Liste mit Vorschlägen erarbeiten, welche umgesetzt werden sollten, damit die Applikation als Trainingsdienstleistung für externe Projektleiter angeboten werden könnte.

---

<sup>1</sup><https://nodejs.org/en/>

<sup>2</sup><https://www.typescriptlang.org>

<sup>3</sup><https://expressjs.com/de/>

<sup>4</sup><https://vuejs.org>

# Inhaltsverzeichnis

<b>Änderungsverlauf</b>	<b>ii</b>
<b>Abstract</b>	<b>iii</b>
<b>Management Summary</b>	<b>iv</b>
<b>1. Einleitung</b>	<b>5</b>
1.1. Ziele . . . . .	5
1.2. Technologien, Vorgehen . . . . .	5
1.3. Aufbau . . . . .	6
<b>2. Praktische Fragestellung</b>	<b>7</b>
2.1. Ausgangslage & Aufgabenstellung . . . . .	7
2.2. Lösungskonzept & Umsetzung . . . . .	8
2.3. Unterschiede zur Vorgängerversion . . . . .	9
2.4. Ergebnis & Schlussfolgerungen . . . . .	11
<b>3. Theoretische Fragestellung</b>	<b>13</b>
3.1. Ausgangslage & Aufgabenstellung . . . . .	13
3.2. Methodik . . . . .	13
3.3. Auswertung Fragebogen . . . . .	14
3.4. Ergebnis & Schlussfolgerungen . . . . .	25
<b>4. Betriebswirtschaftliche Fragestellung</b>	<b>26</b>
4.1. Ausgangslage & Aufgabenstellung . . . . .	26
4.2. Analyse . . . . .	26
4.2.1. Mandantenfähigkeit . . . . .	26
4.2.2. Funktionsumfang . . . . .	27
4.2.3. Mehrsprachigkeit . . . . .	29
4.2.4. Performance . . . . .	30
4.2.5. Sicherheit . . . . .	30
4.2.6. Demo Modus . . . . .	31
4.2.7. Dokumentation . . . . .	32
4.2.8. Spiel Verwaltung . . . . .	32
4.3. Auswertung Fragebogen . . . . .	33
4.4. SWOT Analyse „PM Game“ . . . . .	37
4.5. Ergebnis & Schlussfolgerungen . . . . .	38
<b>5. Software Engineering Dokumentation</b>	<b>40</b>
5.1. Funktionale Anforderungen . . . . .	40
5.1.1. User Storys . . . . .	40

5.1.2. Use Cases . . . . .	41
5.1.2.1. Use Cases ausformuliert . . . . .	44
5.2. Erweiterungsmöglichkeiten . . . . .	51
5.3. Domain Model . . . . .	51
5.4. Nicht-funktionale Anforderungen . . . . .	52
5.4.1. Zu unterstützende Szenarien . . . . .	53
5.4.2. Nicht-funktionale Anforderungen . . . . .	54
5.5. Logische Architektur . . . . .	57
5.6. Deployment-Diagramm . . . . .	58
5.7. GUI-Skizzen . . . . .	59
5.8. Architekturentscheidungen: Diskussion . . . . .	63
5.9. Testing . . . . .	66
5.9.1. Testing der funktionalen Anforderungen . . . . .	66
5.9.1.1. Black-Box-Testing . . . . .	66
5.9.1.2. White-Box-Testing . . . . .	66
5.9.2. Testing der nicht-funktionalen Anforderungen . . . . .	69
5.9.3. Live Tests . . . . .	69
5.9.4. Load Tests . . . . .	71
5.9.5. Erkenntnisse . . . . .	74
5.10. Installationsanleitung . . . . .	76
5.10.1. Voraussetzungen . . . . .	76
5.10.2. Deployment auf OST Server . . . . .	76
5.10.3. Unterteilung Staging & Production . . . . .	78
5.10.4. Konfiguration der Komponenten . . . . .	78
5.10.5. Installation auf einem neuen Host . . . . .	82
5.10.6. Lokales Ausführen der Applikation . . . . .	82
5.10.7. Datensicherung Datenbank . . . . .	83
5.10.7.1. Externe Lagerung . . . . .	84
5.10.7.2. Manuelle Sicherung . . . . .	84
5.10.7.3. Datenbank Wiederherstellung . . . . .	84
5.11. Einrichtung der Entwicklungsumgebung . . . . .	85
5.11.1. Verwendete Tools . . . . .	85
5.12. Benutzerhandbuch (User Manual) . . . . .	86
5.12.1. Benutzer Verwaltung (Admin) . . . . .	86
5.12.2. Anlegen eines neuen Spiels (Admin) . . . . .	86
5.12.3. Spiel starten (Admin) . . . . .	86
5.12.4. Spielereingabe (Spieler) . . . . .	87
5.12.5. Change Request bearbeiten (Admin) . . . . .	88
5.12.6. Runde beenden ankündigen (Admin) . . . . .	88
5.12.7. Runde beenden (Admin) . . . . .	89
5.13. Offene Issues . . . . .	89
5.13.1. Zurückgestellt . . . . .	89
5.13.2. Known Bugs . . . . .	89

5.13.3. Nice to have . . . . .	90
5.14. Screenshots . . . . .	91
5.15. Software-Metriken und Code Analyse . . . . .	99
5.15.1. SLOC . . . . .	99
5.15.2. Komplexität . . . . .	99
5.16. Qualität . . . . .	101
<b>6. Projektdokumentation</b>	<b>104</b>
6.1. Projektausschreibung . . . . .	104
6.2. Aufgabenstellung . . . . .	105
6.3. Zeitplan . . . . .	107
6.4. Meilensteintrendanalyse . . . . .	111
6.5. Projektplan . . . . .	111
6.5.1. Kritischer Rückblick: Vergleich Ist vs. Soll . . . . .	111
6.6. Risikoliste . . . . .	113
6.6.1. Eingetretene Risiken . . . . .	115
6.7. Zeitauswertung . . . . .	115
<b>7. Projektschlussberichte</b>	<b>118</b>
7.1. Martin Iten . . . . .	118
7.2. Dario Caluzi . . . . .	121
<b>Glossar</b>	<b>124</b>
<b>Literatur</b>	<b>126</b>
<b>Abbildungsverzeichnis</b>	<b>127</b>
<b>Tabellenverzeichnis</b>	<b>129</b>
<b>A. Projektplan</b>	<b>130</b>

# 1. Einleitung

Diese Arbeit wurde als Folgearbeit fortgeführt. Im Herbstsemester 2019 wurde für das Unterrichtsmodul PmQm die erste Version eines Projektsimulators umgesetzt. Diese Software dient dazu, um den im Unterricht erlernten Stoff auf praktische Art und Weise in den Übungslektionen anzuwenden.

Die Software ist dabei nicht für den eigenständigen Gebrauch durch die Studierenden ausgelegt. Die Verwendung erfolgt primär in der Klasse während den Übungslektionen des Moduls PmQm. Die Studierenden übernehmen als Gruppe die Aufgabe des Projektleiters in einem vom Dozenten gesteuerten Softwareprojekt. Durch einen rundenbasierten Ansatz kann der Dozent den Verlauf des Projekts nach jeder Runde mit den Studenten besprechen.

## 1.1. Ziele

Die Ziele dieser Arbeit leiten sich aus der theoretischen, der praktischen und der betriebswirtschaftlichen Fragestellung ab:

1. Können die zusätzlichen funktionalen und nicht-funktionalen Anforderungen des Dozenten in einer Software umgesetzt werden? Begründung, wenn die Anforderungen nicht wie gefordert umgesetzt werden konnten
2. Verschafft der Einsatz einer Software in den Übungsstunden einen tatsächlichen Wissensaufbau, da sich die Studierenden damit auseinandersetzen müssen?
3. Was wäre notwendig, um diese Software aus Sicht Ostschweizer Fachhochschule als Trainingsdienstleitung für Externe anzubieten?

## 1.2. Technologien, Vorgehen

Die Vorgängerversion der Software basierte auf Node.js und wurde serverseitig gerendert. Der Code war zudem schwer wartbar und erforderte daher generell eine Überarbeitung in grösserem Umfang, so dass die Software auch künftig sauber weiterentwickelt werden kann.

Wir haben uns aus diesen Gründen dazu entschieden, die Software von Grund auf neu zu schreiben. Das Frontend und Backend haben wir voneinander getrennt. Die gesamte Logik des Spiels befindet sich im Backendserver, welcher eine API zur Verfügung stellt. Dieser API bedient sich der Webclient, welcher auf VUE.js basiert. Mit der Verwendung von Socket.io werden zudem relevante Änderungen im Backend in Echtzeit auf die verbundenen Webclients gepusht. Dadurch können viele Komponenten im Frontend in Echtzeit zwischen den Gruppenmitgliedern synchronisiert werden.

Aufgrund des klar eingegrenzten Zeithorizonts sowie den vorgegebenen funktionalen und nicht-funktionalen Anforderungen an die Software, haben wir uns entschieden das Was-

serfall Modell zu verwenden. Dadurch konnte nach der Planungsphase die Umsetzung in einem Block vorgenommen werden. Durch die kleine Teamgrösse konnten wir zudem auch ohne regelmässige Meetings während der Umsetzungsphase die Arbeiten sehr gut koordinieren. Weiter war mit Martin Iten ein Mitglied des Teams bereits an der Umsetzung der Vorgängerarbeit beteiligt. Dadurch konnten auch die Bedürfnisse des Dozenten an die Software besser abgeschätzt werden. Dies hat die Anzahl an Rückfragen und Meetings ebenfalls reduziert,

### **1.3. Aufbau**

In dieser Dokumentation beantworten wir die im Unterabschnitt 1.1 definierten Fragestellungen. Im ersten Abschnitt wird die praktische Umsetzung behandelt. Anschliessend wird auf die theoretische und betriebswirtschaftliche Fragestellung eingegangen.

- Die praktische Fragestellung wird als Software Engineering Dokumentation in dieser Arbeit beantwortet.
- Die theoretische Fragestellung wird anhand wissenschaftlicher Literatur sowie Umfragen bei Studenten beantwortet.
- Die betriebswirtschaftliche Fragestellung wird mit Analysen der Software beantwortet.

## 2. Praktische Fragestellung

In diesem Kapitel behandeln wir die Resultate der praktischen Umsetzung der Software. Für die Details zum Verlauf des Projekts und zur Umsetzung der Software sei auf das Kapitel 6 „Projektdokumentation“ sowie auf das Kapitel 5 „Software Engineering Dokumentation“ verwiesen.

### 2.1. Ausgangslage & Aufgabenstellung

Wir wurden im Rahmen dieser Arbeit beauftragt, die bestehende Software „PM Game“ zu erweitern. Bei „PM Game“ handelt es sich um eine interaktive Simulation eines Projekts. Die Spieler haben die Möglichkeit, die im Unterricht des Moduls PmQm erlernten Inhalte in einem simulierten Projekt anzuwenden. Mit verschiedenen Stellschrauben wird hierbei der Projektverlauf gesteuert. Die erste Version von „PM Game“ umfasst folgende Stellschrauben:

**Mitarbeiter Management** Die Studenten weisen dem Projekt Mitarbeiter Ressourcen zu. Diese Mitarbeiter Ressourcen setzen pro Runde Function Points um.

**Projektleiter Management** Die Studenten definieren für welche Aufgaben der Projektleiter seine Zeit aufwendet. Aufgrund der getätigten Ausgaben kann der Spielleiter entscheiden, ob ein späterer Change Request gewährt wird.

**Investitionen** Die Studenten investieren Geld in die Ausbildung der Mitarbeiter und in die Risikoreduktion. Durch reduziertes Risiko verliert eine Gruppe weniger Function Points wenn ein Event eintrifft.

**Change Request** Die Studenten können in einer Runde einen Change Request schreiben. Der Spielleiter erhält die Change Requests aller Gruppen und kann entscheiden ob diese angenommen oder abgelehnt werden.

Die Studenten sind in Gruppen organisiert und tätigen die Eingaben über mehreren Runden. Der Spielleiter setzt während des Spiels die Parameter für den Verlauf des Projekts und erhält Statistiken über den Spielverlauf sowie eine detaillierte Einsicht in die Eingaben der Spieler.

Die Vorgängerversion der Software wird über den Webbrowser aufgerufen. Diese basiert auf Node.js und rendert die Seiten serverseitig mit Handlebars.js <sup>5</sup>.

Durch den Projektbetreuer wurden für die neue Version von „PM Game“ Anforderungen gestellt, welche den Funktionsumfang der Applikation erweitern oder bestehende Funktionen verbessern. Der Fokus liegt hierbei auf einem umfangreicheren Risikomanagement sowie einem Messaging System und E-Mail Benachrichtigungen.

---

<sup>5</sup><https://handlebarsjs.com>

## 2.2. Lösungskonzept & Umsetzung

Um die Anforderungen des Projektbetreuers umzusetzen, haben wir ein Softwareprojekt geplant und durchgeführt. Wir haben uns dabei für das Vorgehensmodell „Wasserfall“ entschieden. Dafür wurde der Zeitplan erarbeitet mit den Phasen Anforderungsanalyse, Systemdesign, Programmierung, Testing und Auslieferungen. Jede Phase wurde mit einem Meilenstein am Ende versehen:

- Anforderungsanalyse durchgeführt - 01.10.2020
- Systemdesign erstellt - 06.10.2020
- Programmierung umgesetzt - 11.11.2020
- Testing abgeschlossen - 29.11.2020
- Arbeit abgegeben - 17.12.2020

In der Phase Anforderungsanalyse haben wir die vorliegenden Anforderungen analysiert, die Projektplanung erstellt und mögliche Risiken definiert. Anschliessend konnte in der Phase Systemdesign das Klassendiagramm an die neuen Anforderungen angepasst werden. Ebenfalls wurden in dieser Phase aufgrund der neu gewählten Architektur das Architektur- und Schichtendiagramm komplett überarbeitet. In der Phase Programmierung wurde dann die alte Version des „PM Game“ auf die neue Architektur portiert und die neuen Anforderungen umgesetzt. Die Applikation wurde dann in der Phase Testing durch drei Live Tests eingehend geprüft und gefundene Fehler behoben. In der letzten Phase Auslieferung wurde die Projektdokumentation erstellt und der Code für die Auslieferung an den Projektbetreuer vorbereitet.

Wie auch die Vorgängerversion haben wir die neue Version von „PM Game“ als Webapplikation umgesetzt. Im Gegensatz zur alten Version haben wir uns jedoch für eine Server Client Architektur entschieden. Wir erwarteten hiervon eine bessere Wartbarkeit, besseres Testing sowie grössere Flexibilität bei der Umsetzung neuer und bestehender Funktionen.

Für das Backend setzen wir auf Node.js mit Express.js für die Bereitstellung der API-Endpunkte sowie auf den OR-Mapper „TypeORM“ für die Persistierung der Daten in einer SQL Datenbank. Durch TypeORM konnten wir während der Entwicklung eine einfache SQLite Datenbank und auf der produktiven Umgebung eine MariaDB Datenbank verwenden.

Das Frontend besteht aus einer VUE.js Single Page Application. Sobald die Applikation im Webbrowser geladen ist, werden sämtliche Daten über die API vom Backend abgerufen. Um eine bessere Interaktivität zu ermöglichen, pusht der Backendserver zudem gewisse Meldungen über Socket.io an das Frontend. So ist es z. B. möglich eine Benutzereingabe eines Gruppenmitglied umgehend bei allen anderen Teilnehmern der Gruppe anzuzeigen.

### 2.3. Unterschiede zur Vorgängerversion

Da die Arbeit auf der Vorgängerversion der Software aufbaut, möchten wir aufzeigen was sich zu dieser Version verändert hat. Die nachfolgende Tabelle vergleicht die Hauptfunktionen der Applikation zwischen der Version 1.0 und der Version 2.0.

	Version 1.0 (Studienarbeit)	Version 2.0 (Bachelorarbeit)
Architektur	Node.js Applikation mit serverseitigem Rendering. Sqlite Datenbank zum speichern der Daten.	Node.js Client-Server Applikation. VUE Singlepage Applikation als Frontend. MariaDB Datenbank zum speichern der Daten.
Performance	Getestet für 18 Teilnehmer gleichzeitig.	Getestet für 180 Benutzer gleichzeitig.
Messaging System	Nicht vorhanden.	Kommunikation zwischen Spielleiter und Spieler sowie innerhalb der eigenen Gruppe möglich.
E-Mail Benachrichtigungen	Nicht vorhanden.	E-Mail Funktionalität für Benutzer Nachrichten sowie Passwort Resets.
Dashboard Spieler	Nicht vorhanden.	Die Spieler sehen einen Burndown Chart für das Budget sowie den Scope des Projekt. Ebenfalls sind weitere Kennzahlen ersichtlich.
Dashboard Admin	Enthält Burndown Chart für das Budget und den Scope aller Gruppen	Enthält ein Burndown Chart für das Budget und den Scope sowie eine Grafik für die Risiko Investitionen und Mitarbeiterausbildung, sowie Stand der Eingaben der aktuellen Runde, aller Gruppen.
Rangliste Admin	Nicht vorhanden.	Der Spielleiter sieht eine Rangliste mit den Gruppen eines Spiels. Die Sortierung der Rangliste ist vom gewählten Ziel abhängig.
Mitarbeiter Management	Die Spieler können Mitarbeiter Ressourcen zuweisen.	Die Spieler können einzelne Mitarbeiter dem Projekt zuweisen. Die Mitarbeiter haben unterschiedliche, zufallsgenerierte Werte
Projektleiter Management	Die Spieler teilen die Projektleiterzeit ein.	Der Spieler teilen die Projektleiterzeit ein.

Risiko Management	Die Spieler investieren ein allgemeines Budget pro Runde. Dieses senkt generell das Risiko.	Die Spieler investieren in spezifische Risiken oder ignorieren diese. Tritt ein Risiko ein, so wird nur das Restrisiko von diesem spezifischen Risiko als Schaden gezählt.
Change Request	Die Spieler können einen Change Request schreiben. Der Spielleiter kann diese annehmen, nochmals zurückgeben oder ablehnen.	Die Spieler können einen Change Request schreiben. Der Spielleiter kann diese annehmen, nochmals zurückgeben oder ablehnen.
Projekt Report	Die Spieler sehen Ihre Eingaben der aktuellen sowie der vergangenen Runden.	Die Spieler sehen Ihre Eingaben der aktuellen sowie der vergangenen Runden.
Gruppenübersicht	Nicht vorhanden.	Die Spieler sehen wo die anderen Gruppen stehen.
GUI	Sehr schlichtes und statisches GUI.	Komplett überarbeitetes GUI, Eingaben der Spieler werden innerhalb der Gruppe in Echtzeit synchronisiert. Die Diagramme im Adminbereich aktualisieren sich automatisch bei Eingaben. Mobile freundlich.
Risiko auslösen	Der Spielleiter erstellt ein Event und trägt für jede Gruppe den gewünschten Schaden ein.	Der Spielleiter wählt ein Risiko aus, welches ausgelöst werden soll. Der Schaden wird automatisch anhand der Eingaben der Spieler berechnet. Der Spielleiter kann auch einen Zufallsgenerator verwenden.
Authentisierung	Session Cookie.	JWT Token für den API Zugriff
Benutzer registrieren	Die Spieler registrieren sich mit einem Benutzernamen und einem Passwort.	Die Spieler registrieren sich mit einem Benutzernamen, einer E-Mail Adresse und einem Passwort.
Benutzerübersicht Admin	Nicht vorhanden.	Der Spielleiter sieht eine Übersicht mit allen Spielern sowie deren Gruppenzugehörigkeit. Die Liste kann durchsucht werden.

Spieler registrieren Admin	Nicht vorhanden.	Der Spielleiter kann im Adminbereich direkt neue Spieler hinzufügen. Die erstellten Spieler erhalten eine E-Mail Einladung.
Passwort Reset	Nicht vorhanden.	Die Spieler können selbständig ein Passwort Reset auslösen. Es wird eine E-Mail mit einem Link zum setzen des neuen Passworts zugestellt.
Admin Export Diagramme	Nicht vorhanden.	Der Spielleiter kann die Diagramme als PNG Datei exportieren. Alternativ steht ein CSV Export der Daten zur Verfügung.

Tabelle 2: Vergleich zwischen „PM Game“ Version 1.0 und 2.0

## 2.4. Ergebnis & Schlussfolgerungen

Die funktionalen Anforderungen an die Software konnten weitgehend umgesetzt werden. In der Testphase wurden einige Verbesserungsvorschläge und Fehler aufgenommen, welche für einen reibungslosen Spielbetrieb ebenfalls als wichtig eingestuft wurden. Die betreffenden Issues wurden erfasst und ebenfalls umgesetzt. Aus diesem Grund wurde die Anforderung „Spiel per Token beitreten“ zurückgestellt und schlussendlich mittels Change Request beantragt, das Issue zugunsten der anderen Funktionen nicht umzusetzen. Von den erhobenen 14 User Storys wurden somit 13 umgesetzt. Alle User Storys mit der Priorität „MUSS“ und „SOLL“ wurden erledigt.

Die nicht-funktionalen Anforderungen wurden ebenfalls weitgehend erfüllt. Die Trennung des Frontend und Backend war von allen nicht-funktionalen Anforderungen die Aufwändigste, konnte jedoch vollständig umgesetzt werden. Die Performance der Applikation liegt ebenfalls im Rahmen der definierten Anforderungen. Im Vergleich zur Vorgängerversion wurde das GUI zudem wesentlich verbessert und validiert auch mehr Benutzereingaben. Eine genaue Analyse der nicht-funktionalen Anforderungen findet sich im Kapitel 5.4.2 „Nicht-funktionale Anforderungen“

Folgende Punkte würden wir im Nachhinein anders handhaben:

- Bereits vor den Live Tests einen zusätzlichen kleineren Test mit einer reduzierten Testgruppe ansetzen. Dadurch hätten einige Fehler früher erkannt werden können und der Spielfluss im ersten Live Test wäre reibungsloser abgelaufen.
- Strikt den Datentyp Numbers von Beginn an casten. Unterschiedliche Browser lie-

fern teilweise für ein Number Wert unterschiedliche Datentypen. Dies hat Probleme verursacht und musste im Nachhinein mühsam nachgebessert werden.

- Das Testen für das Frontend strikter handhaben. Für die manuellen Tests mit umfangreichen Checklisten arbeiten.
- Der Testing Phase mehr Zeit einräumen. Diese Phase hat länger gedauert als geplant, weil noch diverse Verbesserungsvorschläge umgesetzt und Fehler behoben wurden.

Folgende Punkte würden wir wieder gleich machen:

- Das Wasserfallmodell hat sich für die kurze Projektlaufzeit mit den fix abgesteckten User Storys bewährt.
- Die CI/CD Pipeline für das komplette Deployment verwenden. Das Deployment auf die produktive Umgebung mittels Releases (Tags) hat sehr gut geklappt.
- Eine Staging Umgebung für das Testen bereitstellen. Dadurch konnte vor einem Release ein Softwarestand nochmals eingehend geprüft werden.
- TypeScript anstatt reines JavaScript verwenden. Der Code mit den Typen ist verständlicher und fehlerresistenter.
- Das Frontend und Backend trennen. Die Applikation wurde hierdurch wesentlich wartungsfreundlicher.
- Live Tests durchführen um Fehler in der Applikation zu finden.

## 3. Theoretische Fragestellung

### 3.1. Ausgangslage & Aufgabenstellung

In diesem Abschnitt versuchen wir die Frage zu beantworten, ob die Studenten durch den Einsatz der Applikation „PM Game“ einen Wissensaufbau erzielen konnten. Leider lässt sich diese Frage nicht direkt beantworten. Dies liegt einerseits daran, dass wir keine Kontrollgruppe hatten, welche „PM Game“ nicht verwendete und andererseits die Dauer dieser Arbeit die Prüfungsphase nicht abdeckt.

Daher haben wir uns entschieden, den Studenten nach Abschluss der Übungen mit „PM Game“ einen Fragebogen zur Selbsteinschätzung abgeben. Die Daten welche wir daraus erhielten, konnten wir zusammen mit wissenschaftlicher Literatur entsprechend interpretieren und somit eine Einschätzung ableiten, ob der Einsatz von „PM Game“ zu einem positiven Lerneffekt führte.

### 3.2. Methodik

Der Fragebogen umfasst 16 Fragen und ist so aufgebaut, dass die Studenten in einem ersten Schritt das Lernen mit einer Lernapplikation (in diesem Fall „PM Game“) anderen Lernmethoden gegenüberstellen und verschiedene Punkte gewichten müssen.

Anschliessend musste beantwortet werden, welche Punkte an einer Lernsoftware für die Studenten wichtig sind und wie zufrieden diese mit dem Einsatz von „PM Game“ waren.

Im nächsten Teil wollten wir von den Studenten wissen, ob Sie durch den Einsatz von „PM Game“ einen persönlichen Lerneffekt erzielen konnten. Um weitere Details zu dieser Einschätzung zu erhalten, haben wir die Studenten aufgefordert diese Einschätzung in schriftlicher Form zu begründen.

Leider mussten wir nachträglich feststellen, dass uns bei der Konzeptionierung des Fragebogens ein Fehler unterlaufen ist. Die Datenerhebung erfolgte grundsätzlich anonym. Daher ist es uns leider nicht möglich, die erhaltenen Daten auf die Live Tests zu referenzieren. Hierzu hätten wir mindestens den Gruppennamen in der Umfrage einschliessen sollen.

Auch wenn uns durch diesen Umstand einige spannende Auswertungen verwehrt bleiben, so beeinträchtigt es die Beantwortung der theoretischen Fragestellung nicht. Wir gehen davon aus, dass ein gutes Ranking in der Simulation nicht mit einem Lernerfolg direkt zusammenhängt, denn die Leistung beim „PM Game“ wird einerseits durch mehrere Teilnehmer einer Gruppe erzielt und ist teilweise auch abhängig vom Zufall. Die Aussagekraft des Rankings ist daher eher als gering einzuschätzen.

### 3.3. Auswertung Fragebogen

Insgesamt haben 48 von 56 Studenten den Fragebogen ausgefüllt. Die durchschnittliche Dauer für die Beantwortung des gesamten Fragebogens lag bei 08:37 Minuten.

Nachfolgend werden die einzelnen Fragen sowie deren Resultate analysiert und wo möglich einen Rückschluss auf den Lerneffekt gezogen. Für den theoretischen Teil dieser Arbeit werden in diesem Kapitel die Fragen 1 - 11 analysiert. Die Fragen 12 - 16 haben den Schwerpunkt auf der betriebswirtschaftlichen Fragestellung und werden daher im Abschnitt 4 behandelt.

#### Frage 1 - Sie müssen Unterrichts-Stoff ohne jegliches Übungsmaterial lernen. Bitte bewerten Sie hierzu folgende Aussagen:

Um die Einschätzungen der Studenten zum Lernen mit einer Applikation wie „PM Game“ besser einteilen zu können, wollten wir zuerst Vergleichswerte zu alternativen Lernmethoden erhalten. Daher sollten die Studenten einschätzen, wie Sie Unterrichtsstoff lernen, wenn durch den Dozenten kein Übungsmaterial zur Verfügung gestellt wird.

Unsere Annahme war, dass die Studenten das Lernen ohne zusätzliches Lernmaterial am ineffektivsten einschätzen.

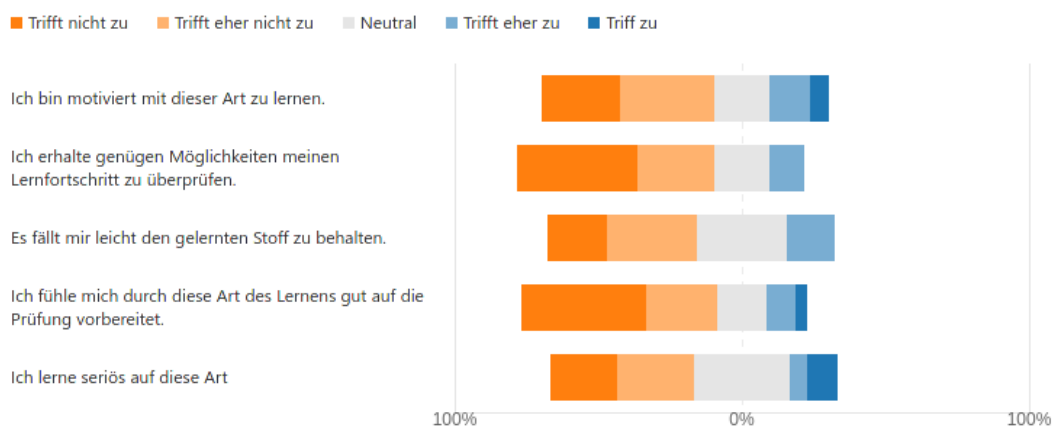


Abbildung 1: Auswertung Frage 1

Diese Annahme hat sich in der Umfrage bestätigt. Die Studenten gaben fast durchwegs an, dass Sie mit dieser Lernmethode weder motiviert, noch gut auf eine etwaige Prüfung vorbereitet wären.

Bei der Frage 3 ziehen wir Vergleiche zu den erhaltenen Resultaten dieser Frage.

**Frage 2 - Sie müssen Unterrichtsstoff lernen und erhalten dazu Übungsaufgaben auf Papier. Bitte bewerten Sie hierzu folgende Aussagen:**

Neben dem Lernen ohne jegliches Übungsmaterial, wollten wir auch wissen wie die Studenten mit klassischem Übungsmaterial (dazu zählen unter anderem Textaufgaben, Rechercheaufträge, Multiple Choice und Quiz) lernen. An der OST sind entsprechende Übungen auf Papier weit verbreitet. Wir gehen daher davon aus, dass der Stoff auf diese Art zu vertiefen bei den Studenten entsprechend akzeptiert ist.

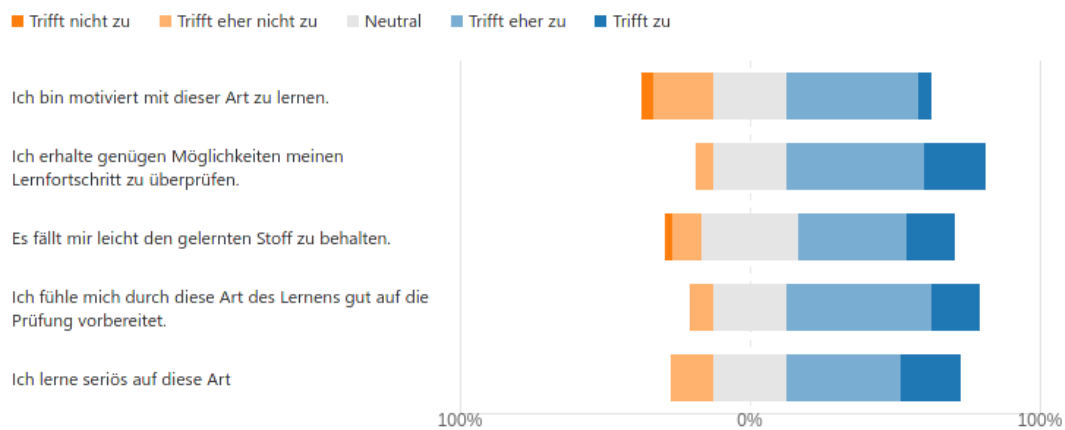


Abbildung 2: Auswertung Frage 2

Die Umfrage zeigt, dass die Akzeptanz und das Vertrauen in diese Übungsart gross sind. Die Studenten sind eher motiviert so zu lernen und fühlen sich gut auf Prüfungen vorbereitet.

Bei der Frage 3 ziehen wir Vergleiche zu den erhaltenen Resultaten dieser Frage.

**Frage 3 - Sie müssen Unterrichtsstoff lernen und haben die Möglichkeit dazu ein passendes Computerspiel wie „PM Game“ zu verwenden. Bitte bewerten Sie hierzu folgende Aussagen:**

Bei dieser Frage wollten wir nun wissen, wie die Studierenden das Lernen mit einer Applikation wie „PM Game“ einschätzen. Dazu haben wir die Studenten gebeten, dieselben Punkte wie bei den vorhergehenden Fragen zu beurteilen. Wir vergleichen diese Punkte anschliessend mit den Resultaten aus den vorhergehenden Fragen.

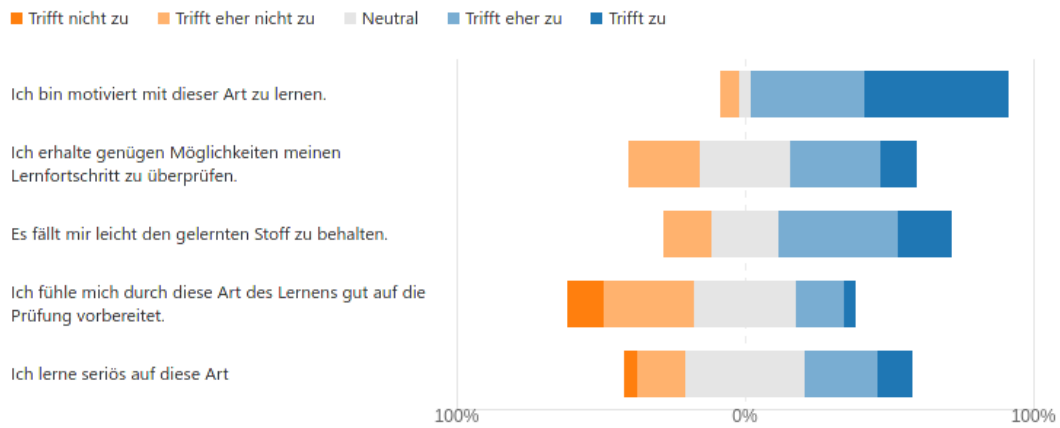


Abbildung 3: Auswertung Frage 3

Nachfolgend möchten wir die einzelnen Punkte dieser Frage genauer untersuchen und mit den Resultaten der Frage 1 und 2 vergleichen. Wir haben dazu den Anteil von „Trifft eher zu“ und „Trifft zu“ der verschiedenen Lernmethoden verglichen.

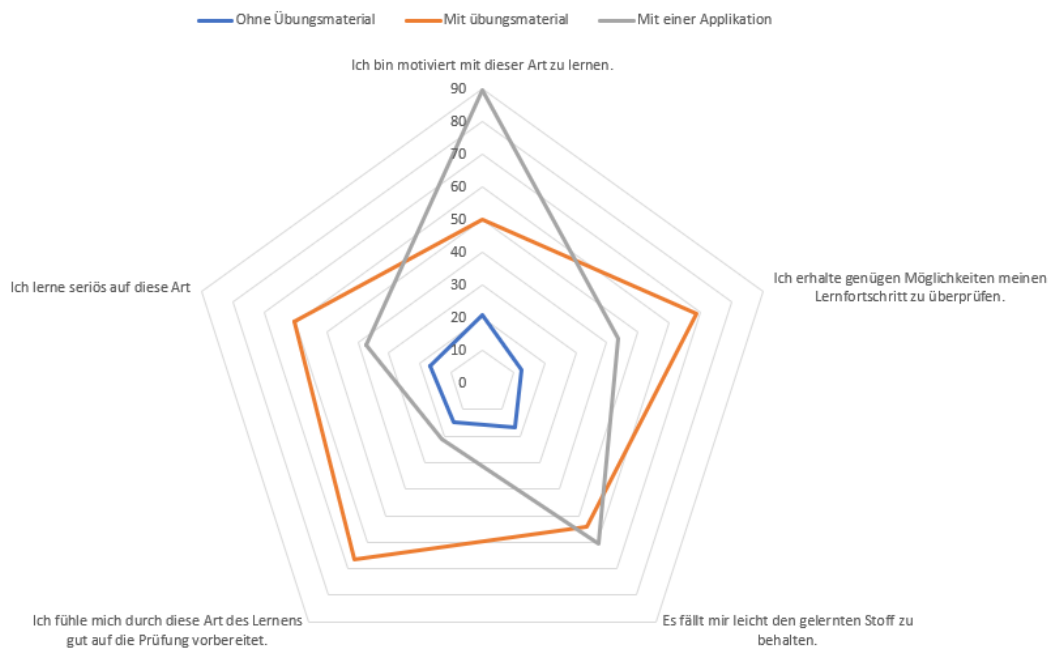


Abbildung 4: Vergleich der drei Lernmethoden anhand des Anteils an „Trifft eher zu“ und „Trifft zu“

**Ich bin motiviert mit dieser Art zu lernen.** Die Mehrheit der Studenten ist mit einer Lernapplikation wie „PM Game“ äusserst motiviert, Stoff zu lernen und zu vertiefen. Die Motivation mit einer interaktiven Applikation zu lernen ist signifikant grösser, als

bei den alternativen Optionen. Dies erachten wir als einen wichtigen Punkt bezüglich des Lerneffekts, da nach aktuellem Forschungsstand zumindest ein mittelstarker Zusammenhang zwischen Motivation und Leistung beim Lernen besteht. [1]

**Ich erhalte genügend Möglichkeiten meinen Lernfortschritt zu überprüfen.** hier hat das Lernen mit einer Lernapplikation wie „PM Game“ solide abgeschnitten. Knapp die Hälfte der Studierenden gaben an, dass Sie ihren Lernfortschritt gut überprüfen können mit der Applikation. Bei dem Lernen mit klassischem Übungsmaterial ist der Anteil dieser Studenten noch etwas höher. Beim Lernen ohne Übungsmaterial gaben die meisten Studenten an, dass diese Aussage nicht zutreffe.

Einen sichtbaren Lernvorschritt zu haben, ist einer der bedeutendsten Einflussfaktoren für den Lernerfolg. [2]. Knapp der Hälfte der Studenten bot „PM Game“ ausreichende Möglichkeiten den Lernfortschritt zu überprüfen.

Es besteht daher durchaus noch Verbesserungspotenzial, damit die Lernapplikation den Lernfortschritt besser darstellen kann und Möglichkeiten bietet, diesen zu überprüfen.

**Es fällt mir leicht den gelernten Stoff zu behalten.** Hier hat das Lernen mit einer Lernapplikation wie „PM Game“ im Vergleich aller Lernmethoden am besten abgeschnitten. Das Lernen mit „PM Game“ fällt in die Kategorie „fallbasiertes Lernen“. Beim fallbasierten Lernen wird eine Anwendungssituation geboten, in der das erworbene Wissen praktisch angewendet wird. Fallbasiertes Lernen gilt als förderlich für den Wissenstransfer [3].

Wir gehen daher davon aus, dass die Studenten durch die praktische Anwendung des Stoffs im „PM Game“, diesen besser verinnerlichen konnten und somit einen Lernvorteil erzielen konnten.

**Ich fühle mich durch diese Art des Lernens gut auf die Prüfung vorbereitet.** Die Werte zum Lernen ohne Übungsmaterial sowie mit klassischem Übungsmaterial liegen im erwarteten Rahmen. Beim Lernen mit dem „PM Game“ hingegen hatten wir eine höhere Akzeptanz in Hinblick auf die Prüfungsvorbereitung erwartet. Knapp 20 % der Studenten fühlten sich durch die Verwendung von „PM Game“ gut auf die Prüfung vorbereitet. Der überwiegende Anteil (43.8 %) fühlte sich jedoch nicht gut auf die Prüfung vorbereitet und 35.4 % der Studenten hatten eine neutrale Meinung (weder noch).

Aus den Resultaten der anderen Fragen sowie dem schriftlichen Feedback geht allerdings hervor, dass ein Wissenstransfer stattgefunden hat und gut 60 % der Studenten einen Lernvorteil durch die Anwendung erzielen konnten. Wir haben daher nach Erklärungen für das Resultat dieser Aussage gesucht.

- Die Applikation „PM Game“ ist nur für einen Teil der Lernziele des PmQm ausgelegt. Möglicherweise wurde die Frage fälschlicherweise so verstanden, dass die

gesamte Prüfung damit gelernt werden soll.

- Möglicherweise fällt es den Studenten schwer, dass durch praktische Anwendung vertiefte Wissen auf eine theoretische Aufgabe anzuwenden.
- Prüfungsaufgaben an der OST basieren häufig auf schriftlichen Aufgaben, welche aus den Übungslektionen bekannt sind. Es fällt den Studenten möglicherweise schwer, sich eine Aufgabe vorzustellen, welche auf der Anwendung von „PM Game“ basiert.
- Möglicherweise zeigt die Applikation „PM Game“ zu wenig auf, welche Entscheidungen in der Simulation mit welchen theoretischen Grundlagen aus dem Unterricht verbunden sind.

In den schriftlichen Feedbacks gaben einige Studierende an, dass durch die Simulation mit „PM Game“ ein besseres Verständnis für reale Projekte erzielt werden konnte:

Ich habe im Spiel gesehen, wie Projektmanagement etwa in einem echten Projekt aussehen könnte und das hat für das Verständnis sehr geholfen.

Ich konnte zum ersten Mal erleben, wie sich Projekt-Management-Entscheidungen auf den Verlauf eines Projekts auswirken. Ich denke, mir bleibt dieser Verlauf durch das „selber erleben“ präsent, wodurch ich bei einem nächsten Projekt/Spiel besser weiss, was es alles zu beachten und zu entscheiden gibt.

Wir gehen daher davon aus, dass zumindest ein Teil der Studierenden den Lernvorteil zwar nicht für die Anwendung an der Prüfung, jedoch für die spätere Anwendung in einem Projekt sehen.

**Ich lerne seriös auf diese Art.** Dieses Resultat hatten wir erwartet. Durch den spielerischen Aspekt dürfte die Verwendung von „PM Game“ etwas weniger seriös eingestuft werden als das Lernen mit klassischem Übungsmaterial.

#### **Frage 4 - Sollten spielbasierte Übungen die klassischen Übungen Ihrer Meinung nach eher ersetzen oder ergänzen?**

Unsere Annahme bei dieser Frage war, dass die meisten Studierenden eine interaktive Lernapplikation wie das „PM Game“ als Ergänzung zu bestehenden Lernmethoden sehen.



Abbildung 5: Auswertung Frage 4

Diese Annahme wurde bestätigt. Nur 8.4 % würden klassische Übungen komplett mit einer Applikation wie „PM Game“ ersetzen wollen. Die restlichen 91.6 % würden die Applikation als Ergänzung zu bestehenden Übungen sehen.

### Frage 5 - Was wäre Ihre bevorzugte Art zu lernen? Sortieren Sie die nachfolgenden Lern-Möglichkeiten aufsteigend

Wir wollten wissen, wie die Studierenden am liebsten den Stoff lernen würden. Dazu konnten sie die Optionen nach ihrer Präferenz sortieren.



Abbildung 6: Auswertung Frage 5

Die Studierenden messen dem Lernen mit klassischen Unterlagen eine grosse Bedeutung bei, sehen jedoch die Möglichkeit, mit einem Spiel zu lernen, als Ergänzung.

### Frage 6 - Wie zufrieden sind sie bisher mit der Applikation „PM Game“?

Auch die generelle Zufriedenheit mit der Applikation hat einen Einfluss auf den Lernerfolg der Studenten.[4] Daher haben wir die Studenten gefragt, wie zufrieden Sie mit der Applikation sind.



Abbildung 7: Auswertung Frage 6

Die Zufriedenheit liegt mit einem Wert von 4.04 aus maximalen 6 Sternen bei 67.3 %. Im Abschnitt 3.3 wird genauer betrachtet, womit die Studenten nicht zufrieden waren.

### Frage 7 - Welche Punkte sind Ihnen bei einem Lernspiel wichtig?

Um zu identifizieren, was den Studenten an einer Lernapplikation wichtig ist, haben wir diese gebeten, eine Einschätzung zu den nachfolgenden Punkten abzugeben:

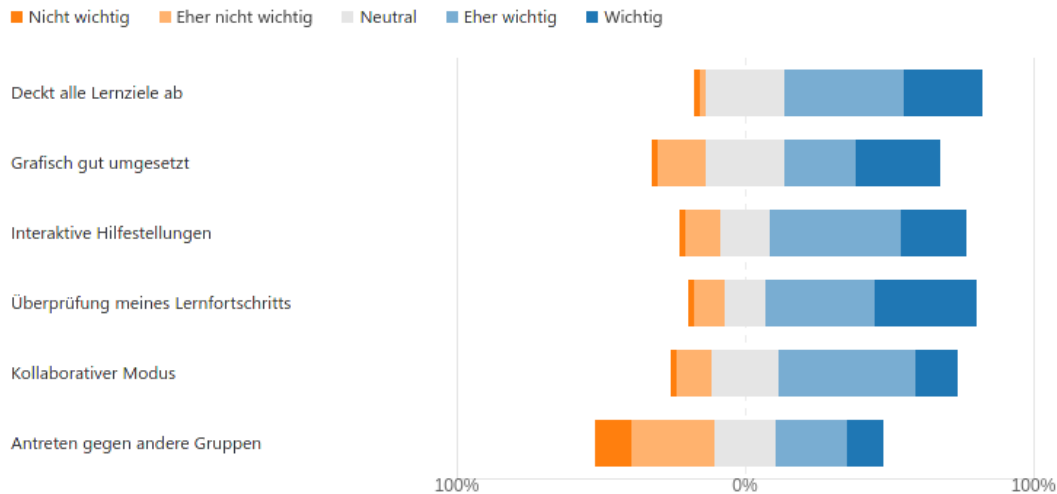


Abbildung 8: Auswertung Frage 7

Am wenigsten wichtig war den Studenten das Antreten gegen andere Gruppen. Den eigenen Stand im Vergleich zu anderen Gruppen zu sehen kann für gewisse Studenten einen Druck darstellen und sogar demotivierend wirken. Andererseits äusserten sich auch Studenten, dass die Rangliste als Ansporn diene, ein bestmögliches Resultat zu erzielen. In Wettbewerbssituationen besteht für das Streben, die eigenen Fähigkeiten zu demonstrieren, ein positiver Zusammenhang mit guten Leistungen.[5]

Am wichtigsten war den Studenten, dass die Applikation die Lernziele gut abdeckt und der Lernfortschritt gut überprüft werden kann.

### Frage 8 - Haben Sie während des Spielens, weitere Hilfsmittel zur Unterstützung oder Berechnung verwendet? (Excel Sheets, PowerBI Dashboards etc.)

Da die Simulation nur grundlegende Informationen zum simulierten Projekt ausgibt, wollten wir von den Studenten wissen, ob externe Hilfsmittel verwendet wurden um die Daten zu sammeln und für die nächsten Runden Berechnungen anzustellen.

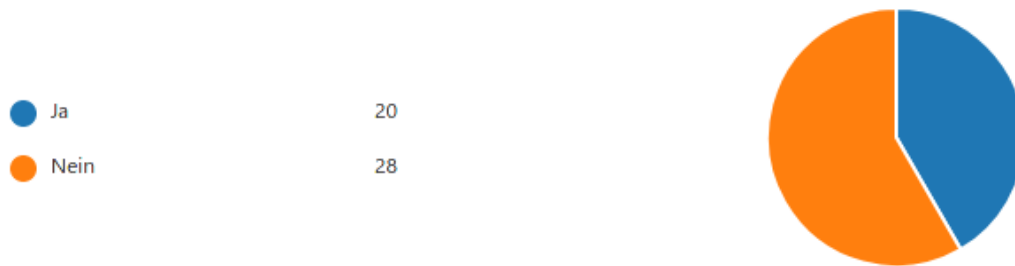


Abbildung 9: Auswertung Frage 8

Von den teilnehmenden Studenten haben 42 % angegeben, nebenbei eigene Berechnungen erstellt zu haben. Wir nehmen an, dass Studenten, welche eigene Hilfsmittel erstellt haben, sich intensiver mit den Zusammenhängen auseinandergesetzt haben und daher eine bessere Prüfungsvorbereitung erfahren haben.

Aus dem persönlichen Feedback eines Studenten, welcher angab neben der Simulation eigene Berechnungen angelegt zu haben, können wir entnehmen, dass der Student einen persönlichen Lernvorteil erzielen konnte.

Es hat mir geholfen ein Projekt nach Ressourcen zu planen (Excel Berechnungen etc.) und die Begrifflichkeiten zu festigen. [...]

### Frage 9 - Konnten Sie dem Lernen mit der Applikation „PM Game“ einen persönlichen Lernvorteil abgewinnen?

Mit dieser Frage sollten die Studenten einschätzen, ob Sie durch die Verwendung der Applikation „PM Game“ einen Lernvorteil erzielen konnten.

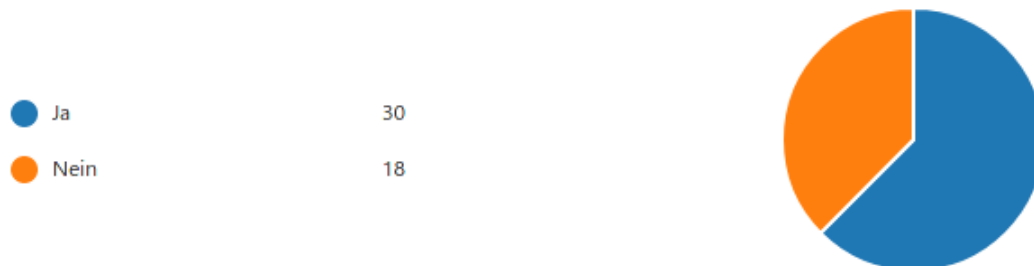


Abbildung 10: Auswertung Frage 9

62.5 % der Studenten gaben an, dass sie dem Lernen mit der Applikation „PM Game“ einen persönlichen Lernvorteil abgewinnen konnten. Im Umkehrschluss bedeutet dies jedoch auch, dass 37.5 % keinen persönlichen Lernvorteil sehen konnten.

## Frage 10 - Feedback von Studenten welche keinen Lernvorteil erzielen konnten

Uns hat interessiert, warum diese Studenten keinen Lernvorteil erzielen konnten. Daher haben wir jeweils auch um ein schriftliches Feedback gebeten. Die erhaltenen Antworten haben wir ausgewertet und dabei folgende Hauptprobleme identifiziert:

Problem	Anzahl	Prozentualer Anteil
Einfluss von Faktoren zu wenig sichtbar	5	46.6 %
Applikation noch in der Entwicklung	3	25 %
Applikation gleicht eher einem Glückspiel	3	25 %
Kein Bezug zum Unterricht	1	8.3 %

Tabelle 3: Auswertung des Feedbacks „keinen Lernvorteil erhalten“

**Einfluss von Faktoren zu wenig sichtbar** Der grösste Teil dieser Studenten hatte Mühe, bei einigen Mechanismen den Einfluss auf den Projektverlauf zu sehen. So wurde z. B. in einem persönlichen Feedback folgende Aussage gemacht:

Viele Faktoren wie der Projektleiter hatten keinen Einfluss auf den Spielverlauf. Auch würde ich in einem realen Projekt anders vorgehen da die Erfüllung des Auftrags und nicht das Gewinnen im Vordergrund steht. [...]

Ein anderer Student äusserte ähnliche Kritik:

Die in der Vorlesung erlernten Themen und Lernziele wurden nicht gut abgedeckt, und z. B. gerade beim Tab Projektmanager war nicht ersichtlich was die Zeitinvestition für Auswirkungen hat.

Dieses Problem kann durch bessere Beschreibungen in der Applikation gelöst werden. Möglicherweise kann auch eine vorhergehende Einführung in die Funktionen sowie deren Auswirkungen vorgenommen werden, um Unklarheiten auszuräumen.

**Applikation noch in der Entwicklung** Von einigen Studenten wurde bemängelt, dass die Applikation noch zu wenig ausgereift sei, um damit einen Lernvorteil zu erzielen.

So äusserte sich ein Student:

[...] Trotzdem würde ich mir wünschen, dass die Applikation zumindest die Daten richtig ablegen würde, sodass die Applikation wirklich effektiv verwendet werden kann. Einige Male wurden die Daten leider nicht richtig übernommen. [...]

Ein anderer Student bemängelte, dass die Simulation noch zu abstrakt sei und gewisse Funktionen fehlen:

Persönlich finde ich das das Tool noch ein wenig zu abstrakt ist. (Gedanke war nicht was ist für ein Projekt sinnvoll, sondern was wird das Tool machen) [...] Ich denke dennoch das das Tool mit einigen Erweiterungen und etwas feinschliff in einen Zustand gebracht werden kann indem man einen deutlichen Lernvorteil ziehen kann.

Wir waren uns bewusst, dass in der Testing-Phase Fehler auftreten können. Wir verstehen aber die betreffenden Studenten. Es ist unerfreulich, wenn Eingaben nicht sauber übernommen werden, so wie es bei den Change Requests anfänglich der Fall war. Die Live Tests waren für uns jedoch sehr hilfreich und haben wertvolle Informationen geliefert, von denen die finale Version sehr stark profitierte. Die Applikation könnte bezüglich Stabilität und Performance vom ersten bis zum letzten Live Test erheblich verbessert werden.

Es ist davon auszugehen, dass die Applikation weiterentwickelt wird und somit bestehende Funktionen verbessert sowie weitere Funktionen hinzugefügt werden. Dadurch werden Studenten künftig einen stärkeren Lernvorteil aus der Applikation ziehen können.

**Applikation gleicht eher einem Glückspiel** Einige Studenten hatten das Gefühl, keinen richtigen Einfluss auf den Projektverlauf nehmen zu können.

Meine Kritik bezieht sich nicht direkt auf die Umsetzung, sondern auf das Konzept des Spiels. Bei diesem Modul geht es um Projektmanagement, das Spiel ist aber eher eine Pokerrunde. Mitarbeiteranzahl, Qualität und Kosten von Mitarbeitern auf dem Markt sowie das Arbeitspensum und Einfluss des Projektleiters sind nicht abschätzbar und fast gar nicht beeinflussbar. [...]

Für diesen Umstand dürften ähnliche Gründe wie bei „Einfluss von Faktoren zu wenig sichtbar“ verantwortlich sein und daher sollte das Problem mit den gleichen Lösungen mitigiert werden können.

### Frage 11 - Feedback von Studenten welche einen Lernvorteil erzielen konnten

Von den Studenten, welche angaben, dass sie einen persönlichen Lernvorteil erzielen konnten, wollten wir wissen, was Ihnen am meisten geholfen hat. Diese Antworten haben wir ausgewertet und konnten so folgende Hauptgründe identifizieren.

Hervorgehobene Punkte	Anzahl	Prozentualer Anteil
Generelle Konzepte	7	36.9 %
Change Request	6	31.6 %
Risikomanagement	2	10.5 %
Direktes Feedback	2	10.5 %
Verlauf eines ganzen Projekts erleben	2	10.5 %

Tabelle 4: Auswertung des Feedbacks „konnte Lernvorteil erzielen“

**Generelle Konzepte** Viele Studenten teilten uns in ihrem Feedback mit, dass das Zusammenspiel verschiedener Punkte des Projektmanagements geholfen hat, die Aspekte eines Projekts besser zu verstehen.

Das Zusammenspiel der unterschiedlichen Werte im Gesamtüberblick betrachten. Da bei Übungen sonst nicht alles zusammen verwendet wird. Das Mindset erhalten, welches danach auch wirklich verwendet werden muss.

Die einzelnen Komponenten des Projektmanagements in Kombination zu erleben und zu verstehen.

Es hat geholfen die Konzepte in der Praxis anzuwenden und dadurch besser zu verstehen.

**Change Request** Das Formular um einen Change Request zu verfassen sowie dessen Auswirkungen auf das Projekt wurden von vielen Studenten hervorgehoben:

[...] Die Wichtigkeit der Kommunikation mit dem Change Board und der Change Requests ist gut hervorgekommen.

Gut fürs Kennenlernen von Change Requests

**Risikomanagement** Die Dynamik im Risikomanagement hat gemäss den Rückmeldungen ebenfalls zu einem besseren Verständnis von Risiken geführt.

Ein Student äusserte hierzu, dass die Auswahl der zu reduzierenden Risiken nicht trivial ist:

Ich konnte erkennen dass Risikomanagement sehr schwer ist und einige Sachen zum Mitigieren teurer sind als wenn man sie belässt. Ich habe die Statistiken in einem Projekt nachverfolgen können und nun sagen mir die Werte mehr.

Ein anderer Student zog aus der Simulation die Erkenntnis, dass Risiken teilweise auch unerwartet eintreten können:

Risiken können eintreten auch wenn man nie gedacht hätte, dass diese wirklich an diesem Punkt der Projektphase eintreten können.

**Direktes Feedback** Eine grosse Stärke des „PM Game“ ist, dass die rundenbasierte Applikation mit der gesamten Klasse live in den Übungslektionen verwendet wird. So kann nach jeder Runde mit den Studierenden den Verlauf des Projekts besprochen werden. Dieses Besprechungen und das direkte Feedback wurde von den Studenten geschätzt:

Der grösste Nutzen, der mit der Applikation verbunden war, lag im direkten Feedback, welches man erhalten hat auf seine Entscheidungen. So konnte man

seine Strategie iterativ anpassen und versuchen, sie auf die nächste Runde zu verbessern. [...]

Diskussionen in der Gruppe und Vergleiche pro Durchgang habe andere denkwiese und Strategien offen gelegt.

**Verlauf eines ganzen Projekts erleben** Schlussendlich wurde auch geschätzt, dass der Verlauf eines Projektes erlebt werden konnte.

Ich konnte zum ersten Mal erleben, wie sich Projekt-Management Entscheidungen auf den Verlauf eines Projekts auswirken. Ich denke, mir bleibt dieser Verlauf durch das „selber erleben“ präsent, wodurch ich bei einem nächsten Projekt/Spiel besser weiss, was es alles zu beachten und zu entscheiden gibt.

Das Spiel veranschaulicht sehr gut den Prozess eines Projekts anhand eines Burndown-Charts.

### 3.4. Ergebnis & Schlussfolgerungen

Wir schliessen aus dem Feedback der Studenten, dass die Verwendung von „PM Game“ zu einem Wissensaufbau führt. Dieser scheint vor allem praktischer Natur zu sein, weshalb die Studenten eher einen Vorteil für reale Projekte als für die Prüfung sehen.

Einige Studenten gaben auch an, keinen Wissensaufbau durch die Verwendung von „PM Game“ erzielt zu haben. Hier haben wir aus den Feedbacks Massnahmen abgeleitet, um den Anteil dieser Studenten in einer künftigen Version der Applikation zu reduzieren:

- Einleitende oder interaktive Erklärungen zu den Funktionen.
- Erweiterung des Funktionsumfangs um ein Projekt detailgetreuer abbilden zu können.<sup>6</sup>
- Bessere Möglichkeiten bieten den Lernfortschritt zu überprüfen.
- Stabilere Version der Applikation bereitstellen um den Spielfluss nicht zu unterbrechen.

Die Applikation bietet somit eine gute Basis, welche in späteren Projekten noch weiter ausgebaut werden kann.

---

<sup>6</sup>Im Abschnitt 4 wird eine Liste mit entsprechenden Funktionen beschrieben

## 4. Betriebswirtschaftliche Fragestellung

### 4.1. Ausgangslage & Aufgabenstellung

In diesem Abschnitt möchten wir die Frage klären, was notwendig wäre, um die Software aus Sicht Ostschweizer Fachhochschule als Trainingsdienstleistung für Externe anzubieten. Für die Beantwortung dieser Fragestellung vergleichen wir zwei bereits in der vorhergehenden Studienarbeit untersuchte Anbieter für Projektsimulationen<sup>[6]</sup> mit der Applikation „PM Game“. Dieses Mal liegt der Fokus jedoch nicht auf der Wirtschaftlichkeit von „PM Game“, sondern was die Applikation voraussetzen muss, um als externe Trainingsdienstleistung konkurrenzfähig zu werden.

Die Anbieter für Projektsimulationen sind:

**SimulTrain** Die Software des Schweizer Anbieters STS<sup>7</sup> bietet eine realitätsnahe Simulation eines vordefinierten Software-Projekts. Sie wurde gezielt für den Einsatz bei Managementschulungen entwickelt. 4er-Teams arbeiten jeweils zusammen über ein Gerät an einer Simulation und übernehmen die Rolle des Projektleiters. Die Teams können nicht gegeneinander antreten. Eine direkte Vergleichsmöglichkeit mit anderen Gruppen fehlt somit.

**Cesim Project** Der Schwedische Anbieter Cesim<sup>8</sup> entwickelt bereits seit 1996 Managementsimulationen und bietet neben der Projektsimulation auch Simulatoren für diverse andere Einsatzzwecke. Mit Cesim Project befindet sich auch eine Applikation im Sortiment, welche ein vordefiniertes Software-Projekt simuliert. Die Spieler verwalten zu viert ein Entwicklerteam, welches in einem vorgefertigten Szenario eine Applikation umsetzt.

### 4.2. Analyse

Für die Analyse der Mitbewerber haben wir die jeweilige Online Demo sowie deren Dokumentation genauer untersucht.

Der Anbieter Simultrain<sup>9</sup> bietet eine öffentlich zugängliche Demo der Applikation an. Bei Cesim muss eine Demo beim Hersteller angefragt werden<sup>10</sup>. Den Zugang haben wir anschliessend per E-Mail erhalten.

#### 4.2.1. Mandantenfähigkeit

Ein mandantenfähiges System unterstützt mehrere Kunden auf derselben Plattform. Somit muss nicht für jeden Kunden eine eigene Instanz der Software hochgefahren werden.

---

<sup>7</sup><https://www.simultrain.swiss>

<sup>8</sup><https://www.cesim.com>

<sup>9</sup><https://simultrain.swiss/smt12/?pwd=0>

<sup>10</sup><https://www.cesim.com/request-a-business-simulation-demo>

**Aktuelle Situation:** Die Applikation „PM Game“ unterstützt derzeit nur einen Mandanten. Es müsste daher für jeden Mandant eine eigene Instanz auf einem Server inkl. einer eigenen Datenbank hochgefahren werden. Dies erachten wir als äusserst unpraktisch.

Theoretisch kann für jeden Mandanten zwar ein eigenes Spiel angelegt werden, die Spielerdaten und die Spieldaten sind jedoch global nur für einen Admin zugänglich. Dies kann in einem kleinen Rahmen noch funktionieren, wird aber schnell unübersichtlich bei mehreren Mandanten auf einer Instanz von „PM Game“.

**Chance:** Kann die Applikation „PM Game“ so umgesetzt werden, dass mehrere Mandanten auf einer Instanz laufen können, würde dies die Inbetriebnahme der Applikation für eine neuen Mandaten erheblich vereinfachen und die Kosten der Applikation senken. Die Applikation wird dadurch also insgesamt wettbewerbsfähiger.

#### 4.2.2. Funktionsumfang

Damit eine Projektsimulation attraktiv für Kunden ist, sollte diese die behandelnde Thematik (in unserem Fall Projektmanagement) gut abdecken.

**Aktuelle Situation:** „PM Game“ bringt einen soliden Grundstock an Funktionen mit. Für eine vollständige Simulation eines Projekts reicht dies jedoch nicht aus. Hierzu müssten bestehende Funktionen verfeinert und weitere Funktionen hinzugefügt werden.

**Vergleich Mitbewerber:** Im Funktionsumfang sind sich Cesim Project und Simultrain sehr ähnlich. Beide bieten detaillierte Steuerungsmöglichkeiten für die Mitarbeiter. So werden bei beiden Applikationen die Tasks auf die Mitarbeiter verteilt.

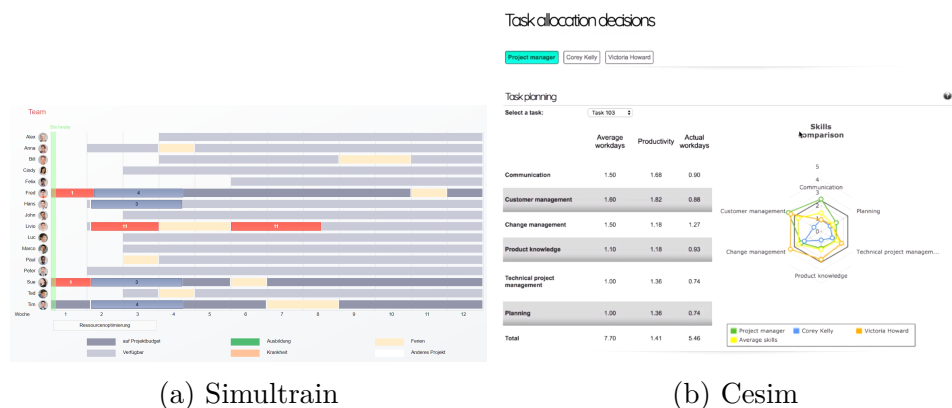


Abbildung 11: Zuteilung der Tasks bei Simultrain und Cesim

Bei Cesim fehlt ein Risikomanagement. Bei Simultrain hingegen können Risiken mit verschiedenen Möglichkeiten reduziert werden. Diese haben unterschiedliche Kosten und unterschiedliche Auswirkungen, sollte das Risiko eintreten.

Risiko 3. Der Server, auf dem sich die PM-Software befindet, stürzt ab.		Wahrscheinlichkeit: 0.05	Mehrkosten: 6700
		Bewältigung: Geplant	Mehraufwand, Tage: 0
	Bewältigung	Bewältigungskosten	
<input type="checkbox"/>	Ich kaufe für 4000 - Ersatzmaterial.	4000	
<input checked="" type="checkbox"/>	Ich bereite eine Liste von Lieferanten vor und verhandle die Konditionen für eine schnelle Lieferung.	500	
<input checked="" type="checkbox"/>	Ich bilde die Teammitglieder aus, um das Material rasch und effizient auszutauschen.	800	
<input type="checkbox"/>	Ich unterzeichne einen Vertrag, um den Server auszuwechseln und eine zusätzliche Datensicherung vorzunehmen.	2000	
<input type="checkbox"/>	Ich akzeptiere das Risiko und treffe keine dieser Massnahmen.	0	

Abbildung 12: Reduktion eines Risikos bei Simultrain

Bei Cesim Project und Simultrain wird die Simulation in einer Gruppe durchgespielt und erfordert die Kommunikation untereinander. Bei beiden Applikationen kann der Spieler die Resultate seiner Spieler einsehen, eine direkte Einflussnahme auf den Projektverlauf ist jedoch nicht möglich, da die Szenarien von den Applikationen fix vorgegeben sind.

Simultrain wie auch Cesim Project berücksichtigen die Qualität der geleisteten Arbeit. Auf die Qualität hat z. B. die Zufriedenheit der Mitarbeiter einen erheblichen Einfluss. Bei Simultrain wird diese massgeblich über die Arbeitslast (Überstunden) sowie planbare Freizeitaktivitäten (Gemeinsame Mittagessen, Grillabende) gesteuert.

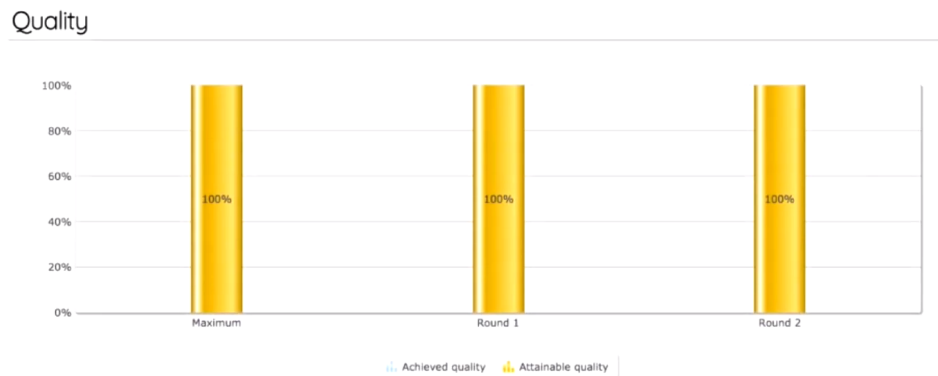


Abbildung 13: Auswertung der Qualität der geleisteten Arbeit bei Cesim Project

**Chance:** Mit einer Verfeinerung der bestehenden sowie dem Hinzufügen weiterer Funktionen könnte „PM Game“ erheblich an Attraktivität gewinnen. Die Funktionen, welche

von den Mitbewerbern angeboten werden, lassen sich grundsätzlich auch bei „PM Game“ realisieren.

Aus dem Feedback der Studenten sowie, dem Vergleich mit bestehenden Lösungen sowie unserer eigenen Erfahrung haben wir folgende Punkte eruiert:

- Die Mitarbeiter sollten nicht in jeder Runde wieder neu eingestellt werden müssen.
- Die Mitarbeiter sollten einem spezifischen Task zugewiesen werden. Tasks können hierbei bestimmte Mitarbeitertypen erfordern.
- Tasks sollten in einem Gantt Diagramm Diagramm und/oder einem Netzplan abgebildet werden.
- Das Training sollte auf spezifische Mitarbeiter angewandt werden. Ggf. sollte der Mitarbeiter während einem Training nicht oder nur eingeschränkt an dem Projekt arbeiten können.
- Der Projektleiter sollte einen Einfluss auf die Motivation der Mitarbeiter haben.
- Für die Reduktion von Risiken sollten jeweils verschiedene Möglichkeiten zur Mitigation möglich sein.
- Der Projektleiter sollte einen Einfluss auf die Qualität der geleisteten Arbeit haben.

#### 4.2.3. Mehrsprachigkeit

Eine Applikation kann ein grösseres Zielpublikum erreichen, wenn mehrere Sprachen unterstützt werden.

**Aktuelle Situation:** Derzeit wird von „PM Game“ nur Deutsch unterstützt. Die Applikation bietet keine Unterstützung für mehrere Sprachen. Diese Funktionalität müsste daher von Grund auf implementiert werden.

**Vergleich Mitbewerber:** Cesim und Simultrain bieten ihre Simulationen in mehreren Sprachen an. Der Spieler kann die Sprache jeweils selbständig Einstellen.

**Chance:** Wäre „PM Game“ in weiteren Sprachen wie Englisch oder Französisch verfügbar, so würde sich die potentielle Zielgruppe auf einen Schlag wesentlich vergrössern. Auf die Schweiz bezogen wäre Französisch eine wichtige Sprache, welche unterstützt werden sollte, da es sich um die zweithäufigste verwendete Landessprache handelt. Englisch hingegen ist in vielen international ausgerichteten Schweizer Unternehmen verbreitet und sollte daher ebenfalls angeboten werden.

#### 4.2.4. Performance

Weist eine Webapplikation eine schlechte Performance auf, so ist dies für Benutzer der entsprechenden Applikation häufig ein Grund zu einem Konkurrenten zu wechseln.<sup>[7]</sup> Schlecht Performance zeigt sich unter anderem durch:

- Lange Ladezeiten der Webseite
- Verzögerte Eingaben
- Fehlermeldungen z. B. aufgrund Ladefehler

**Aktuelle Situation:** Die Applikation „PM Game“ ist für die Durchführung im PmQm Unterricht ausgelegt. Es haben daher nicht mehr als gut 25 Benutzer gleichzeitig auf der Applikation gearbeitet. In einem Lasttest konnten wir bei 180 Benutzer ebenfalls eine akzeptable Performance erzielen. Für grössere Anzahl Benutzer müssten jedoch Optimierungen vorgenommen werden.

Aus unserer Sicht können folgende Punkte optimiert werden:

- Die Datenbank müsste mit spezifischen Indexen für gewisse Abfragen optimiert werden.
- Die minimale Hardware Anforderung müsste für mehr Benutzer erhöht werden.
- Gewisse Daten könnten auch in einen Cache abgelegt werden, anstelle diese direkt in die Datenbank zu persistieren.

Ein generelles Performance Problem sehen wir bei der Applikation „PM Game“ jedoch nicht.

**Vergleich Mitbewerber:** Wir konnten bei keinem der Mitbewerber ein Problem bei der Performance feststellen.

**Chance:** Damit die Applikation „PM Game“ wettbewerbsfähig ist, sollte diese gut skalieren, damit diese auch bei einer grösseren Benutzeranzahl performant läuft. Die Version 2.0 von „PM Game“ konnte zwischen dem ersten und zweiten Live Test durch Verbesserung an der Software erheblich optimiert werden. Eine detaillierte Analyse der Applikation würde helfen diese noch weiter zu optimieren und für grössere Benutzerzahlen verwendbar zu machen.

#### 4.2.5. Sicherheit

Die Sicherheit einer Applikation ist oftmals nicht direkt sichtbar als Anwender. Für den Betreiber einer Projektsimulation ist es jedoch erstrebenswert, dass die Benutzer keine Eingaben manipulieren können und die Benutzerdaten sicher sind.

Würden sich Eingaben der Benutzer manipulieren lassen, wäre die Applikation für Unternehmen nicht mehr interessant, da das Resultat einer durchgeführten Simulation wenig Aussagekraft hätte. Der Reputationsschaden wenn Benutzerdaten abhandenkommen kann ebenfalls immens sein und sollte daher zwingend verhindert werden.

**Aktuelle Situation:** Bei der Entwicklung der Version 2.0 von „PM Game“ wurde auf die Sicherheit der Applikation geachtet. So werden die sensitiven Benutzerdaten (vorherhand die Passwörter) als Salted Hash in der Datenbank abgelegt. Auch wurden die Formulare gegen XSS und SQL Injection abgehärtet. In den Live Tests sind ebenfalls noch einige Flaws aufgefallen, welche behoben werden konnten.

Für eine öffentliche Anwendung der Software ist ein umfassender Sicherheitsaudit empfohlen. Die Live Tests haben uns gezeigt, dass sich auch bei einer sorgfältigen Entwicklung Fehler durch Unkenntnis oder konzeptionellen Schwächen einschleichen können.

**Vergleich Mitbewerber:** Bei unserer Recherche konnten wir keine Informationen über etwaige Datenleaks bei Cesim oder Simultrain finden. Eine umfassende Überprüfung der Sicherheit konnte aber im Rahmen dieser Arbeit nicht durchgeführt werden.

**Chance:** Das Vertrauen in eine Applikation ist wichtig, damit diese von Kunden akzeptiert wird. Wir erachten es daher als wichtig die Sicherheit der Applikation im Auge zu behalten und regelmässig zu überprüfen.

#### 4.2.6. Demo Modus

Eine Demo Modus erlaubt es potentiellen Kunden die Applikation in einem kleineren Umfang zu testen und somit eine Entscheidungsgrundlage zu bilden, ob dies die für ihn passende Applikation ist.

**Aktuelle Situation:** Die Applikation „PM Game“ hat derzeit keinen Demo Modus.

**Vergleich Mitbewerber:** SimulTrain hat einen Demo Modus, welcher selbständig über die Webseite verwendet werden kann. Bei Cesim kann eine Demoinstanz über die Webseite beantragt werden.

**Chance:** Durch einen Demo Modus kann die Applikation einfach einem potentiellen Interessenten gezeigt werden. Eine Integration in die bestehende Applikation sollte technisch mit vertretbarem Aufwand realisierbar sein und ist daher zu empfehlen.

#### 4.2.7. Dokumentation

Eine Dokumentation hilft dem Spielleiter wie auch den Spieler die Spielmechanik der Simulation besser zu verstehen.

**Aktuelle Situation:** Eine umfassende Dokumentation fehlt derzeit für „PM Game“. Dies umfasst einerseits die ingame Dokumentation der Funktionen, als auch ein umfassendes Handbuch für etwaige Spielleiter. Auch fehlen auf das „PM Game“ abgestimmte Schulungsunterlagen, welche zur Vorbereitung auf eine Simulation mit „PM Game“ genutzt werden können.

**Vergleich Mitbewerber:** Alle Mitbewerber bieten auch Dokumentationen für die Spielleiter und auch die Spieler an. Bei Cesim wird für den Spielleiter ein umfassendes Benutzerhandbuch mitgeliefert sowie ein Projekt Guide<sup>11</sup>. Bei Simultrain erhält der Spielleiter eine umfangreiche Anleitung inkl. best Practices sowie einen Guide für die Spieler<sup>12</sup>.

**Chance:** Eine aussagekräftige Dokumentation scheint ein wichtiger und nachgefragter Bestandteil einer solchen Trainingsdienstleistung zu sein. Um den Mitbewerbern ebenbürtig zu sein, sollte für „PM Game“ ebenfalls eine Dokumentation erstellt werden. Diese Dokumentation empfiehlt zusammen mit einem Experten für Projektmanagement und Qualitätsmanagement zu erstellen. Nicht nur die technischen Abläufe der Simulation, sondern auch die theoretischen Zusammenhänge sollten für Projektleiter wie auch Spieler festgehalten werden.

#### 4.2.8. Spiel Verwaltung

Damit die Applikation im Dauerbetrieb verwendet werden, kann muss diese entsprechende Funktionen bereitstellen, um die Spiele und Benutzer zu verwalten.

**Aktuelle Situation:** Es fehlen derzeit noch Funktionalitäten um beendete Spiele zu archivieren oder auch laufende Spiele abzubrechen. Die Applikation war bisher darauf ausgelegt um während einem Semester gespielt zu werden. Im neuen Semester wird einfach die ganze Applikation zurückgesetzt. Dies ist jedoch für einen Dauerbetrieb der Applikation nicht praktikabel.

Aus unserer Sicht fehlen daher noch folgende Funktionen in der Applikation.

- Spiel abbrechen
- Spiel archivieren

---

<sup>11</sup><https://www.cesim.com/download-the-cesim-project-guide-book>

<sup>12</sup><https://www.simultrain.swiss/#Resources>

- Archivierte Spiele einsehen
- Spiel löschen
- Benutzer löschen

Ohne diese Funktionen ist zwar ein regulärer Spielbetrieb möglich, aber die Organisation der Spiele und Benutzer erschwert sich zunehmend mit einer höheren Anzahl an Benutzern und Spielen auf einer Instanz der Applikation.

**Chance:** Die Implementation dieser Funktionen erachten wir als zwingend nötig um einen Dauerbetrieb der Applikation zu ermöglichen.

### 4.3. Auswertung Fragebogen

In unserem Fragebogen haben wir vier Fragen platziert, um die Einschätzung der Studenten zur Wirtschaftlichkeit der Applikation zu erhalten. Die Studenten konnten auf einer Skala von 0 bis 10 angeben, wie sehr Sie der Aussage zustimmen. (0 = keine Zustimmung, 10 = volle Zustimmung)

#### Frage 12 - Würden Sie die Applikation „PM Game“ auch im Rahmen eines Kurses oder Weiterbildung auch einem Unternehmen empfehlen?

Wir wollten wissen, ob die Studenten den Einsatz der Applikation in einem Unternehmen sehen.

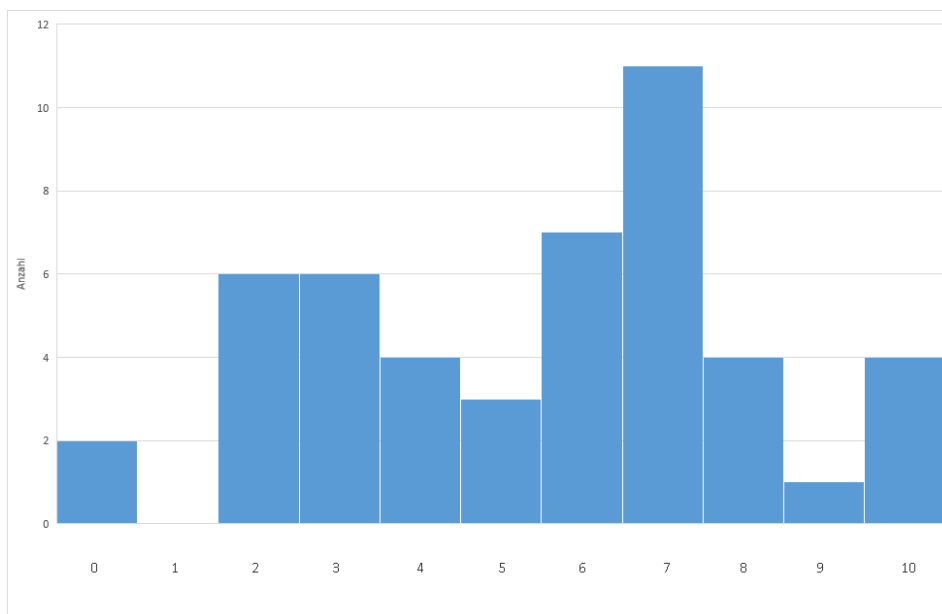


Abbildung 14: Auswertung Frage 12

Die Studenten tendieren leicht dazu „PM Game“ einem Unternehmen weiterzuempfehlen. Dies ist eine gute Basis und könnte durch Verbesserungen der Applikation sicherlich noch ausgebaut werden.

**Frage 13 - Würden Sie eine Unterrichtsstätte mit starkem Angebot an interaktiven Lernhilfen gegenüber einer Lernstätte mit wenig interaktiven Lernhilfen bevorzugen?**

Ebenfalls wollten wir wissen, ob die Studenten eine Unterrichtsstätte mit einem starken Angebot an interaktiven Lernhilfen gegenüber einer Lernstätte mit wenig interaktiven Lernhilfen bevorzugen würden.

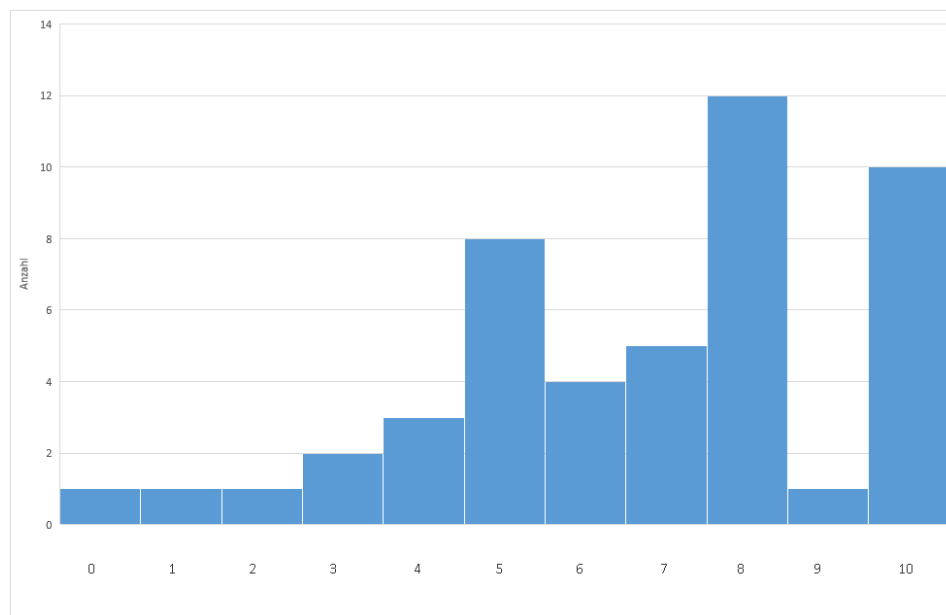


Abbildung 15: Auswertung Frage 13

Es scheint als könnte der Einsatz einer Applikation wie „PM Game“ für eine Fachhochschule wie die OST ein Wettbewerbsvorteil darstellen. Die Verwendung der Applikation könnte daher auch für andere Unterrichtsstätten von Interesse sein.

**Frage 14 - Würden Sie solche Applikationen auch für andere Unterrichts-Module wünschen?**

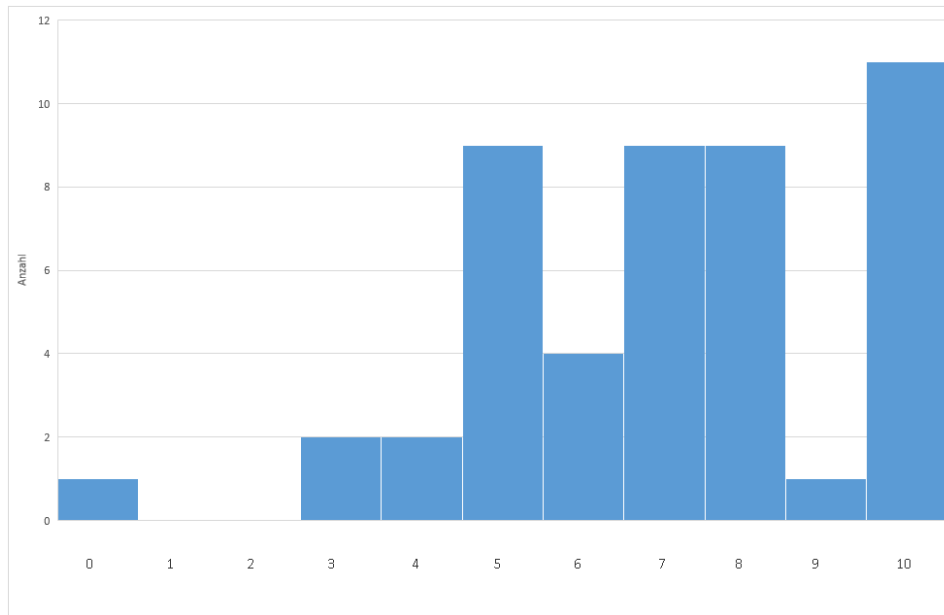


Abbildung 16: Auswertung Frage 14

Der Grossteil der Studenten würde eine entsprechende Applikation in anderen Modulen ebenfalls begrüßen.

**Frage 15 - Sollten die Leistungen in einem Game wie dem „PM Game“ in die Notengebung miteinfließen?**

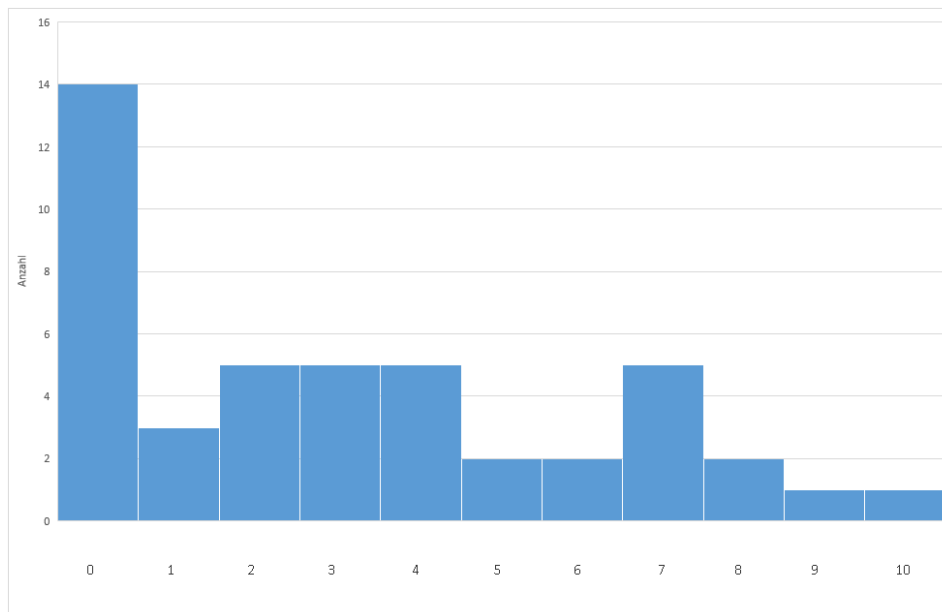


Abbildung 17: Auswertung Frage 15

Davon die Notengebung durch die Leistung eines Studenten in einer interaktiven Applikation wie dem „PM Game“ zu bestimmen, halten die gefragten Studenten sehr wenig. Die Studenten sehen in solchen Applikation eher eine Ergänzung für die Übungen. Ein Ausbau der Applikation zu einem Tool, welches auch Prüfungssituationen zulässt, scheint daher wenig vielversprechend.

**Frage 16 - Können Sie sich vorstellen, dass der gesamte Unterricht eines Modules über eine Game-Applikation abgedeckt werden könnte. (Wissens-Vermittlung, Repetition, Übungen und Prüfungen)?**

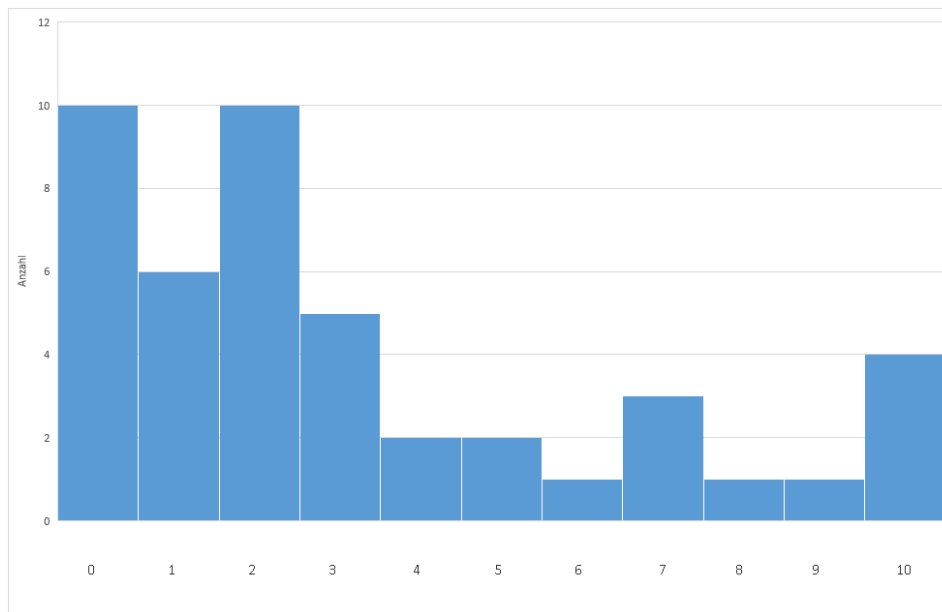


Abbildung 18: Auswertung Frage 16

Ebenfalls eher negativ sehen die Studenten die vollständige Abwicklung einer Unterrichts Veranstaltung durch eine Lernapplikation. Zu einem ähnlichen Ergebnis kommt auch eine Untersuchung der Universität Rostock. Dabei kommt die Autorin Christina Wigger zum Schluss, dass E-Learning Angebote die konventionelle Lehre nicht ersetzen, sondern lediglich ergänzen und vertiefen sollten. [8].

Der Fokus bei einer Weiterentwicklung der Applikation sollte daher vor allem auf den ergänzenden Einsatz für Übungen gelegt werden.

#### **4.4. SWOT Analyse „PM Game“**

Die SWOT Analyse erlaubt auf einen Blick die Stärken und Schwächen von „PM Game“ zu sehen.

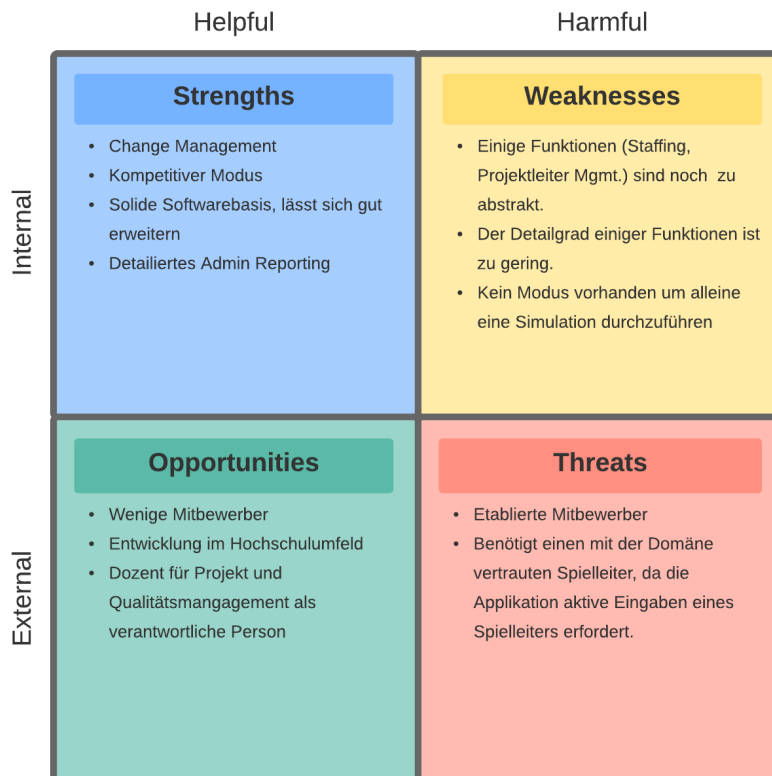


Abbildung 19: SWOT Analyse „PM Game“

## 4.5. Ergebnis & Schlussfolgerungen

Wir sehen in dem Feedback der Studenten das Potential, die Applikation auch für externe Schulen und Unternehmen zur Verfügung zu stellen. Die aktuelle Version 2.0 ist hierfür jedoch aus unserer Sicht noch nicht geeignet. Daher resultiert aus der Untersuchung eine Liste mit Punkten, welche umgesetzt werden sollten, damit die Applikation in einer späteren Version für einen externen Einsatz verwendet werden kann.

MUSS Punkte:

- Die Grundfunktionen um ein Spiel zu verwalten müssen umgesetzt werden.
  - Spiel abbrechen.
  - Spiel archivieren.
  - Spiel löschen.
  - Benutzer löschen.
  - Archivierte Spiele einsehen

SOLL Punkte:

- Die Applikation sollte verschiedene Mandanten unterstützen, damit ein unkomplizierter und wirtschaftlicher Betrieb für mehrere Kunden möglich ist.
- Die Performance der Applikation sollte überprüft und ggf. optimiert werden.
- Die Sicherheit der Applikation sollte überprüft werden.
- Eine umfassende Dokumentation für den Spielleiter mit Best Practices sollte erstellt werden.
- Der Funktionsumfang sollte erweitert werden.
  - Mitarbeiterzuteilung über mehrere Runden.
  - Mitarbeiterzufriedenheit hat einen Einfluss auf das Projekt.
  - Mitarbeiter werden einem Task zugewiesen.
  - Qualität der Arbeit hat einen Einfluss auf das Projekt.
  - Risiken werden mit konkreten Massnahmen reduziert.

KANN Punkte:

- Ein Demo Modus der Applikation würde potentiell interessierten Kunden die Möglichkeit geben, die Applikation vorgängig zu testen.
- Eine Dokumentation zur Vorbereitung auf die Simulation würde den Spielern helfen die Theorie im Spiel besser anzuwenden.
- Eine Dokumentation im Spiel würde einige Fragen zur Verwendung von „PM Game“ direkt klären.

## 5. Software Engineering Dokumentation

Dieser Teil enthält alle Dokumente, die im direkten Zusammenhang mit der Programmierung der Software erstellt wurden.

### 5.1. Funktionale Anforderungen

#### 5.1.1. User Storys

Ein Teil der funktionalen Anforderungen wurden aus der vorhergehenden Studienarbeit übernommen. Es handelt sich um funktionale Anforderungen, welche nicht implementiert werden konnten. Zu Beginn der Bachelorarbeit wurden vom Projektbetreuer weitere Anforderungen in Form von User Storys vorgegeben. In der Anforderungsanalyse Phase haben wir die bestehenden User Storys geprüft und angepasst, sowie zwei neue User Stories als Vorschlag hinzugefügt. Zum Zeitpunkt des Meilensteins „Anforderungsanalyse durchgeführt“ hatten wir 13 User Storys, welche vom Projektbetreuer abgenommen wurden (siehe Abbildung 20).

Der Projektbetreuer hat zudem die User Storys von Priorität 1 bis 3 priorisiert. Die Prioritäten sind folgendermassen zu interpretieren: 1 = MUSS, 2 = SOLL, 3 = KANN.

Alle User Storys von Priorität 1 und 2 konnten wir erfolgreich umsetzen. Die User Story 12 „Ich als Spieler trete einem Spiel bei, indem ich den dazugehörigen Token eingebe“ wurde mittels Change Request an den Projektbetreuer, zugunsten während der Live Tests festgestellten Bugs, zurückgestellt. Die anderen Priorität 3 User Storys konnten wir umsetzen.

ID	Beschreibung	Quelle	Use Case	Kommentar	Priorität	Status
US01	Ich als Spielerleiter habe die Möglichkeit die Teams via E-Mail zu informieren	Vorgaben Claudio Fuchs	UC04, UC03, UC02, UC01		2	Erledigt
US02	Ich als Spielerleiter kann einen Report über die Performance ziehen	Vorgaben Claudio Fuchs	UC05		3	Erledigt
US03	Ich als Spielerleiter kann einen Katalog an Projektrisiken vorgeben	Vorgaben Claudio Fuchs	UC06		1	Erledigt
US04	Ich als Spielerleiter kann festlegen, welche Risiken in welcher Runde auftreten	Vorgaben Claudio Fuchs	UC07, UC12		1	Erledigt
US05	Ich als Spielerleiter kann definieren, welcher Typ von Mitarbeitenden für das Projekt zur Verfügung stehen	Vorgaben Claudio Fuchs	UC07		1	Erledigt
US06	Ich als Spieler habe einen groben Überblick, wie weit die anderen Teams sind (Details nur für Spielerleiter)	Vorgaben Claudio Fuchs	UC08		2	Erledigt
US07	Ich als Spieler werde per E-Mail informiert, wenn für meine Gruppe eine Eingabe ausstehend ist	Vorgaben Claudio Fuchs	UC09		2	Erledigt
US08	Ich als Spieler muss entscheiden, welche Projektrisiken ich in welcher Runde reduziere	Vorgaben Claudio Fuchs	UC10		1	Erledigt
US09	Ich als Spieler kann mir die Projektmitarbeitenden mit Ihren Skills selbst zusammenstellen	Vorgaben Claudio Fuchs	UC11		3	Erledigt
US10	Ich als Spieler ändere meine E-Mailadresse	Übernahme SA	UC13, UC14		3	Erledigt
US11	Ich als Spieler erfasse einen neuen Spieler anhand dessen E-Mail Adresse	Übernahme SA	UC16		2	Erledigt
US12	Ich als Spieler trete einem Spiel bei, indem ich den dazugehörigen Token eingabe.	Vorschlag Gruppe	UC17	Zurückgestellt	3	Offen
US13	Ich als Spielerleiter kann alle Benutzer stehen welche registriert sind	Vorschlag Gruppe	UC15		3	Erledigt

Abbildung 20: List of User Storys

### 5.1.2. Use Cases

Aus den User Storys haben wir anschliessend Use Cases erarbeitet (siehe Abbildung 21). Die Use Cases beschreiben den genauen Ablauf der entsprechenden Aktionen, sowie

deren Erweiterungen und Abhängigkeiten. Auf Basis der Use Cases wurde anschliessend die Software dazu geschrieben.

ID	Name	Abhängigkeiten
UC1	Nachricht erstellen	
UC2	Nachricht versenden	UC1
UC3	Nachricht erhalten	UC2
UC4	Nachricht beantworten	UC2
UC5	Performance Report exportieren	
UC6	CRUD Risiko	
UC7	Spiel anlegen	UC6
UC8	Überblick über andere Gruppen erhalten	
UC9	Runde beenden ankündigen	
UC10	Projektrisiko reduzieren	
UC11	CRUD Arbeiter	
UC12	Risiko eintreten lassen	UC6
UC13	E-Mail Adresse ändern	
UC14	E-Mail Adresse bestätigen	UC13
UC15	Alle User einsehen	
UC16	User erstellen	
UC17	Spiel per Token beitreten	

Abbildung 21: List of Use Cases

Im Use Case Diagramm (siehe Abbildung 22) sind zusätzlich die Abhängigkeiten unter den unterschiedlichen Use Cases sowie den Aktoren ersichtlich.

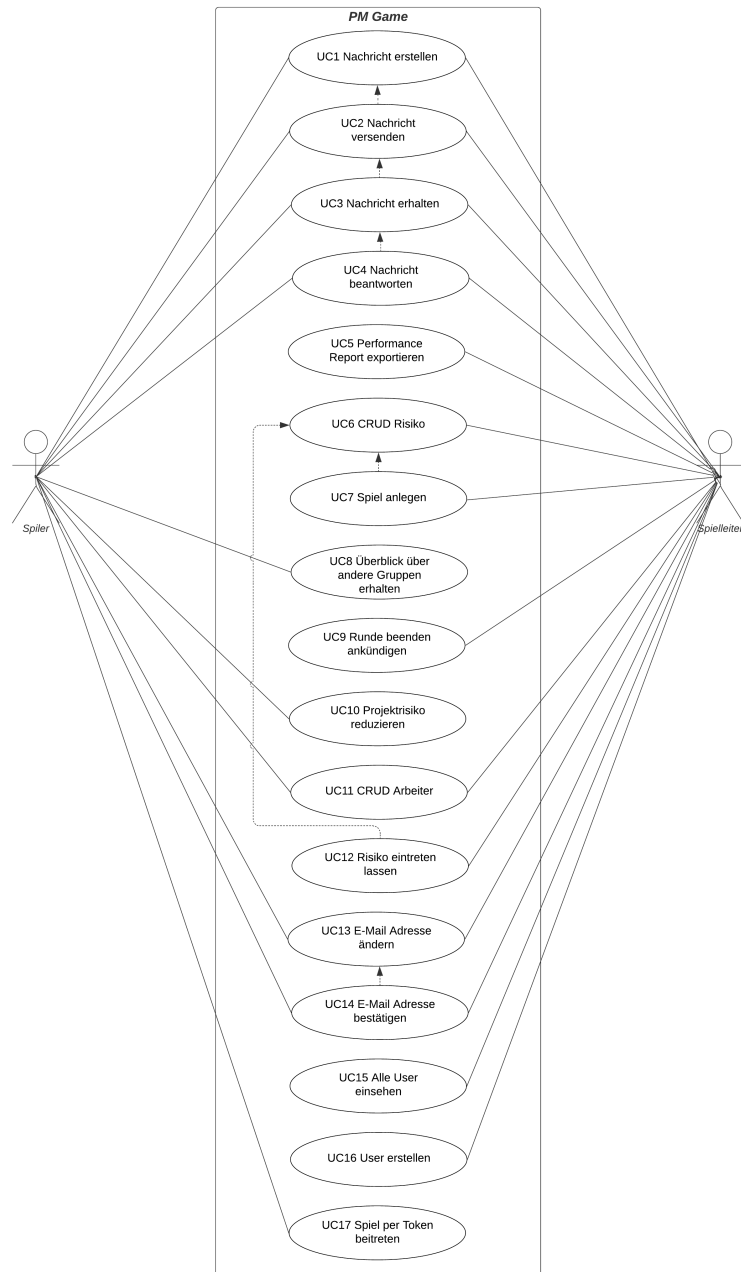


Abbildung 22: Use Case Diagram

### 5.1.2.1. Use Cases ausformuliert

Dieser Abschnitt zeigt die vollständig beschriebenen Use Cases.

ID	UC1
Name	Nachricht erstellen
Ziel im Kontext	Der Spielleiter erstellt eine neue Nachricht an einen oder mehrere Benutzer
Akteure	Spielleiter
Trigger	
Essentielle Schritte	<ol style="list-style-type: none"> <li>1. Der Spielleiter klickt auf ein aktives Spiel auf der Administratoren-Oberfläche</li> <li>2. Der Spielleiter klickt auf den Button "Benachrichtigung erstellen"</li> <li>3. Der Spielleiter wählt einen oder mehrere Spieler aus der Liste der dem Spiel zugehörigen Spieler</li> <li>4. Der Spielleiter schreibt seine Nachricht in das Textfeld.</li> <li>5. Der Spielleiter wählt ob die Nachricht nur in-game oder auch per E-Mail zugestellt wird.</li> <li>6. Der Spielleiter klickt auf den "Senden" Button</li> </ol>
Erweiterungen	<p>1a.1 Der Spielleiter navigiert auf der Administratoren-Oberfläche zur "Benutzer"- Übersicht.</p> <p>1a.2 Der Spielleiter wählt einen oder mehrere Spieler aus.</p> <p>1a.3 Der Spielleiter schreibt seine Nachricht in das Textfeld</p> <p>1a.4 Der Spielleiter wählt ob die Nachricht nur in-game oder auch per E-Mail zugestellt wird.</p> <p>1a.5 Der Spielleiter klickt auf den Button "Senden"</p>
Abhängigkeiten	-
User Stories	US1

ID	UC2
Name	Nachricht zustellen
Ziel im Kontext	Das System versendet eine vom Spielleiter oder Spieler erstellte Benachrichtigung.
Akteure	System
Trigger	UC1, UC4
Essentielle Schritte	<ol style="list-style-type: none"> <li>1. Das System erhält eine Nachricht</li> <li>2. Das System informiert alle aktiven Empfänger in-game per Benachrichtigung</li> </ol>
Erweiterungen	2a.1 Falls die Nachricht auch per E-Mail versendet werden soll, erhält der Spieler eine Benachrichtigung an seine E-Mail-Adresse mit den Nachrichten Text und einem Link um auf die Nachricht zu Antworten
Abhängigkeiten	UC1
User Stories	US1
Kommentar	Im System wird ein Objekt "Nachricht" gespeichert welches einen Empfänger und Absender hat. Diese Nachricht wird im Spiel in einer Art "Mailbox" angezeigt. Falls der Absender wünscht, dass auch eine E-Mail versendet wird, so löst das System eine Benachrichtigung an die E-Mail Adresse des Empfängers aus welche den Text der Nachricht sowie einen Link auf die Mailbox im System enthält.

ID	UC3
Name	Nachricht erhalten
Ziel im Kontext	Der Spieler oder der Spielleiter erhält eine Nachricht in-game

Akteure	Spieler, Spielleiter
Trigger	
Essentielle Schritte	<ol style="list-style-type: none"> <li>1. Der Spieler erhält einen visuellen Hinweis, dass eine neue Nachricht eingegangen ist.</li> <li>2. Der Spieler klickt auf den Hinweis</li> <li>3. Der Spieler sieht die Nachricht</li> </ol>
Erweiterungen	<p>1a.1 Der Spieler klickt auf "Nachrichten" im Game</p> <p>1a.2 Der Spieler sieht in der Übersicht alle Nachrichten</p> <p>1a.3 Der Spieler klickt auf eine Nachricht</p> <p>1a.4 Der Spieler sieht die Nachricht.</p>
Abhängigkeiten	UC2
User Stories	US1

ID	UC4
Name	Nachricht beantworten
Ziel im Kontext	Der Spieler oder Spielleiter beantwortet eine erhaltene Nachricht
Akteure	Spieler, Spielleiter
Trigger	UC3
Essentielle Schritte	<ol style="list-style-type: none"> <li>1. Der Spieler klickt in einer offenen Nachricht auf den Button "Antworten"</li> <li>2. Der Spieler trägt seine Antwort in das Textfeld ein.</li> <li>3. Der Spieler klickt auf den Button "Senden"</li> </ol>
Erweiterungen	<p>1a1. Der Spieler klickt auf "Nachrichten" im Game</p> <p>1a2. Der Spieler sieht in der Übersicht alle Nachrichten</p> <p>1a3. Der Spieler klickt auf eine Nachricht</p> <p>1a4. Der Spieler sieht die Nachricht.</p>
Abhängigkeiten	UC3
User Stories	US1

ID	UC5
Name	Performance Report exportieren
Ziel im Kontext	Der Spielleiter exportiert die aktuellen Spieldaten über die Administratoren-Oberfläche.
Akteure	Spielleiter
Trigger	
Essentielle Schritte	<ol style="list-style-type: none"> <li>1. Der Spielleiter klickt auf ein aktives Spiel</li> <li>2. Der Spielleiter klickt auf den Button exportieren</li> <li>3. Der Spielleiter erhält eine CSV File mit allen Werten für die aktuelle und vergangenen Runden</li> </ol>
Erweiterungen	
Abhängigkeiten	-
User Stories	US2

ID	UC6
Name	CRUD Risiko
Ziel im Kontext	Der Spielleiter erstellt ein neues Risiko, welches eintreten kann.
Akteure	Spielleiter

Trigger	
Essentielle Schritte	<ol style="list-style-type: none"> <li>1. Der Spielleiter klickt auf der Administratoren-Oberfläche auf "Stammdaten"</li> <li>2. Der Spielleiter sieht bereits erstellte Risiken</li> <li>3. Der Spielleiter klickt auf "Risiko hinzufügen"</li> <li>4. Der Spielleiter setzt folgende Daten <ol style="list-style-type: none"> <li>a. Risiko Name</li> <li>b. Risiko Beschreibung</li> </ol> </li> <li>5. Der Spielleiter klickt auf den Button "Speichern"</li> </ol>
Erweiterungen	<p>3a.1 Der Spielleiter wählt ein existierendes Risiko aus</p> <p>3a.2 Der Spielleiter klickt auf den Löschen Button</p>
Abhängigkeiten	-
User Stories	US3

ID	UC7
Name	Spiel anlegen
Ziel im Kontext	Der Spielleiter erstellt ein neues Spiel
Akteure	Spielleiter
Trigger	
Essentielle Schritte	<ol style="list-style-type: none"> <li>1. Der Spielleiter klickt auf "Spiel erstellen"-Button</li> <li>2. Der Spielleiter trägt einen Namen ein</li> <li>3. Der Spielleiter trägt folgende Vorgaben ein: <ol style="list-style-type: none"> <li>a. Startbudget</li> <li>b. Anzahl Runden</li> <li>c. Start Function Points</li> <li>d. Ausbildung</li> <li>e. Stakeholder</li> <li>f. Projektziele</li> <li>g. Gruppengrösse</li> <li>h. Aufwände</li> <li>i. Risiken</li> <li>j. Mitarbeitertypen</li> </ol> </li> <li>4. Der Spielleiter bestätigt das Spiel</li> </ol>
Erweiterungen	<p>2a. Der Name existiert schon</p> <p>2a1. Der Spielleiter erhält eine Warnmeldung</p> <p>2a2. Der Spielleiter wählt einen neuen Namen</p>
Abhängigkeiten	UC6
User Stories	US4
Kommentar	Dieser Use-Case erweitert einen bereits umgesetzten Use-Case aus dem ersten Projekt um die Funktion dem Spiel Risiken zuzuteilen.

ID	UC8
Name	Überblick über andere Gruppen erhalten
Ziel im Kontext	Der Spieler erhält einen groben Überblick wie weit die anderen Gruppen sind
Akteure	Spieler
Trigger	
Essentielle Schritte	<ol style="list-style-type: none"> <li>1. Der Spieler klickt in der Spiel-Übersicht auf den Button "Rangliste"</li> </ol>

	2. Der Spieler sieht eine Liste der Gruppen mit aktuellem Budget und aktuellen Function-Points
Erweiterungen	
Abhängigkeiten	-
User Stories	US6

ID	UC9
Name	Runde beenden ankündigen
Ziel im Kontext	Der Spielleiter kündig an per wann eine Runde beendet sein wird, alle Gruppen mit fehlenden Eingaben erhalten eine Nachricht.
Akteure	Spielleiter
Trigger	
Voraussetzungen	a. Es ist eine Runde aktiv
Essentielle Schritte	<ol style="list-style-type: none"> <li>1. Der Spielleiter klickt auf der Administratoren-Oberfläche auf ein aktives Spiel mit aktuell laufender Runde.</li> <li>2. Der Spielleiter klickt auf den Button "Ende der Runde ankündigen"</li> <li>3. Der Spielleiter trägt das Datum und die Uhrzeit des geplanten Endes der Runde ein.</li> <li>4. Das System versendet an alle Gruppen, welche noch keine Eingaben getätigt haben, eine Nachricht.</li> </ol>
Erweiterungen	
Abhängigkeiten	-
User Stories	US7

ID	UC10
Name	Projektrisiko reduzieren
Ziel im Kontext	Der Spieler reduziert ein Risiko indem er Geld diesem Risiko zuweise.
Akteure	Spieler
Trigger	
Voraussetzungen	b. Es ist eine Runde aktiv
Essentielle Schritte	<ol style="list-style-type: none"> <li>1. Der Spieler navigiert im Spiel zu "Risiko Management"</li> <li>2. Der Spieler sieht alle verfügbaren Risiken</li> <li>3. Der Spieler wählt ein Risiko aus und weist diesem eine bestimmte Summe zu.</li> <li>4. Der Spieler speichert den Vorgang</li> </ol>
Erweiterungen	
Abhängigkeiten	-
User Stories	US8

ID	UC11
Name	CRUD Arbeiter
Ziel im Kontext	Der Spieler kann aus unterschiedlichen Arbeitern auswählen, welche er dem Projekt zuweisen möchte.
Akteure	Spieler
Trigger	

Voraussetzungen	c. Es ist eine Runde aktiv
Essentielle Schritte	<ol style="list-style-type: none"> <li>1. Der Spieler navigiert im Spiel zu "Projekt Mitarbeiter"</li> <li>2. Der Spieler sieht eine Liste mit den aktuell dem Projekt zugeteilten Mitarbeiter</li> <li>3. Der Spieler sieht einen Vorschlag eines Mitarbeiters pro Mitarbeiterkategorie mit dessen Werten <ol style="list-style-type: none"> <li>a. Kosten</li> <li>b. Effizient</li> </ol> </li> <li>4. Der Spieler kann den Mitarbeiter dem Projekt durch den "Einstellen" Button zuweisen.</li> </ol>
Erweiterungen	<p>3a.1 Der Spieler kann einen bestehenden Mitarbeiter feuern indem er auf den "Löschen" Button klickt.</p> <p>-----</p> <p>4a.1 Der Spieler kann den Mitarbeiter abweisen, worauf ein neuer Mitarbeiter angezeigt wird.</p>
Abhängigkeiten	-
User Stories	US9
Kommentar	Der Spielleiter definiert in den Stammdaten der Applikation Mitarbeitertypen mit Grundwerten für Kosten und Effizienz. Anhand dieser Grunddaten wird dem Spieler ein Mitarbeiter zur Auswahl gestellt, dessen Werte mit einer bestimmten Varianz variieren.

ID	UC12
Name	Risiko eintreten lassen
Ziel im Kontext	Der Spielleiter kann ein Risiko eintreten lassen, welches sich auf die Gruppen auswirkt
Akteure	Spielleiter
Trigger	
Voraussetzungen	d. Es ist eine Runde aktiv
Essentielle Schritte	<ol style="list-style-type: none"> <li>1. Der Spielleiter wählt auf der Administratoren-Oberfläche ein aktives Spiel mit aktiviert Runde aus.</li> <li>2. Der Spielleiter klickt auf den "Risiko auslösen" Button</li> <li>3. Der Spielleiter wählt ein verfügbares Risiko aus</li> <li>4. Der Spielleiter sieht von allen Gruppen wie viel in die Risikominimierung dieses Risikos investiert wurde.</li> <li>5. Der Spielleiter definiert welchen Schaden "Negative Function Points" oder "Kosten" entstehen</li> <li>6. Der Spielleiter schreibt einen kurzen Text zur Ursache des eingetretenen Risikos</li> <li>7.</li> </ol>
Erweiterungen	
Abhängigkeiten	UC6
User Stories	US4

ID	UC13
Name	E-Mail-Adresse ändern
Ziel im Kontext	Der Spieler kann seine eigene E-Mail-Adresse ändern.
Akteure	Spieler, System
Trigger	
Essentielle Schritte	<ol style="list-style-type: none"> <li>1. Der Spieler navigiert zu den Einstellungen</li> </ol>

	<ol style="list-style-type: none"> <li>2. Der Spieler setzt eine neue E-Mail-Adresse und klickt auf den "Speichern" Button</li> <li>3. Das System speichert nun die neue E-Mail-Adresse temporär und sendet eine Bestätigung mit einem Bestätigungslink an die neue E-Mail-Adresse</li> </ol>
Erweiterungen	<ol style="list-style-type: none"> <li>3a.1 Die E-Mail-Adresse existiert bereits.</li> <li>3a.2 Das System gibt eine Fehlermeldung aus.</li> </ol>
Abhängigkeiten	-
User Stories	US10

ID	UC14
Name	E-Mail-Adresse bestätigen
Ziel im Kontext	Das System erhält einen Token, um eine E-Mail-Adresse zu bestätigen
Akteure	System
Trigger	UC13
Essentielle Schritte	<ol style="list-style-type: none"> <li>1. Das System erhält einen Token (z.B. von einem angeklickten Bestätigungslink)</li> <li>2. Ist der Token valide, setzt das System vom dazugehörigen User die temporäre E-Mail-Adresse als E-Mail-Adresse</li> </ol>
Erweiterungen	<ol style="list-style-type: none"> <li>2a.1 Der Token ist nicht valide.</li> <li>2a.2 Das System gibt eine Fehlermeldung aus.</li> </ol>
Abhängigkeiten	UC13
User Stories	US10

ID	UC15
Name	Alle User einsehen
Ziel im Kontext	Der Spielleiter kann alle registrierten Benutzer einsehen.
Akteure	Spielleiter
Trigger	
Essentielle Schritte	<ol style="list-style-type: none"> <li>1. Der Spielleiter klickt auf der Administratoren-Oberfläche auf "Benutzer-Verwaltung"</li> <li>2. Der Spielleiter sieht eine Liste mit allen Benutzern, welche folgende Daten enthält <ol style="list-style-type: none"> <li>a. Name</li> <li>b. Aktives Spiel</li> <li>c. Registrations-Datum</li> <li>d. Letztes Login</li> </ol> </li> </ol>
Erweiterungen	
Abhängigkeiten	-
User Stories	US13

ID	UC16
Name	User erstellen
Ziel im Kontext	Der Spielleiter legt einen neuen Spieler an
Akteure	Spielleiter
Trigger	
Essentielle Schritte	<ol style="list-style-type: none"> <li>1. Der Spielleiter klickt auf der Administratoren-Oberfläche auf "Benutzer-Verwaltung"</li> </ol>

	<ol style="list-style-type: none"> <li>2. Der Spielleiter klickt auf Benutzer hinzufügen.</li> <li>3. Der Spielleiter trägt Name und E-Mail-Adresse des Spielers ein.</li> <li>4. Der Spielleiter speichert die Eingaben.</li> </ol>
Erweiterungen	<ol style="list-style-type: none"> <li>3a.1 Die E-Mail-Adresse ist bereits vorhanden</li> <li>3a.2 Das System gibt eine Fehlermeldung aus</li> </ol>
Abhängigkeiten	-
User Stories	US10

ID	UC17
Name	Spiel per Token beitreten
Ziel im Kontext	Der Spieler tritt einem Spiel mittels Tokens bei
Akteure	Spieler
Trigger	
Voraussetzungen	<ol style="list-style-type: none"> <li>e. Der Spieler ist noch keinem Spiel beigetreten</li> <li>f. Das Spiel wurde noch nicht gestartet</li> <li>g. Der Spieler ist im System angemeldet</li> </ol>
Essentielle Schritte	<ol style="list-style-type: none"> <li>1. Der Spieler trägt den Token im Textfeld ein</li> <li>2. Der Spieler bestätigt mit dem Button "Spiel beitreten"</li> </ol>
Erweiterungen	<ol style="list-style-type: none"> <li>2a.1 Der Token ist ungültig</li> <li>2a.2 Das System gibt eine Fehlermeldung aus</li> </ol>
Abhängigkeiten	-
User Stories	US13
Kommentar	<p>Der Token wird dem Spielleiter bei der Erstellung eines Spiels ausgegeben und kann anschliessend den Studenten per E-Mail oder mündlich mitgeteilt werden.</p> <p>Diese Funktion würde die offene Spielauswahl ergänzen oder ersetzen.</p>

## 5.2. Erweiterungsmöglichkeiten

Im Rahmen der Live Tests im PmQm Unterricht haben wir weitere Erweiterungsmöglichkeiten festgestellt, welche in einer eventuellen dritten Version umgesetzt werden könnten. Nachfolgend eine Liste mit den Überlegungen. Diese Überlegungen können in einem nächsten Schritt in User Storys umgeschrieben werden und in den Product Backlog aufgenommen werden.

Core Features:

- Die Mitarbeiter, welche in einer Runde eingestellt wurden, für die nächste Runde übernehmen und nur auf Wunsch der Spieler oder durch Verkleinerung der vom Spielleiter zur Verfügung gestellten Stellenprozente eine Änderung an der Personalbesetzung vornehmen.
- Die Ausbildung hat nur eine Auswirkung auf genau diese Mitarbeiter, welche in der aktuellen Runde auch eingestellt waren.
- Eine beschränkte Anzahl an Mitarbeiter pro Mitarbeitertyp, welche pro Runde eingestellt werden können, zur Verfügung stellen. Besser ausgebildete Mitarbeiter sind schwieriger zu finden.
- Die Zuteilung der Aufgaben des Projektleiters haben eine Auswirkung. Beispielsweise, wenn der Projektleiter kein Aufwand in Teambildung investiert, verlieren die Mitarbeiter an Motivation und somit auch an Produktivität.
- Inkludiert im gesamten Scope müssen spezifische Arbeitspakete abgearbeitet werden und gewisse Arbeitspakete können nur von spezifischen Mitarbeitertypen erledigt werden. Bei der Mitarbeiterzuteilung kann man dann die Arbeitspakete an Mitarbeiter zuweisen.
- Eine komplette Mandantenfähigkeit. Mit mehreren Mandanten könnte die komplette Umgegangen sauber getrennt von mehreren Kunden benutzt werden. Beispielsweise ein Mandant für die OST und ein Mandant für externe Kunden.
- Der Projektleiter hat einen Einfluss auf die Zufriedenheit der Mitarbeiter. Diese Zufriedenheit wirkt sich auf die Qualität der geleisteten Arbeit aus.

## 5.3. Domain Model

Während der Anforderungsanalyse haben wir ein neues Domain Model erstellt. Wir haben darauf geachtet, dass wir das Domain Model der vorhergehenden Studienarbeit soweit wie möglich übernehmen konnten. Damit alle zusätzlichen Use Cases und nicht-funktionale Anforderungen aber umgesetzt werden konnten, mussten wir gewisse Erweiterung am Domain Model vornehmen.

Um eine sehr hohe Komplexität des Domain Models durch die neu hinzukommenden Nachrichten Funktionalität zu verhindern, haben wir uns dazu entschieden, dass wir

Nachrichten an mehrere Empfänger auch mehrfach speichern. So ersparen wir uns komplexe Zwischentabellen, welche die Abarbeitung des „nicht gelesen“ Flags zur selben Nachricht von mehreren unterschiedlichen Benutzern ermöglicht. Die Menge an Daten, welche durch eine doppelte Abspeicherung der Text Nachricht entsteht hat unserer Meinung nach keinen negativen Einfluss auf die Performance der Datenbank.

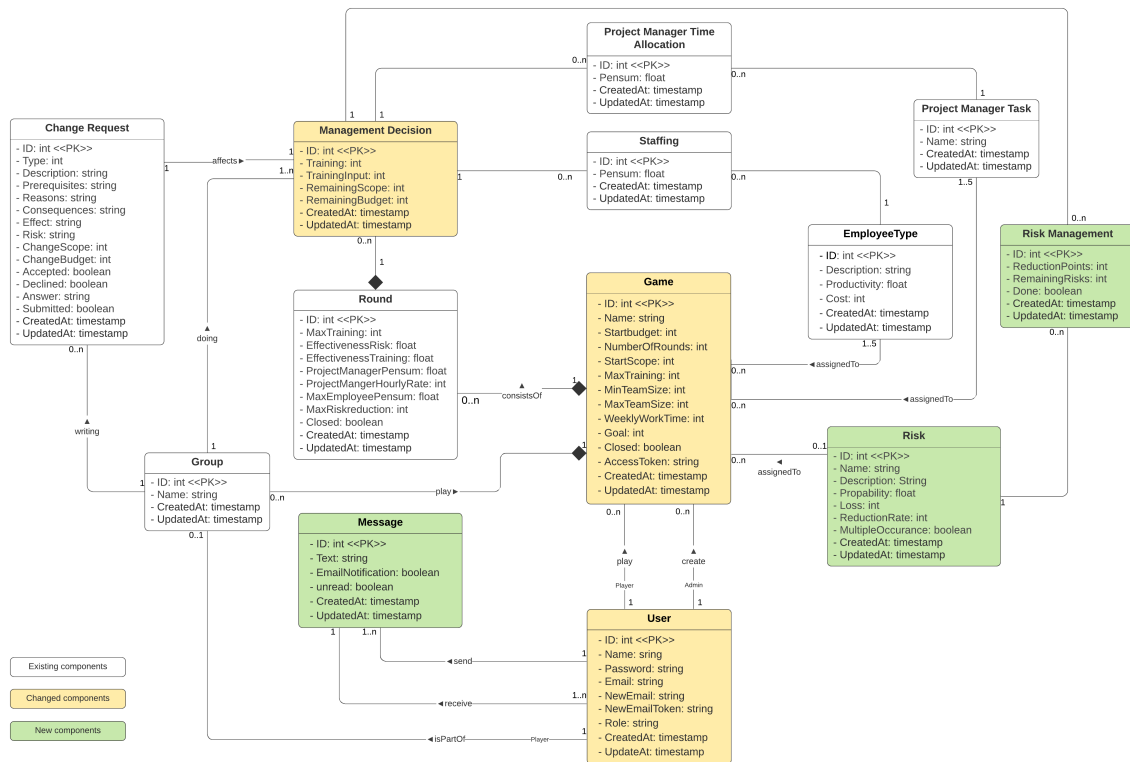


Abbildung 23: Domain Model

Da wir nach dem Wasserfall Modell gearbeitet haben, legten wir in der Anforderungsanalyse viel Wert auf ein komplettes Domain Model, damit es während der Umsetzung keine Änderungen mehr gibt.

Die einzige Änderung, die wir im Nachhinein vorgenommen haben, ist dass wir die Verbindungen zwischen „User“ und „Risk“ sowie „User“ und „EmployeeType“ entfernt haben. Diese Verbindungen hatten keinen Einfluss auf die Software, da es nicht relevant ist, welcher User die Risiken und Mitarbeitertypen erstellt hat.

## 5.4. Nicht-funktionale Anforderungen

In diesem Abschnitt gehen wir genauer auf die nicht-funktionalen Anforderungen ein.

Die in der vorhergehenden Studienarbeit erstellten nicht-funktionalen Anforderungen haben wir als Basis übernommen und anschliessend mit nicht-funktionalen Anforderungen

für die neuen Funktionalitäten ergänzt, welche erst in der Bachelorarbeit hinzugekommen sind. Beispiele dafür sind die Anforderungen an die Modularität, welche verlangt, dass neu das Frontend vom Backend des Systems abgekoppelt ist oder die Performance Anforderungen für das Versenden von E-Mails aus der Applikation. Zudem haben wir vereinzelt auch Verschärfungen der bestehenden Anforderungen vorgenommen, wie z. B. dass neu bei Fehlereingaben der User nicht nur entsprechend die Art der Eingabe validiert sondern auch eine passende Fehlermeldung ausgegeben wird. Auch Bereiche wie Reliability wurden verschärft, so dass zum Beispiel während den Live Tests die Applikation 100 % verfügbar sein soll.

#### **5.4.1. Zu unterstützende Szenarien**

Wie auch bei der Studienarbeit werden wir uns zur Überprüfung der nicht-funktionalen Anforderungen auf die zwei bereits definierten Szenarios stützen. Dies hat zum Vorteil, dass die bei der Studienarbeit umgesetzten Szenarien sicherlich weiter unterstützt werden und bietet auch einen Vergleich der aktuellen zur Vorgängerversion.

**Szenario 1:** Workshop: in 1 Übungslektion mit 1 Übungsgruppe (1 Spielleiter und 18 Teilnehmer, die 6 3er-Gruppen bilden) ein ganzes Spiel mit 10 Runden durchspielen.

**Szenario 2:** über den Lauf eines ganzen Semesters mit 1 Spielleiter und 126 Teilnehmern, die 32 Gruppen bilden (30 4er-Gruppen und 2 3er-Gruppen), 1 Spiel mit 14 Runden spielen, wobei pro Woche 1 Runde gespielt wird (Eingabe bis Freitag- abend, Spielleiter beendet Runde am Samstag, Resultate am Montag verfügbar)

Für beide Szenarien gilt: 5 Aufwände und 1 Event pro Spiel/Runde.

Zielgruppen:

**Spielleiter** Eine Lehrkraft mit Ausbildung im Bereich „Projektmanagement“ (oder vergleichbare Kompetenzen) mit mindestens rudimentären EDV-Anwenderkenntnisse.

**Spieler** Haben mindestens rudimentäre EDV-Anwenderkenntnisse.

## 5.4.2. Nicht-funktionale Anforderungen

Nachfolgend die von uns während der Anforderungsanalyse spezifizierten Qualitätsattribute (Anforderungskategorien gemäss ISO/IEC 25010):

Merkmal	Beschreibung
<b>Functional Stability</b>	<p><i>Functional Completeness:</i></p> <ul style="list-style-type: none"> <li>Alle User-Stories, die mit Priorität "MUSS" gekennzeichnet sind, sind implementiert.</li> </ul> <p><i>Functional Correctness:</i></p> <ul style="list-style-type: none"> <li>Das Budget, der Scope und die Restrisiken einer Runde eines Teams werden basierend auf den Stellschrauben (Aufwände, Events, Wirksamkeiten, Risiken etc.) korrekt berechnet.</li> <li>Die Rangliste am Ende des Spiels zeigt die korrekte Reihenfolge der Teams basierend auf dem Projektziel des Spiels.</li> </ul>
<b>Performance efficiency</b>	<p><i>Time behaviour:</i></p> <p>Alle nachfolgenden Anforderungen gehen von einer Internetleitung eines Schweizer Standardhaushalt (SH) mit 30MBit/s aus:</p> <ul style="list-style-type: none"> <li>Antwortzeiten auf alle "CRUD Spielleiter"-Aktionen betragen maximal 1 Minute (Szenario 1 und Szenario 2) (bei 1 User im System online)</li> <li>Nachdem der Spielleiter ein neues Spiel bestätigt hat, beträgt die Antwortzeit maximal 1 Minute (Szenario 1 und Szenario 2) (bei 1 User im System online)</li> <li>Nachdem der Spielleiter oder ein Spieler eine Nachricht als E-Mail verfasst hat, wird das E-Mail innert 40 Sekunden (Szenario 1; bei 19 User im System online) respektive 240 Sekunden (Szenario 2; bei 127 User im System online) verschickt. Die E-Mails werden gestaffelt, maximal 40 Mails pro Minute, versendet, um Blacklisting des Servers zu vermeiden.</li> <li>Nachdem der Spielleiter oder ein Spieler eine in-game Nachricht an einen anderen Spieler sendet, wird diese beim Empfänger innerhalb von 10 Sekunden angezeigt.</li> <li>Nachdem ein Benutzer eine Gruppe erstellt oder verändert, wird die Änderung nach maximal 20 Sekunden (Szenario 1; bei 19 User im System online) respektive 30 Sekunden (Szenario 2; bei 127 User im System online) bei allen anderen Benutzern angezeigt.</li> <li>Nachdem der Spielleiter das <b>Spiel</b> gestartet hat, beträgt die Antwortzeit maximal 30 Sekunden (Szenario 1; bei 19 User im System online) respektive 1 Minuten (Szenario 2; bei 127 User im System online)</li> <li>Nachdem der Spielleiter eine <b>Runde</b> gestartet hat, beträgt die Antwortzeit maximal 3 Sekunden (Szenario 1; bei 19 User im System online) respektive 10 Sekunden (Szenario 2; bei 127 User im System online)s</li> <li>Nachdem der Spieler einer Gruppe die Eingaben für eine Runde bestätigt hat, beträgt die Antwortzeit maximal 3 Sekunden (Szenario 1; bei 19 User im System online) respektive 10 Sekunden (Szenario 2; bei 127 User im System online)</li> <li>Nachdem der Spielleiter eine Runde beendet hat, dauert das Berechnen der Rundenresultate maximal 1 Minute (Szenario 1; bei 19 User im System online) respektive 3 Minuten (Szenario 2; bei 127 User im System online)</li> </ul> <p><i>Capacity:</i></p> <ul style="list-style-type: none"> <li>Es können mindestens 126 Spieler (maximale Teilnehmerzahl des HSR-Moduls "Projekt- und Qualitätsmanagement") und 1 Spielleiter gleichzeitig im System eingeloggt sein.</li> </ul>
<b>Compatibility</b>	<p><i>Installability:</i></p> <ul style="list-style-type: none"> <li>Die Server-Software kann auf jedem Server ausgeführt werden welcher NodeJS unterstützt.</li> </ul> <p><i>Interoperability:</i></p> <ul style="list-style-type: none"> <li>Das System versendet valide E-Mails gem. gültigen RFC Standards, welche von den SMTP Servern der Empfänger entgegengenommen werden.</li> </ul>
<b>Usability</b>	<i>Appropriateness recognizability:</i>

	<ul style="list-style-type: none"> <li>• Für noch nicht registrierte Benutzer ist eine Info Seite mit einem Beschrieb sowie mehreren Printscreens verfügbar.</li> </ul> <p><i>Learnability:</i></p> <ul style="list-style-type: none"> <li>• Das GUI ist von der Zielgruppe ohne besondere Schulung verwendbar.</li> <li>• Die Applikation verfügt über ein Benutzerhandbuch.</li> </ul> <p><i>Operability:</i></p> <ul style="list-style-type: none"> <li>• Es ist kein vertikales Scrollen nötig bei Zoom-Level 100% für die optimierten Geräte und Auflösungen (siehe "Look &amp; Feel")</li> <li>• Ein Spielleiter erfasst (maximal) 5 benutzerdefinierte Aufwände in unter 10 Eingabeaktionen (Startpunkt Adminübersicht).</li> <li>• Ein Spielleiter erfasst 1 benutzerdefiniertes Risiko in unter 5 Eingabeaktionen (Startpunkt Adminübersicht).</li> <li>• Ein Spielleiter lanciert 1 Spiel mit (maximal) 5, bereits vorgängig erstellten, benutzerdefinierten Aufwänden und maximal 10 Gruppen in unter 5 Eingabeaktionen (Gruppen sind bereits im Spiel registriert) (Startpunkt Adminübersicht).</li> <li>• Spielzugeingabe: Ein User hat Einsicht in die Eingaben seiner Gruppe aus den vorherigen Runden mit einer Eingabeaktion (Startpunkt Spielübersicht).</li> <li>• Spielübersicht: Ein User hat Einsicht in den Zwischenstand der anderen Teams mit einer Eingabeaktion.</li> <li>• Spielübersicht: Spielzüge aus vorherigen Runden und der Zwischenstand der Teams sind zusammen auf einem Screen einsehbar für einen User.</li> </ul> <p><i>User error protection:</i></p> <ul style="list-style-type: none"> <li>• Alle Eingabefelder werden entsprechend der Art der Eingabe, welche sie repräsentieren, validiert.</li> <li>• Falls ein Fehler auftritt, erhält der Benutzer eine passende Fehlermeldung.</li> <li>• Der Benutzer wird davor geschützt, dass ungespeicherte Eingaben, unbemerkt verloren gehen.</li> </ul> <p><i>User interface aesthetics ("Look &amp; Feel"):</i></p> <ul style="list-style-type: none"> <li>• Das Design des GUI ist durchgehend konsistent.</li> <li>• "Project Manager" wird für die folgenden Geräte und Auflösungen optimiert und getestet: <ul style="list-style-type: none"> <li>• Desktop: Auflösung minimal 1920 x 1080</li> </ul> </li> <li>• Für alle die optimierten Geräte und Auflösung sind folgende Punkte erfüllt: <ul style="list-style-type: none"> <li>• wirkt vertrauenswürdig (Grammatik, Rechtschreibung, Farben, keine verpixelten Bilder, keine blinkenden Elemente)</li> <li>• Controls sind intuitiv angeordnet</li> <li>• Zweck (Hauptfunktion) der Seite ist klar</li> </ul> </li> </ul>
<b>Reliability</b>	<p><i>Availability:</i></p> <ul style="list-style-type: none"> <li>• Während den Live Tests ist das System zu 100% verfügbar.</li> </ul> <p><i>Fault tolerance:</i></p> <ul style="list-style-type: none"> <li>• Die Server Software crash im normalen Betrieb nicht. Sämtliche absehbare Fehler werden abgefangen.</li> </ul> <p><i>Recoverability:</i></p> <ul style="list-style-type: none"> <li>• Täglich wird 1 Backup der Datenbank erstellt.</li> <li>• Wiederherstellung bei Systemausfall: Da wir nur 1 tägliches Backup machen, ist der aktuellste Stand vom Vortag wiederherstellbar. Änderungen des laufenden Tages sind verloren.</li> </ul>
<b>Security</b>	<p><i>Confidentiality:</i></p> <ul style="list-style-type: none"> <li>• Die Applikation speichert nur relevante Personendaten.</li> <li>• Die Authentifizierung erfolgt via E-Mail-Adresse und Passwort.</li> <li>• Die E-Mail-Adresse wird durch einen Token verifiziert.</li> <li>• Das Passwort ist in der Datenbank als salted Hash abgelegt.</li> </ul>

	<ul style="list-style-type: none"> <li>• Benutzerrechte: Die Spieler haben keinen Zugriff auf Bereiche, die nur der Spielleiter verwalten darf.</li> </ul> <p><i>Integrity:</i></p> <ul style="list-style-type: none"> <li>• XSS: Sämtliche Formulare sind gegen XSS-Angriffe abgesichert.</li> <li>• Konfigurationen: Die Konfigurationsdateien sind nicht öffentlich erreichbar.</li> <li>• Login: Ein Spieler hat nur Einsicht in jene Spieldaten, welche mit seinem Login verknüpft sind (Ausnahme: die Funktionalität wurde explizit definiert und implementiert, siehe funktionale Anforderungen).</li> <li>• Passwortstärke: Ein Passwort besteht mindestens aus 8 Zeichen</li> <li>• Passwortstärke: Ein Passwort besteht mindestens aus 1 Grossbuchstaben, 1 Kleinbuchstaben, 1 Zahl und 1 Sonderzeichen.</li> </ul> <p><i>Recovery:</i></p> <ul style="list-style-type: none"> <li>• Die Datenbank kann einfach gesichert werden, so dass die Daten bei einem Systemabsturz wiederhergestellt werden können.</li> </ul>
<b>Maintainability</b>	<p><i>Modularity:</i></p> <ul style="list-style-type: none"> <li>• Das Frontend und Backend sind getrennt.</li> </ul> <p><i>Analysability:</i></p> <ul style="list-style-type: none"> <li>• Fehler, welche auf dem Server auftreten, werden in einem separaten Error Log gesammelt.</li> <li>• Stack traces werden nicht bis ins Frontend weitergegeben.</li> </ul> <p><i>Modifiability:</i></p> <ul style="list-style-type: none"> <li>• Die CI/CD Pipeline verfügt über eine Infrastruktur, um mindestens die Stages "build", "test", "deploy" (entweder auf Staging- oder Production-Server) zusammen automatisch auszuführen.</li> <li>• Die CI/CD Pipeline verfügt über eine Infrastruktur, um automatisch von der Staging- auf die Production-Umgebung zu deployen.</li> <li>• Die Software wird in einem Container auf den Host deployed und gestartet.</li> <li>• Beim Betrieb der Applikation in der ausgelieferten Version fällt kein Wartungsaufwand an.</li> </ul> <p><i>Testability:</i></p> <ul style="list-style-type: none"> <li>• Das Main Success-Szenario aller implementierten Use Cases wird mittels Unit Tests überprüft.</li> <li>• Zu den Live Tests werden Testspezifikationen und Testprotokolle erstellt.</li> </ul>
<b>Portability</b>	<p><i>Adaptability:</i></p> <ul style="list-style-type: none"> <li>• Die Applikation ist anbieterunabhängig: Das heisst, dass die Applikation auf jedem Server lauffähig ist, auf dem Node.js ausgeführt werden kann.</li> </ul> <p><i>Installability:</i></p> <ul style="list-style-type: none"> <li>• Die Applikation ist mit weniger als 10 Kommandozeilenaktionen installierbar.</li> <li>• Die Applikation verfügt über ein Readme, indem die Installationschritte detailliert beschrieben werden.</li> </ul>

## 5.5. Logische Architektur

Im folgenden Schichten Diagramm (Abbildung 24) haben wir dargestellt, wie die einzelnen Teile der Software in unterschiedliche Layer aufgeteilt werden. Wichtig sind dabei vor allem die durch die Pfeile dargestellten Abhängigkeiten zwischen den Layer.

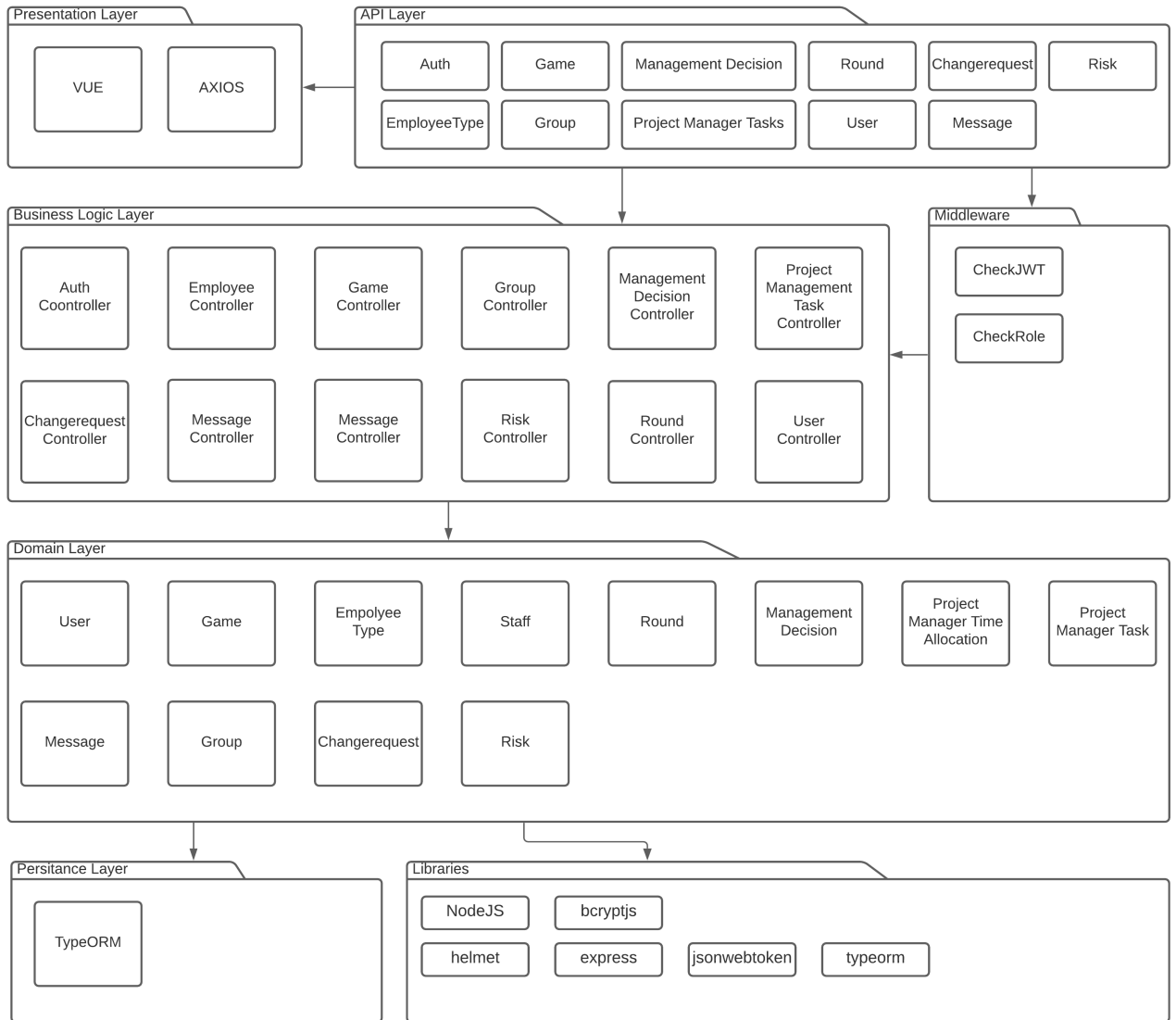


Abbildung 24: Schichten Diagramm

Wir haben uns dazu entschieden, die einzelnen Funktionen im Diagramm wegzulassen da dies die Übersichtlichkeit negativ beeinflusst hätte, ohne einen Mehrwert zu generieren.

## 5.6. Deployment-Diagramm

Für das Deployment der Applikation haben wir den bestehenden virtuellen Server, welcher für die Studienarbeit bereitgestellt wurde, übernommen und werden ihn entsprechend an das neue Setup angepasst.

Wie in Abbildung 25 zu sehen ist, haben wir neu die Frontend und Backend Komponenten der Applikation getrennt und je in einem eigenen Docker Container deployed. Das Frontend sowie auch die Backend API sind über den Traefik Reverse Proxy von außerhalb der Docker Umgebung über die Domain [www.pmgame.ch](http://www.pmgame.ch) erreichbar. Traefik ist auch für das bereitstellen der SSL Verbindungen verantwortlich. Ein MariaDB Container stellt die Datenbank zur Verfügung und persistiert so die Daten.

Für das Frontend und Backend gibt es auch je einen Staging Container, welche über die Subdomain [staging.pmgame.ch](http://staging.pmgame.ch) erreichbar sind. Das Staging Backend hat jedoch keine Verbindung zur Datenbank der produktiven Umgebung, diese benutzt eine dateibasierte SQLite Datenbank, welche direkt auf dem Staging Backend Container zur Verfügung gestellt wird.

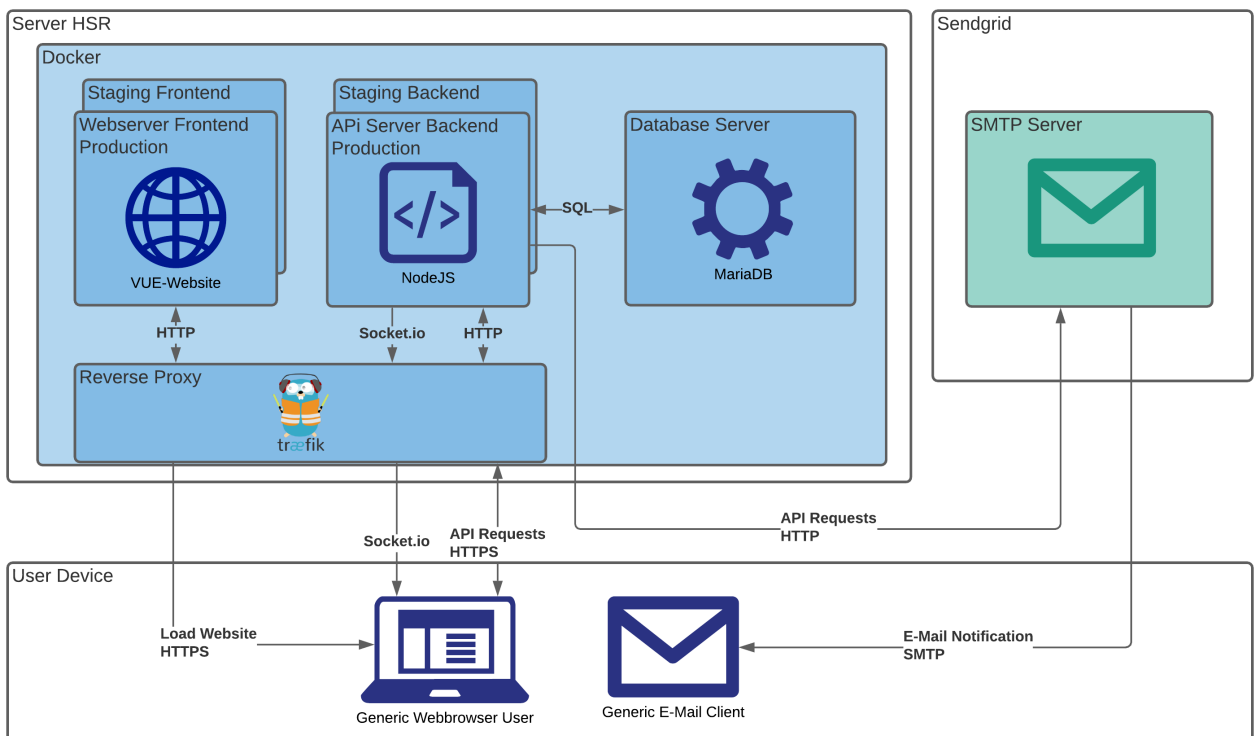


Abbildung 25: Deployment-Diagramm

Um E-Mails zu versenden benutzen wir den externen Service von SendGrind, welcher ein SMTP Server zur Verfügung stellt. Das Backend sendet API Requests mit den entspre-

chenden E-Mail Inhalten an SendGrid, wo diese als E-Mails an die jeweiligen Empfänger versendet werden.

### 5.7. GUI-Skizzen

Wir haben zur grafischen Umsetzung der Applikation diverse GUI-Skizzen angelegt.



Abbildung 26: Übersicht Game

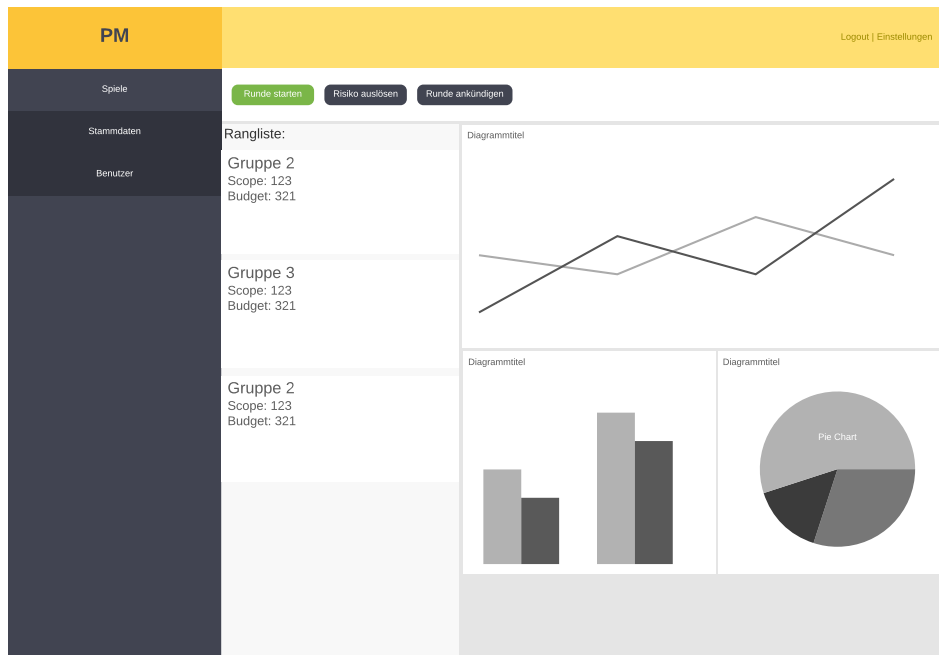


Abbildung 27: Übersicht Admin

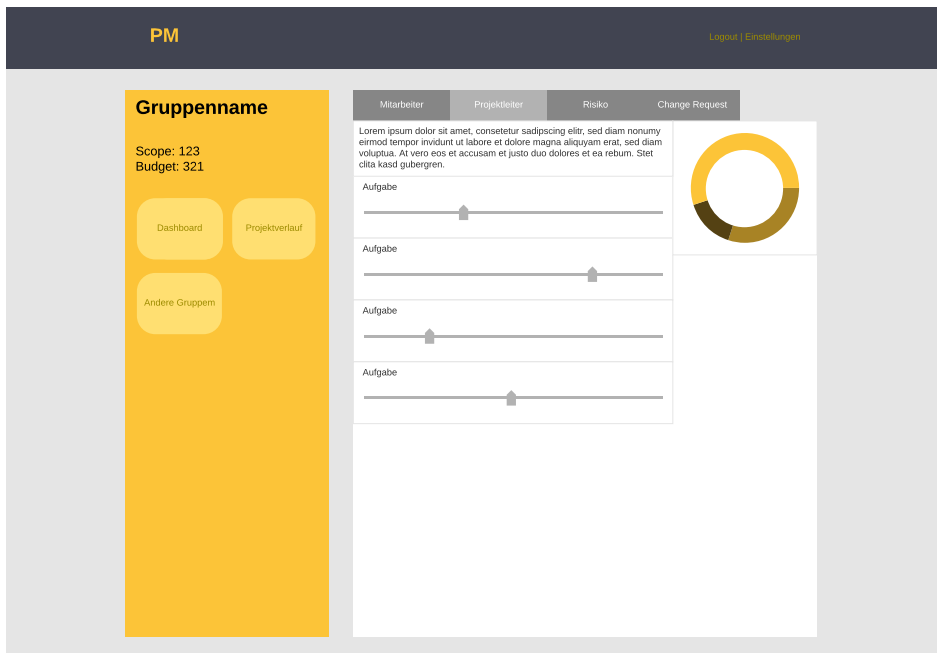


Abbildung 28: Spiel - Projektleiter Management

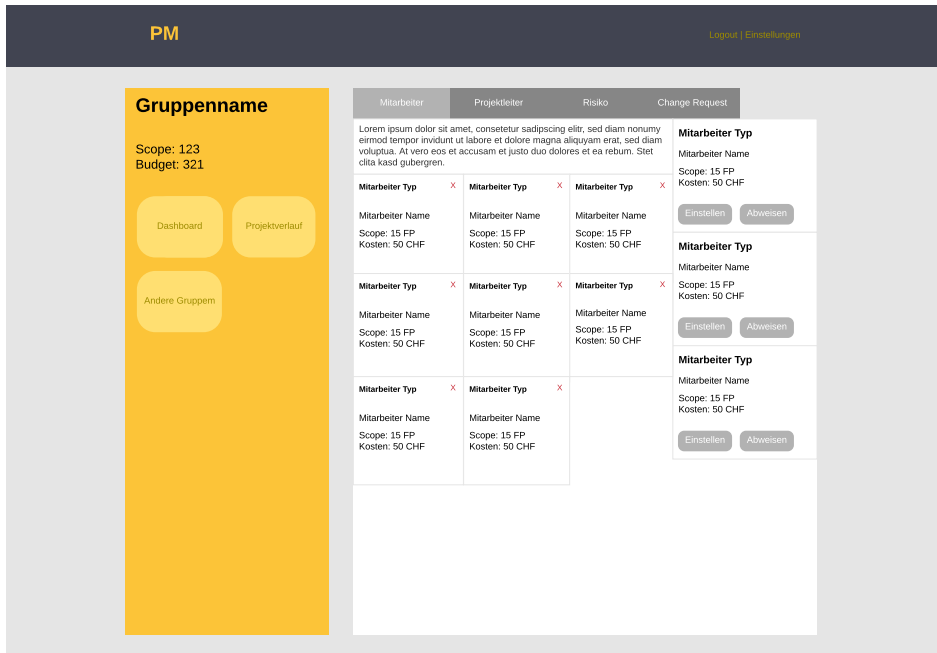


Abbildung 29: Spiel - Mitarbeiter Management

## 5.8. Architekturentscheidungen: Diskussion

Item ( <i>Anforderung</i> )	Beschreibung
TypeORM <sup>13</sup> <i>Portability</i>	Wir setzten TypeORM ein, um flexibel von einer SQLite-Datenbank auf eine MariaDB- oder MySQL-Datenbank wechseln zu können. So konnten wir lokal bei der Entwicklung ohne grossen Installationsaufwand mit einer SQLite-Datenbank beginnen. Auf der produktiven Umgebung hingegen konnten wir eine MariaDB Datenbank in einem separaten Dockercontainer verwenden, ohne am Code Anpassungen vorgenommen werden mussten. Im Gegensatz zu knexjs (verwendet in der Version der Studienarbeit ) ist TypeORM ein vollständiger OR Mapper. Das Schema der Datenbank wird dadurch automatisch erstellt und die Klassen können als Objekte verwendet werden. Zudem bringt die Library viele Standardoperationen (wie z.B. find, update, delete) für die einzelnen Objekte mit, bei Bedarf kann jedoch auch natives SQL verwendet werden.
Node.js <sup>14</sup> <i>Portability</i>	Durch die guten Erfahrungen mit Node.js in der Version der Studienarbeit haben wir uns dazu entschieden, Node.js weiter zu verwenden.
TypeScript <sup>15</sup> <i>Functional Stability,</i> <i>Reliability</i>	Wir wollen auch weiterhin eine typsicherere Applikation anbieten, deshalb werden wir auch weiterhin TypeScript einsetzen.
Trennung Frontend Backend <i>Maintainability</i>	Wir haben uns dazu entschieden, das Frontend und Backend voneinander zu trennen. In der Version der Studienarbeit wurde das Frontend und Backend noch von derselben Applikation bereitgestellt. Durch die Trennung gibt es eine schöne Abgrenzung von der Auslieferung der Webseite und der Logik der Applikation. Das Backend stellt eine API zur Verfügung, sodass der WebClient im Browser direkt die Daten abfragen kann. Dadurch wird die Wartbarkeit des Codes stark verbessert. Weiter können die Server für Frontend und Backend auch separat an die benötigten Ressourcen angepasst werden.

---

<sup>13</sup><https://typeorm.io/>

<sup>14</sup><https://nodejs.org/>

<sup>15</sup><https://www.typescriptlang.org/>

<p>MariaDB<sup>16</sup> als dedizierter Datenbank-Server <i>Performance</i></p>	<p>Durch das Einsetzen eines dedizierten MariaDB Servers kann die Stabilität und Zugriffsgeschwindigkeit gegenüber der SQLite Datenbank stark verbessert werden. Ebenfalls ist ein direkter Zugriff auf die Datenbank über die Konsole möglich ohne dass ein separates Tool nötig ist. Dies vereinfacht auch Wartungen und Sicherungen der Datenbank.</p>
<p>Socket.io<sup>17</sup> <i>Usability</i></p>	<p>Socket.io ermöglicht es, bei einer Änderung von Daten im Backend eine Benachrichtigung an alle Clients zu versenden, welche die entsprechenden Daten aktuell verwenden. Dadurch können die Clients die neuen Daten selbständig abrufen und dem Spieler die aktuellsten Daten anzeigen. So können beispielsweise Teammitglieder, welche an den selben Eingaben arbeiten, direkt die Inputs der anderen Spieler erhalten. Dadurch können doppelte Eingaben oder das Überschreiben der vorherigen Eingaben durch ein anderes Teammitglied verhindert werden.</p>
<p>Vue.js<sup>18</sup> <i>Usability</i></p>	<p>Wir hatten durch andere Module an der OST bereits Erfahrungen mit Vue.js gesammelt. Mit Vue.js können wesentlich interaktivere User Interfaces umgesetzt werden, was ein Vorteil gegenüber den serverseitig gerenderten Applikationen ist. Daher haben wir uns dazu entschieden, in diesem Projekt für das Frontend Vue.js einzusetzen.</p>
<p>Docker<sup>19</sup> <i>Reliability</i></p>	<p>Docker erlaubt das einfache Erstellen und Verwalten von Container. Dadurch ist es möglich ein effizientes Deployment aufzubauen. Da wir bereits beide Erfahrungen mit Docker gesammelt haben, war es für uns naheliegend auf Docker zu setzen.</p>

---

<sup>16</sup><https://mariadb.org/>

<sup>17</sup><https://socket.io/>

<sup>18</sup><https://vuejs.org/>

<sup>19</sup><https://docker.com/>

<p>GitLab-CI/CD-Pipeline<sup>20</sup> <i>Maintainability</i></p>	<p>Die CI/CD Pipeline wird in den folgenden 6 Schritten abgearbeitet:</p> <ul style="list-style-type: none"> <li>• <b>install dependencies</b> Installiert alle benötigten Software Pakete. Wird nur ausgeführt, wenn es Änderungen gab im package.json.</li> <li>• <b>build</b> Erstellt die ausführbare Software.</li> <li>• <b>assemble</b> Erstellt die Docker Images mit der Software.</li> <li>• <b>test</b> Führt alle Tests aus und stoppt den ganzen Prozess, falls Fehler auftreten sollte.</li> <li>• <b>deploy</b> Stellt die Dockercontainer mit den zuvor erstellten Images auf dem HSR Server bereit. Dieser Schritt muss manuell gestartet werden.</li> <li>• <b>data migration</b> Erstellt eine neue Datenbank. Wird ebenfalls manuell gestartet.</li> </ul> <p>Beim Pushen auf den Master Branch, wird das Deployment auf die Staging Umgebung gestartet. Damit auf die Produktionsumgebung bereitgestellt wird, muss in Gitlab ein neuer Release mit einem entsprechenden Tag erstellt werden. Dadurch wird automatisch eine neue Pipeline mit denselben Schritten für die Produktion gestartet. Bei der Entwicklung von neuen Features arbeiten wir jeweils mit eigenen Branches, welche durch Reviews des Anderen geprüft und anschliessend in den Master Branch gemerged werden.</p>
<p>bcrypt<sup>21</sup> NodeJS <i>Security</i></p>	<p>Wir haben die in NodeJS verfügbare bcrypt Library benutzt, um das Passwort in Form eines Salted-Hash in der Datenbank zu speichern.</p>

<sup>20</sup><https://docs.gitlab.com/ee/ci/>

<sup>21</sup><https://www.npmjs.com/package/@types/bcrypt>

## 5.9. Testing

### 5.9.1. Testing der funktionalen Anforderungen

**5.9.1.1. Black-Box-Testing** Die Abbildung 30 zeigt die Ergebnisse des Black-Box Tests. Im Black-Box Test haben wir unsere Applikation in einer aktuellen Version von Frontend und Backend genommen und gegen die Use Case Beschreibungen getestet, welche in Unterunterabschnitt 5.1.2 beschrieben und ersichtlich sind.

ID	Name	Priorität	Resultat	Bemerkung
UC1	Nachricht erstellen	2	Erfüllt	Nur der Administrator hat die Möglichkeit Nachrichten auch als E-Mail zu versenden. Aus dem Grund, dass wir die MX Records vor Spamlisten Einträge schützen wollen, wenn zu viel Mails versendet werden.
UC2	Nachricht versenden	2	Erfüllt	Der Admin kann an alle Spieler senden. Die Spieler nur an andere Spieler aus der selben Gruppe oder an die Admins.
UC3	Nachricht erhalten	2	Erfüllt	
UC4	Nachricht beantworten	1	Erfüllt	
UC5	Performance Report exportieren	3	Erfüllt	Alle Diagramme welche in der Admin Ansicht angezeigt werden, können als Grafik oder CSV heruntergeladen werden.
UC6	CRUD Risiko	1	Erfüllt	
UC7	Spiel anlegen	1	Erfüllt	
UC8	Überblick über andere Gruppen erhalten	2	Erfüllt	Spieler sehen einen Überblick über den Fortschritt jeder Gruppe. Admin kann die Projektreports aller Gruppen einsehen um einen Detaillierten Überblick über alle Eingaben zu erhalten.
UC9	Runde beenden ankündigen	2	Erfüllt	
UC10	Projektrisiko reduzieren	1	Erfüllt	
UC11	CRUD Arbeiter	3	Erfüllt	
UC12	Risiko eintreten lassen	1	Erfüllt	
UC13	E-Mail Adresse ändern	3	Erfüllt	
UC14	E-Mail Adresse bestätigen	3	Erfüllt	
UC15	Alle User einsehen	3	Erfüllt	
UC16	User erstellen	2	Erfüllt	
UC17	Spiel per Token beitreten	3	Zurückgestellt	Wurde durch einen Change Request zurückgestellt, damit wir mehr Zeit haben die aus den Livetest identifizierten Bugs und Verbesserungen umzusetzen.

Getestet am 17.12.2020

Version: Backend commit: 645d533e241b9ceaaf8f7938c6985bdb2f8e0a5, Frontend commit: 6c2db9b24c15e97d0212361e5280b0979156efd2

Tester: Dario Caluzi

Abbildung 30: Black-Box Test Ergebnisse

**5.9.1.2. White-Box-Testing** Das White-Box Testing bestand aus zwei Komponenten, den Unit Tests und Integrationstests.

**Unit Test** Die Unit Tests laufen vollkommen automatisiert ab und werden bei jedem Push auf das Git Repository in der CI/CD Pipeline ausgeführt. Sie testen die Programmlogik ausführlich und stellen sicher, dass bei Änderungen an den Funktionen die erwartete Ausgabe erfolgt. Die Tests helfen ebenfalls, nachfolgenden Entwicklern sicherzustellen, dass die Erweiterung der Software in bestehenden Teilen keine Fehler verursacht.

Die Test Coverage ist dabei sehr wichtig, denn sie zeigt auf, wie viele der Codezeilen durch die Tests tatsächlich abgedeckt sind. Wir wollten möglichst viele Fehlerfälle abde-

cken und haben daher im Projektplan definiert, dass wir mindestens eine Coverage von 65 % erreichen wollen. Nachfolgend in Abbildung 31, Abbildung 32, Abbildung 33 ist zu sehen welche Coverage wir tatsächlich erreicht haben.

**All files**

84.46% Statements 1543/1827 77.87% Branches 197/253 61.3% Functions 141/230 85.73% Lines 1466/1718

Press *n* or *j* to go to the next uncovered block, *b*, *p* or *k* for the previous block.

File	Statements	Branches	Functions	Lines
src	100%	4/4	100%	4/4
src/config	100%	1/1	100%	1/1
src/controllers	85.17%	1246/1463	73.95%	1196/1403
src/entity	78.29%	238/304	100%	212/248
src/middlewares	98.18%	54/55	100%	53/54

Abbildung 31: Coverage aller Files

Mit einer gesamten Coverage von 84.46 % haben wir unser Ziel fast um 20 % übertroffen.

Obwohl der Wert der Coverage für die *src/entity* grundsätzlich die Anforderungen von mindestens 65% erfüllt ist diese Coverage zu vernachlässigen, da in den Entitäten keine Logik enthalten ist. Es wurden keine expliziten Tests für die Entitäten geschrieben. Sie werden jedoch bei den Tests der Controller auch mit durchlaufen, wenn mit Daten manipuliert wird.

**All files src/middlewares**

98.18% Statements 54/55 100% Branches 38/38 100% Functions 2/2 98.15% Lines 53/54

Press *n* or *j* to go to the next uncovered block, *b*, *p* or *k* for the previous block.

File	Statements	Branches	Functions	Lines
checkAccess.ts	98.18%	54/55	100%	53/54

Abbildung 32: Coverage der Middleware

In der Middleware (Abbildung 32) ist nur die Access Controll vorhanden, welche mit einem Wert von 98.8 % sehr gut abgedeckt ist, was sehr wichtig ist.

### All files src/controllers

85.17% Statements 1246/1463 73.95% Branches 159/215 85.63% Functions 137/168 85.25% Lines 1196/1483

Press n or j to go to the next uncovered block, b, p or k for the previous block.

File	Statements	Branches	Functions	Lines
AuthController.ts	83.04%	142/171	84.38%	27/32
ChangeRequestController.ts	100%	188/188	91.43%	32/35
EmailController.ts	56.9%	33/58	33.33%	2/6
EmployeeController.ts	100%	27/27	100%	2/2
GameController.ts	81.92%	145/177	40.91%	9/22
GroupController.ts	87.04%	94/108	100%	4/4
ManagementDecisionController.ts	78.62%	114/145	76.92%	20/26
MessageController.ts	87.01%	67/77	62.5%	5/8
ProjectManagementTaskController.ts	100%	23/23	100%	2/2
RiskController.ts	92.86%	26/28	100%	2/2
RiskManagementController.ts	100%	30/30	100%	4/4
RoundController.ts	81.34%	170/209	68.75%	33/48
UserController.ts	84.23%	187/222	70.83%	17/24

Abbildung 33: Coverage aller Controllerns

Der grösste Teil der Logik ist in den Controllern vorhanden. Dem entsprechend sehen wir in Abbildung 33 die Coverage aufgebrochen auf die jeweiligen Controller. Der einzige Controller, welcher den gesetzten Wert von 65 % nicht erreicht, ist der *Email-Controller.ts*. Dort fehlen Tests für zwei der E-Mail Versandfunktionen. Die betreffenden Funktionen sind jedoch nicht kritisch und können gut zu einem späteren Zeitpunkt noch durch Tests abgedeckt werden.

Alle anderen Controller liegen mit der Test Coverage zwischen 78 % und 100 %. Dies ist eine sehr gute Grundlage für weitere Arbeiten an der Applikation.

**Integrationstest** Die Integrationstests waren Bestandteil unseres Review Prozesses und wurden manuell bei jedem Merge Request durchgeführt. So konnten wir immer das Zusammenspielen zwischen Frontend, Backend, Datenbank sowie des SendGrid Mailserver ausführlich testen. Wir konnten so vermeiden, dass in der Staging und Produktiv Umgebung Code deployed wurde, welcher zu Problemen mit den jeweiligen Komponenten geführt hätte.

Aus zeitlichen Gründen haben wir darauf verzichtet, für jeden Durchgang ein Testprotokoll zu verfassen. Gefundene Probleme wurden direkt im Gitlab Merge Request als Diskussion erfasst und an den Ersteller zur Behebung zurückgegeben. Nachdem der Ersteller die Änderungen in den Merge Request gepusht hat, wurden erneute Integration Tests durchgeführt, um die korrekte Funktionsweise sicherzustellen. Erst nachdem die neuen Funktionen umfassend getestet und für gut befunden wurde, erfolgte der Merge auf den Master Branch mit nachfolgendem Deployment.

## 5.9.2. Testing der nicht-funktionalen Anforderungen

Dieses Kapitel testet die in Unterunterabschnitt 5.4.1 aufgestellten nicht-funktionalen Anforderungen und ob diese erreicht wurden.

Kategorie	Subkategorie	Resultat	Bemerkung
Functional Stability	Functional Completeness	erfüllt	
	Functional Correctness	erfüllt	Die Berechnung des Spiels wurden vom Projektbetreuer kontrolliert und abgenommen.
Performance efficiency	Time behaviour	erfüllt	Während den Live Tests wurden die Zeitwerte für das Szenario 1 (mit 19 Usern online) nie überschritten.
	Capacity	erfüllt	Bei den Live Tests waren maximal 18 User simultan online, deshalb konnte die Kapazität für das Szenario 1 so getestet werden. Für das Szenario 2 haben wir mit Gatling
Compatibility	Installability	erfüllt	
	Interoperability	erfüllt	SendGrid versendet E-Mails im unter der pmgame.ch Domain welche Validiert ist.
Usability	Appropriateness recognizability	nicht erfüllt	Es ist keine Infoseite verfügbar.
	Learnability	erfüllt	
	Operability	erfüllt	Basierend auf dem Feedback der letzten Live Tests.
	User error protection	teilweise erfüllt	Im Change Request Management muss der User selbständig auf Speichern drucken. Überall sonst wird automatisch gespeichert.
	User interface aesthetics ("Look & Feel")	erfüllt	Basierend auf dem Feedback der letzten Live Tests.
Reliability	Availability	erfüllt	War 100% verfügbar.
	Fault tolerance	erfüllt	Es gab keinen Crash bis zu diesem Zeitpunkt.
	Recoverability	erfüllt	Backup Script erstellt automatische tägliche Sicherungen welche einfach wiederhergestellt werden können.
Security	Confidentiality	erfüllt	
	Integrity	erfüllt	
	Recovery	erfüllt	
Maintainability	Modularity	erfüllt	
	Analysability	erfüllt	Es werden keine Fehlermeldungen vom Backend ans Frontend weitergegeben. Es gibt ein separates Error Log der Applikation.
	Modifiability	erfüllt	
	Testability	teilweise erfüllt	Von den Live Tests wurden keine Testprotokolle erstellt sonder direkt Gitlab Issues welche danach abgearbeitet wurden.
Portability	Adaptability	erfüllt	
	Installability	erfüllt	

Abbildung 34: Ergebnisse der Tests zu den nicht-funktionalen Anforderungen

## 5.9.3. Live Tests

Wir hatten drei Mal die Möglichkeit Live Tests durchzuführen. Die Live Tests erfolgten in den Übungslektionen des PmQm Unterricht, wobei die Applikation jeweils als Spiel eingesetzt wurde. Der Dozent fungierte als Spielleiter und hat drei Spiele für die drei Übungsgruppen vorbereitet. Diese Spiele wurden in den drei Live Tests jeweils am Donnerstag Nachmittag mit den Studenten komplett durchgespielt.

Für uns stellten die Tests eine Gelegenheit dar, die Applikation unter Live-Bedingungen im Betrieb zu sehen und Fehler sowie Verbesserungsmöglichkeiten zu finden. Auch konnten wir viele hilfreiche Inputs der Studenten aufnehmen, deren Umsetzung die Applikation verbessert hat.

Wir haben alle Rückmeldungen, welche wir erhalten haben, direkt als Issues im GitLab erfasst und in der Phase Testing bearbeitet.

Issue ID	Title
66	Admin soll Gruppe löschen können
67	Risiko Bubble Diagramm max. Wert auf x-Achse
71	Projektleiterzeit benötigt in der ersten Runde lange um die Diagramm Daten zu generieren.
77	Join Group missing error message
79	Diagram für Ausbildungsinvestition
81	BUG: Firefox store.getters['auth/isAdmin']
87	Sonderzeichen für Passwort Check ergänzen,
88	Es kann mehrfach der selbe Benutzernamen registriert werden.
90	In der Gruppe soll eine Liste aller Gruppenmitglieder ersichtlich sein.
91	Projektleiter Aufwand speichern
92	Lohn pro Runde
93	Risiko Manager Reduktion wird erst nach neu laden angepasst
95	Project Manager Tasks duplicates in PL Management
97	Performance nach Runde starten / beenden.
98	Gruppe id = 2 hat mehrfache Projektleiter Time Allocation Einträge
99	Reihenfolge Kosten und Produktivität im Mitarbeiter Mgmt. anpassen
100	Zufallsgenerator Range FP und Kosten weniger breit einstellen
101	Risiko "Missverständnis bei den Anforderungen ist in der zweiten Gruppe verschwunden.
102	Reduzierte Risiken können nicht mehr bearbeitet werden klar ersichtlich machen.
103	Mitarbeiter Mgmt. im Backend generieren.
107	Datenbank bereinigen von duplicates im time allocations
108	E-Mail override funktioniert nicht mehr
110	Risiko Management beim Reduzieren bleibt der Wert im Formular
112	Change Request werden nicht erstellt, und können nicht gespeichert werden.
113	Im Projekt-Report wird beim Changerequest beim Scope das Budget ausgegeben.
114	Im Risikomanagement Column totals ausgeben
115	Bei den einzustellenden Mitarbeitern ist die Ausbildung noch nicht dazugerechnet.
118	RiskRedcutionEffectiveness nicht im Response mitgeben
121	EV Berechnung stimmt nicht im Dashboard

Tabelle 6: Issues welche während den Livetests erfasst wurden.

Zudem haben wir durch die Live Tests im Unterricht gesehen, dass das Anwendungsszenario 1 (siehe Unterunterabschnitt 5.4.2) unterstützt wird.

#### 5.9.4. Load Tests

In den Live Tests konnte lediglich bewiesen werden, dass die Applikation die Last, welche in Szenario 1 beschrieben ist, zu stemmen vermag. Daher haben wir zusätzlich Load Tests durchgeführt, um auch Auslastungen der Applikation, welche im Szenario 2 definiert sind, (beschrieben in Unterunterabschnitt 5.4.2) zu testen.

Die Load Tests haben wir mit dem Tool Gatling <sup>22</sup> durchgeführt. Dazu haben wir folgenden Ablauf definiert, welcher danach mit 160 Usern simuliert wurde.

1. Einloggen
2. Dashboard aufrufen
3. Mitarbeiter Management aufrufen
4. Drei Mitarbeiter abweisen
5. Projektmanager Management aufrufen
6. Einen neuen Wert setzen
7. Risiko Management aufrufen
8. Einen gespeicherten Change Request aufrufen
9. Projekt Report aufrufen
10. Abmelden

Folgend sind die Ergebnisse der Load Tests ersichtlich. Die Abbildung 35 zeigt eine detaillierte Ansicht aller unterschiedlichen Request. Für jeden Request Typ ist die totale Anzahl der gesendete Requests, die dazugehörige Fehlerzahl sowie die durchschnittliche Zeit der dazugehörenden Responses ersichtlich.

Wir können erkennen, dass es bei keinem Request fehlerhafte Responses gab. Zudem waren 75 % der Requests unter einer Zeit von 1729 ms, was ein guter Wert ist und 50 % sind sogar unter 314 ms, was sehr schnell ist.

---

<sup>22</sup><https://gatling.io/>

STATISTICS												Expand all groups   Collapse all groups	
Requests +	Executions					Response Time (ms)							
	Total	OK	KO	% KO	Cnt/s	Min	50th pct	75th pct	95th pct	99th pct	Max	Mean	Std Dev
Global Information	9954	9954	0	0%	158	10	314	1729	4766	7484	9912	1156	1682
request_...th/login	158	158	0	0%	2.508	64	966	2008	4584	7813	7970	1427	1602
request_...group/19	158	158	0	0%	2.508	49	2680	4203	6295	8366	9390	2702	2129
request_...peBudget	158	158	0	0%	2.508	101	2653	3942	6319	8398	9367	2572	2088
request_...user/me	158	158	0	0%	2.508	46	2077	3209	6321	8060	8562	2281	1931
request_...h/game/4	158	158	0	0%	2.508	20	1465	2674	5035	7653	7957	1844	1747
request_...BurnDown	158	158	0	0%	2.508	138	2608	4461	6518	8624	9536	2874	2274
request_...ageCount	158	158	0	0%	2.508	14	1595	2965	5326	7710	8045	1939	1792
request_...sion/197	158	158	0	0%	2.508	21	1362	2542	4900	7596	7860	1745	1701
request_...BurnDown	158	158	0	0%	2.508	86	2970	4408	6514	8623	9577	2904	2230
request_...h/game/4	158	158	0	0%	2.508	66	3184	5511	6874	8824	9912	3334	2529
request_.../session	158	158	0	0%	2.508	43	2195	3620	5532	7499	8627	2328	1877
request_...gameId=4	158	158	0	0%	2.508	52	2042	3220	5658	7886	8641	2230	1873
request_...oupId=19	158	158	0	0%	2.508	64	1603	2922	5248	7603	7892	1981	1788
request_...oupId=19	158	158	0	0%	2.508	97	2582	4010	6290	8394	9109	2635	2119
request_...agement	158	158	0	0%	2.508	90	2494	3895	6295	7883	9390	2577	2062
request_...gameId=4	158	158	0	0%	2.508	110	1437	2767	5303	7580	7890	1897	1794
request_...sion/197	158	158	0	0%	2.508	15	2226	3453	5498	8006	8709	2310	1909
request_.../session	158	158	0	0%	2.508	19	1510	2609	5597	7613	7885	1920	1796
request_.../session	158	158	0	0%	2.508	61	2154	3355	5989	7970	8752	2346	1952
request_.../session	158	158	0	0%	2.508	114	2058	3425	5916	7961	8601	2277	1925
request_...BurnDown	158	158	0	0%	2.508	16	1652	2801	4910	7604	7797	1941	1728
request_...peBudget	158	158	0	0%	2.508	16	1517	2771	5605	7650	7989	1910	1813
request_...group/19	158	158	0	0%	2.508	76	1496	2741	5154	7608	7939	1884	1792
request_...agement	158	158	0	0%	2.508	38	1363	2930	4944	7625	7875	1878	1776
request_...ageCount	158	158	0	0%	2.508	21	2631	3979	6049	8412	9358	2616	2058
request_...user/me	158	158	0	0%	2.508	26	1442	2828	5345	7591	7925	1887	1793
request_...th/login	158	158	0	0%	2.508	167	2243	3514	6023	8067	8472	2386	1980
request_...BurnDown	158	158	0	0%	2.508	31	1359	2866	4614	7637	7943	1766	1655
request_...staffing	158	158	0	0%	2.508	11	62	536	958	1693	1842	272	377
request_...sion/197	158	158	0	0%	2.508	31	544	1589	2166	2421	2885	890	729
request_...ng/2/get	158	158	0	0%	2.508	11	74	355	665	1236	1565	209	264
request_...sion/197	158	158	0	0%	2.508	14	72	355	690	1547	1801	221	295
request_...ng/3/get	158	158	0	0%	2.508	33	539	1578	2157	2541	2886	876	732
request_...h/game/4	158	158	0	0%	2.508	39	1006	2387	3485	3940	3966	1423	1184
request_...ng/2/get	158	158	0	0%	2.508	33	544	1573	2157	2274	2390	869	700
request_...ng/3/get	158	158	0	0%	2.508	14	69	355	668	1550	1651	213	283
request_...ng/1/get	158	158	0	0%	2.508	12	75	355	690	1574	1802	228	313
request_...staffing	158	158	0	0%	2.508	34	541	1583	2162	2422	2885	885	730
request_...ng/1/get	158	158	0	0%	2.508	32	543	1596	2158	2444	2886	884	730
request_...ng/1/get	158	158	0	0%	2.508	16	319	732	1965	2475	2749	541	645
request_...ng/2/get	158	158	0	0%	2.508	12	23	72	524	832	1553	108	211
request_...ng/2/get	158	158	0	0%	2.508	17	148	303	1923	2655	3164	500	722
request_...ng/3/get	158	158	0	0%	2.508	17	152	302	1935	2651	3163	499	721
request_...ng/3/get	158	158	0	0%	2.508	10	40	79	465	1070	1919	116	231
request_...agerTime	158	158	0	0%	2.508	11	22	41	238	844	975	67	141
request_...agerTime	158	158	0	0%	2.508	22	143	329	1593	1838	1858	359	492
request_...location	158	158	0	0%	2.508	12	34	49	173	441	767	58	87
request_...location	158	158	0	0%	2.508	20	140	351	1595	1856	1921	359	497
request_...location	158	158	0	0%	2.508	10	19	27	138	184	260	32	39
request_...agerTime	158	158	0	0%	2.508	21	80	165	490	1475	1631	163	245
request_...location	158	158	0	0%	2.508	24	123	328	608	1987	2679	234	360
request_...agement	158	158	0	0%	2.508	10	17	24	41	93	106	21	13
request_...agement	158	158	0	0%	2.508	16	43	63	402	495	1883	90	177
request_...sion/197	158	158	0	0%	2.508	11	14	19	26	40	46	16	6
request_...sion/197	158	158	0	0%	2.508	13	24	45	169	174	182	41	40
request_...BurnDown	158	158	0	0%	2.508	11	16	20	26	38	39	17	6
request_...BurnDown	158	158	0	0%	2.508	11	21	28	35	38	48	22	7
request_...BurnDown	158	158	0	0%	2.508	21	66	99	155	249	317	77	51
request_...BurnDown	158	158	0	0%	2.508	20	66	95	153	256	324	77	51
request_...ranking	158	158	0	0%	2.508	11	14	18	29	38	46	17	6
request_...ranking	158	158	0	0%	2.508	15	24	34	107	135	139	32	25
request_.../session	158	158	0	0%	2.508	12	14	15	22	25	27	15	3
request_.../session	158	158	0	0%	2.508	13	18	20	30	40	51	19	5

Abbildung 35: Detaillierte Ergebnisse der Load Tests

In der Abbildung 36 wird die Verteilung der Antwortzeiten aufgezeigt. Wir sehen, dass der gesamte Test eine Durchlaufzeit von ca. 30 Sekunden hat. Die orange Linie zeigt uns die aktiven User, welche bis auf die gesetzten 160 User hoch gehen. Der interessanteste Teil dieser Grafik ist jedoch die Verteilung der Response Zeiten. Hauptsächlich ist die Applikation im grünen und blauen Bereich, was sehr gute Antwortzeiten sind. Bei sehr vielen gleichzeitigen Requests gab es jedoch auch Antwortzeiten im roten Bereich, dies aber nur in der Spitze. Dies wird im Vergleich mit der Abbildung 37 deutlich, da sieht man, dass in dieser Zeit ca. zwischen 500 und 800 Requests pro Sekunde verarbeitet wurden.

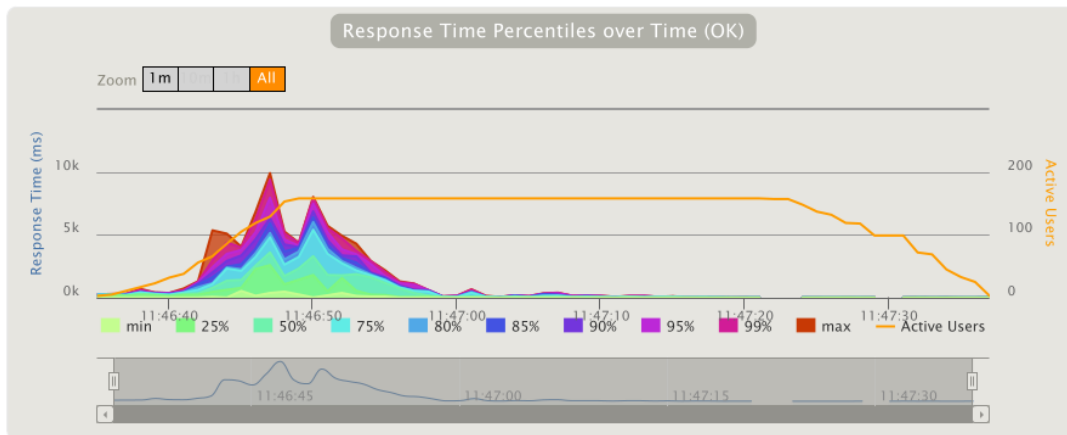


Abbildung 36: Response Zeiten und deren Prozentsätze über die Zeit

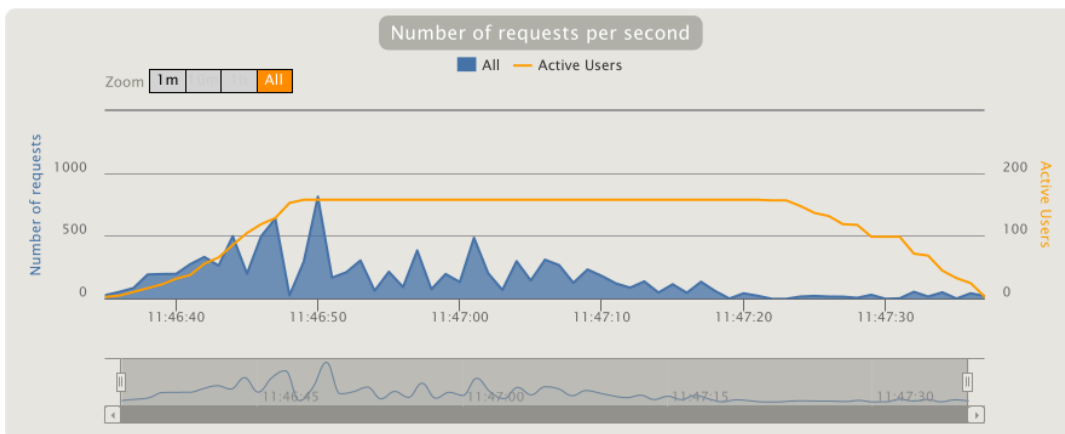


Abbildung 37: Requests über Zeit

Zum Schluss sehen wir noch die Anzahl der Responses sowie den dazugehörigen Status. In der Abbildung 38 ist ersichtlich, dass die Applikation in den Spitzen über 400 Responses pro Sekunde versendet hat. Es ist ersichtlich, dass keine fehlerhaften Responses versendet wurden, diese wären im Diagramm rot gekennzeichnet und zusätzlich in der Verteilung (Kuchendiagramm oben links) erkennbar.

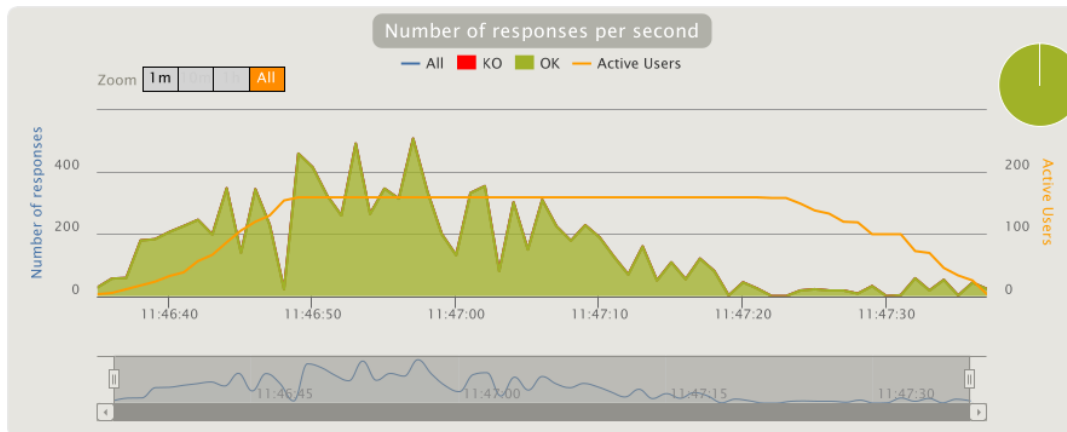


Abbildung 38: Responses über Zeit

Der Load Test haben wir auf der produktiven Umgebung durchgeführt. Dies bedeutet, dass auch Komponenten wie der Traefik Reverse Proxy und die Datenbank mit getestet wurden. Nur das Frontend war nicht Teil des Tests, da dieses direkt beim Client im Browser läuft. Die API Requests wurden daher direkt an das Backend gesendet.

Wir sind mit den Resultaten zufrieden. Die Applikation vermag die Last von 160 User, welche 9954 Requests innerhalb von ca. 30 Sekunden tätigen, gut stemmen. Mit diesem Load Test können wir auch die nicht-funktionale Anforderung zur Capacity (ersichtlich in Unterunterabschnitt 5.4.2) als erfüllt einstufen.

### 5.9.5. Erkenntnisse

Die Unit Tests im Backend waren sehr wertvoll. Am Anfang des Projekts war das Schreiben der Tests sehr aufwändig und mühsam, da nicht direkt ein Mehrwert davon ersichtlich war. Im Laufe des Projekts jedoch, wurden die Vorteile immer grösser, vor allem wenn es darum ging Verbesserungen vorzunehmen oder Fehler zu korrigieren. Falls eine Anpassung am Code zu Änderungen an den Ausgaben führte, wurde dies sofort erkannt und weitere Fehler konnten vermieden werden.

Die manuelle Durchführung der Integrationstests bei den Merge Requests sowie bei neuen Releases war sehr anstrengend. Es war schwierig darauf zu achten, dass wirklich alle Funktionen sauber getestet werden, ohne dass eine Funktion vergessen geht. Hier wäre es nützlich gewesen, eine entsprechende Checkliste zu erstellen, welche dann jedoch auch nach jeder Änderung am Code dementsprechend nachgeführt werden muss.

Das Testen hat sich als wertvoll erwiesen, denn wir konnten damit vermeiden, dass beim Deployment Probleme auftraten. Ebenfalls hatten wir auch keinen Crash der Applikation und die Live Tests konnten ohne grössere Probleme durchgeführt werden.

Es gäbe sicherlich noch weitere Ausbaumöglichkeiten zum Test Setup, wie beispielsweise automatisierte Integrationstests und automatisierte End to End Tests. Diese würden die

Codequalität weiter verbessern und dabei helfen, Fehler noch früher zu erkennen. Jedoch wäre das Einrichten solcher Tests auch ein Mehraufwand, was in einer zeitlich begrenzten Bachelorarbeit schwierig unterzubringen ist, ohne dass der Funktionsumfang darunter leidet. Da wir uns auch beide mit automatisierten End to End Tests nicht auskennen, müssten wir erheblich Zeit investieren, um uns in die Thematik und die verfügbaren Tools einzuarbeiten.

## 5.10. Installationsanleitung

### 5.10.1. Voraussetzungen

Die Applikation benötigt nur wenig Ressourcen, um betrieben werden zu können. Dennoch sollten mindestens folgende Anforderungen an die Hardware erfüllt sein.

- min. 4 vCPU Cores
- min. 4 GB freier Arbeitsspeicher
- min. 10 GB freier Speicher
- min. Docker Version 19.03
- min. Docker-Compose Version 1.17.1

### 5.10.2. Deployment auf OST Server

Wir haben auf GitLab eine CI/CD Pipeline eingerichtet, welche automatisches Deployment auf der „Staging“- und „Production“-Umgebung erlaubt. Dieses Setup haben wir für das Frontend sowie für das Backend Repository im Einsatz.

Beide Repository haben folgende CI/CD Stages:

1. **Install dependency** Hier werden alle von der Applikation benötigten Softwarebibliotheken installiert und im CI/CD Cache gespeichert. Diese Stage wird nur ausgeführt, wenn Änderungen in dem File package.json enthalten sind, ansonsten wird diese Stage nicht ausgeführt. Dadurch kann die Build Time erheblich gesenkt werden, da ansonsten bei jedem Durchlauf alle Softwarepakete neu heruntergeladen und installiert werden müssten.
2. **Build** Hier wird der gesamte TypeScript Code inklusive den zuvor installierten Softwarebibliotheken, welche aus dem Cache geladen werden, gebuildet.
3. **Assemble** Hier wird mit der zuvor gebuildeten Applikation ein Docker Image erstellt. Anschliessend wird das Image auf die Gitlab Docker Registry gepusht. Dadurch wird das Docker Image für die Installation zur Verfügung gestellt.
4. **Test** Hier werden alle geschriebenen Tests auf der gebuildeten Applikation ausgeführt. Falls einer der Tests nicht erfolgreich abgeschlossen werden kann, wird die gesamte Pipeline hier abgebrochen. Dies verhindert, dass zuvor übersehene Fehler in der Staging oder der Production Umgebung deployed werden.
5. **Deploy** Hier verbindet sich die Pipeline direkt auf den OST Server und lädt sich dort von der Gitlab Docker Registry die zuvor erstellten Docker Images herunter. Der bestehende Docker Container mit der alten Softwareversion wird heruntergefahren und es wird ein neuer Docker Container basierend auf dem neuen Docker

Image mit der neuen Softwareversion hochgefahren. Da diese Stage die aktuell laufende Staging und Production Umgebung beeinflusst muss dieser Schritt manuell im Gitlab gestartet werden.

Für das Backend Repository gibt es zudem noch eine weitere Stage:

7. **Database migration** Dieser Schritt erstellt unabhängig von der verwendeten Datenbank (z. B. Staging SQLite und Prod. MariaDB wie von uns verwendet) eine neues Datenbankschema und initialisiert dieses mit benötigten Daten wie dem Admin Benutzer. Dabei wird die alte Datenbank gelöscht. Dieser Schritt wird nur benötigt, wenn es eine Änderung am Datenbankschema gegeben hat. Da dies Einfluss auf die Daten hat, muss auch dieser Schritt manuell ausgeführt werden.

In den folgenden Abbildungen sehen wir jeweils ein Screenshot aus einer Pipeline. Die Frontend Pipeline Abbildung 39 zeigt, dass bei dieser Ausführung keine Änderungen an den verwendeten Softwarebibliotheken gemacht wurden, deshalb ist die Stage Install dependencies nicht vorhanden.

Im Backend Abbildung 40 hingegen wurde die Install dependencies Stage ausgeführt. Zusätzlich sehen wir die Database migration Stage, welche in diesem Fall nicht ausgeführt wurde, da keine Änderungen am Datenbankschema gemacht wurden.

Bei beiden ist anhand des „Play“ Symbols ersichtlich, dass die Deploy Stage manuell ausgeführt werden muss.

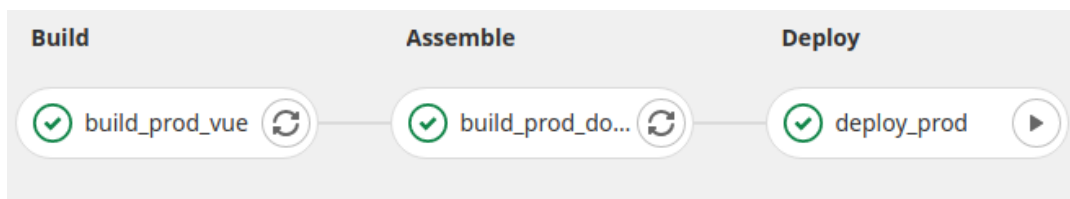


Abbildung 39: Frontend Pipeline

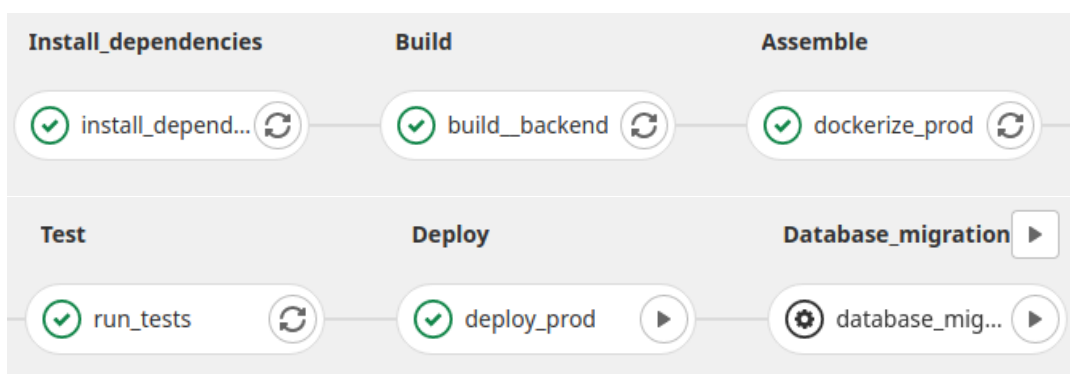


Abbildung 40: Backend Pipeline

Die jeweiligen Konfigurationsfiles (.gitlab-ci.yml) aller Pipelines sind auf dem Gitlab Repository in der Gruppe „PM Game“ <sup>23</sup> unter dem jeweiligen Projekt für Frontend und Backend verfügbar.

### 5.10.3. Unterteilung Staging & Production

Der Unterschied der beiden Pipelines besteht nur darin, dass in der **Assemble** Stage ein separates Docker Image mit dem Tag `:staging` oder `:prod` pro Umgebung erstellt wird und anschliessend in der **Deploy** Stage auch separate Docker Container daraus erstellt werden. Ansonsten sind die beiden Pipelines sowie auch Docker Images und Container identisch.

Das Deployment in die **Staging** Umgebung wird automatisch jedes Mal durchgeführt, wenn eine Änderung auf dem **Master Branch** erfolgt. Da der Master Branch geschützt ist, können nur Änderungen in Form von Merges aus anderen Branches erfolgen. Da bei einem Merge Request Prozess unter anderem die manuellen Integration Tests (beschrieben in Unterabsatz 5.9.1.2) durchgeführt werden, werden nur funktionierende Änderungen auf den Master Branch gemerged.

Damit die Pipeline für die **Production** Umgebung gestartet wird. Muss im Gitlab ein neuer Release mit Release Notes und einem Tag erstellt werden. Sobald der Tag erstellt wurde, wird eine neue Pipeline für die Production Umgebung gestartet.

Der Vorteil davon ist, dass der Code immer sauber gemerged werden kann und kein Branch für eine produktive Umgebung geführt werden muss. Zusätzlich ist es mit den Releases möglich, bei einem Problem ein Rollback der Applikation auf ein vorhergehendes Release vorzunehmen.

### 5.10.4. Konfiguration der Komponenten

Wir haben **Docker Compose** dazu verwendet, um die Konfiguration aller zusammenspielenden Komponenten zu verwalten. In einem entsprechenden `docker-compose.yml` File können alle Konfigurationen getätigt werden. Mit dem Befehl `docker-compose up` werden alle Komponenten entsprechend in Docker Container gestartet.

---

<sup>23</sup><https://gitlab.com/pm-game>

So sieht unser **docker-compose.yml** File aus:

```
version: '3'

services:

  traefik:
    image: traefik:v1.7.16
    container_name: pm_traefik
    command: --docker --docker.domain=sinv-56092.edu.hsr.ch
    ports:
      - 80:80
      - 443:443
    networks:
      - proxy
      - internal
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock3
      - ./traefik.toml:/traefik.toml
      - ./acme.json:/acme.json
    labels:
      - "traefik.frontend.rule=Host:sinv-56092.edu.hsr.ch"
      - "traefik.port=8080"
    restart: always

  pm_staging_frontend:
    image: registry.gitlab.com/pm-game/frontend:staging
    container_name: pm_staging_frontend
    environment:
      - PORT=8080
      - VUE_APP_API_URL=https://api.staging.pmgame.ch
    labels:
      - "traefik.backend=pm_staging_frontend"
      - "traefik.frontend.rule=Host:staging.pmgame.ch"
      - "traefik.docker.network=proxy"
      - "traefik.port=8080"
    expose:
      - "8080"
    networks:
      - internal
      - proxy

  pm_staging_backend:
    image: registry.gitlab.com/pm-game/backend:staging
```

```

container_name: pm_staging_backend
environment:
  - TYPEORM_CONNECTION=sqlite
  - TYPEORM_DATABASE=staging.sqlite
  - TYPEORM_SYNCHRONIZE=true
  - TYPEORM_LOGGING=false
  - TYPEORM_ENTITIES=entity/**/*.*js
  - TYPEORM_MIGRATIONS=migration/*.js
  - CORS_ORIGIN=https://staging.pmgame.ch
  - PORT=3000
expose:
  - "3000"
labels:
  - "traefik.backend=pm_staging_backend"
  - "traefik.frontend.rule=Host:api.staging.pmgame.ch"
  - "traefik.docker.network=proxy"
  - "traefik.port=3000"
networks:
  - internal
  - proxy

pm_prod_frontend:
image: registry.gitlab.com/pm-game/frontend:prod
container_name: pm_prod_frontend
environment:
  - PORT=8081
  - VUE_APP_API_URL=https://api.pmgame.ch
labels:
  - "traefik.backend=pm_prod_frontend"
  - "traefik.pm_prod_frontend.frontend.rule=Host:pmgame.ch"
  - "traefik.docker.network=proxy"
  - "traefik.port=8081"
expose:
  - "8081"
networks:
  - internal
  - proxy

pm_prod_backend:
image: registry.gitlab.com/pm-game/backend:prod
container_name: pm_prod_backend
environment:
  - TYPEORM_CONNECTION=mariadb
  - TYPEORM_HOST=pm_mariadb

```

```

- TYPEORM_SYNCHRONIZE=true
- TYPEORM_LOGGING=false
- TYPEORM_ENTITIES=entity/**/*.js
- TYPEORM_MIGRATIONS=migration/*.js
- CORS_ORIGIN=https://pmpgame.ch
- PORT=3001
expose:
  - "3001"
labels:
  - "traefik.backend=pm_prod_backend"
  - "traefik.frontend.rule=Host:api.pmpgame.ch"
  - "traefik.docker.network=proxy"
  - "traefik.port=3001"
depends_on:
  - mariadb
restart: always
networks:
  - internal
  - proxy

```

```

mariadb:
  image: mariadb:latest
  container_name: pm_mariadb
  environment:
    MYSQL_ROOT_PASSWORD: */secret-password/*
    MYSQL_DATABASE: pmpgame
    MYSQL_USER: pmpgame_user
    MYSQL_PASSWORD: */secret-password/*
  networks:
    - internal

```

```

networks:
  proxy:
    external: true
  internal:
    external: false

```

Im docker-compose ist ersichtlich, wie wir aus der GitLab Registry die entsprechenden Docker Images verwenden und anhand der Version Tags die Zuweisung der Staging oder Productive Umgebung vornehmen.

Wie in Unterabschnitt 5.6 beschrieben ist, verwenden wir ein Traefik Container als Reverse Proxy und ein MariaDB Container als Datenbank für das produktive Backend. Diese beiden Services können auch weggelassen werden, wenn man andere Services als

Reverse Proxy oder Datenbank verwenden möchte.

Sollte dies der Fall sein, müssen die Environment Variablen in den Frontend und Backend Services dementsprechend angepasst werden.

Mit den Labels werden jeweils die Konfigurationen von Traefik gesteuert. So wird beispielsweise definiert, unter welcher URL das Frontend respektive das Backend erreichbar ist.

### 5.10.5. Installation auf einem neuen Host

Falls die Software auf einem anderen Host als dem OST Server betrieben werden soll, kann dazu das in Unterunterabschnitt 5.10.4 beschriebene `docker-compose.yml` File verwendet werden. Je nach Setup kann natürlich ein beliebiger Reverse Proxy und Datenbank (sofern von TypeORM unterstützt <sup>24</sup>) verwendet werden. In diesem Fall können die beiden Services entfernt und die Environment Variablen angepasst werden.

Folgende Schritte sind zur Installation notwendig:

1. Das `docker-compose.yml` File in das gewünschte Verzeichnis kopieren.
2. Damit der Zugang zu der Gitlab Registry gewährleistet ist, sollte ein `docker login` ausgeführt werden.
3. Alle Docker Services mit `docker-compose up -d` im Hintergrund starten.

### 5.10.6. Lokales Ausführen der Applikation

Die Applikation kann auch lokal gestartet werden. Dazu muss Node.js auf dem betreffenden System installiert sein.

Folgende Schritte werden zur Ausführung des **Frontend** benötigt:

1. Der Quellcode wird vom Git Repository heruntergeladen.
2. Im Verzeichnis ein `.env` File für die Umgebungsvariablen erstellen.
3. Folgende Umgebungsvariablen (für den Port und Zugang zum Backend) im `.env` File setzen:

```
PORT=  
VUE_APP_API_URL=
```

4. Im Verzeichnis, in dem der Quellcode liegt, werden nun folgenden Befehle im Terminal ausgeführt: `npm run dev`
5. Das Frontend startet und ist auf dem Localhost nun über den im `.env` File definierten Port erreichbar.

---

<sup>24</sup>Unterstützte Datenbanken sind unter folgendem Link ersichtlich:<https://typeorm.io/>

Folgende Schritte werden zur Ausführung des **Backend** benötigt:

1. Der Quellcode wird vom Git Repository heruntergeladen.
2. Im Verzeichnis ein `.env` File für die Umgebungsvariablen erstellen.
3. Folgende Umgebungsvariablen (für den Port und Zugang zum Backend) im `.env` File setzen:

```
TYPEORM_CONNECTION =
TYPEORM_DATABASE =
TYPEORM_USERNAME =
TYPEORM_PASSWORD =
TYPEORM_HOST =
TYPEORM_SYNCHRONIZE = true
TYPEORM_LOGGING = false
TYPEORM_ENTITIES = src/entity/**/*ts
TYPEORM_MIGRATIONS = src/migration/*ts
CORS_ORIGIN =
PORT =
SENDGRID_API_KEY =
```

4. Im Verzeichnis, in dem der Quellcode liegt, werden nun folgenden Befehle im Terminal ausgeführt: `npm run start`
5. Das Backend startet und stellt eine Verbindung mit der Datenbank her. Sobald der Startprozess abgeschlossen wurde, steht die API zur Verfügung.

### 5.10.7. Datensicherung Datenbank

Damit die Datenbank der produktiven Umgebung regelmässig gesichert wird, haben wir ein Script dazu geschrieben, welches dies übernimmt.

Das folgende Script macht einen kompletten Dump der Datenbank und speichert diesen mit dem aktuellen Datum und der Uhrzeit im Namen auf dem Server im Verzeichnis Sicherungen ab.

Zusätzlich werden alle Sicherungen welche älter als 30 Tage sind automatisch gelöscht um keine Speicherprobleme zu verursachen.

```
#!/bin/bash

# redirect stdout and stderr to syslog
exec 1> >(logger -s -t $(basename $0)) 2>&1

datum=$(date +%d-%m-%Y_%H-%M)

echo PMGame Backup started at $datum
```

```

# create backup from pmgame DB Docker Container
docker exec pm_mariadb sh -c 'exec mysqldump --all-databases -
    ↪ uroot -p"$MYSQL_ROOT_PASSWORD"' > /home/pm_user/sicherungen
    ↪ /pmgame_save_`${datum}`.sql

# remove older saves than 30 days
find /home/pm_user/sicherungen/ -name "pmgame_save*" -type f -
    ↪ mtime +30 -exec rm -v {} \;

echo PMGame Backup ended at $datum

```

Dieses Script wird durch einen **Crontab** Job täglich um 23:00 Uhr ausgeführt.

Crontab Eintrag:

```

0 23 * * * /home/pm_user/sicherungen/script/pmgame_sicherung.sh
    ↪ >/dev/null 2>&1

```

**5.10.7.1. Externe Lagerung** Damit die Sicherungen nicht nur auf dem Server verfügbar sind, können diese auch auf externen Netzwerkspeicher kopiert werden. Dort können die Sicherungen auch länger archiviert werden.

**5.10.7.2. Manuelle Sicherung** Eine manuelle Sicherung kann ebenfalls vorgenommen werden. Dazu kann man folgenden Befehl auf der Konsole im Verzeichnis des *docker-compose.yml* Files ausführen.

```

docker exec pm_mariadb sh -c 'exec mysqldump --all-databases -
    ↪ uroot -p"MYSQL_ROOT_PASSWORD"' > /Pfad/zu/Sicherungen/all-
    ↪ databases.sql

```

Dieser Befehl verbindet sich direkt mit dem Docker Container mit dem Namen *pm\_mariadb* und startet dort ein MySQL Dump und legt diesen ab.

**5.10.7.3. Datenbank Wiederherstellung** Die Datenbank kann komplett wiederhergestellt werden. Suchen sie den gewünschten Datenbank Dump mit den korrekten Daten heraus.

Wiederherstellen:

```

docker exec -i pm_mariadb sh -c 'exec mysql -uroot -p"
    ↪ $MYSQL_ROOT_PASSWORD"' < /Pfad/zu/Sicherungen/all-databases
    ↪ .sql

```

Wählen sie anstatt der all-databases.sql das gewünschte Dump File aus. Es wird direkt im Docker Container mit dem Namen pm\_mariadb in der Datenbank Instanz eingespielt. **Achtung!** Diese Operation überschreibt den kompletten aktuellen Datenbestand der Datenbank.

## 5.11. Einrichtung der Entwicklungsumgebung

Damit auch lokal an der Applikation entwickelt werden kann, benötigt man einige Tools. Wir wollten das Setup jedoch sehr flexibel halten, so dass von möglichst viele unterschiedlichen Betriebssystemen aus gearbeitet werden kann. Dazu haben wir Tools ausgewählt, welche für die gängigsten Betriebssysteme verfügbar sind.

- Eine Integrated Development Environment (IDE), welche TypeScript interpretieren kann, um den Code zu schreiben.
- Eine funktionierende Installation von **Node.js**<sup>25</sup>.
- Den JavaScript Paket Manager **npm**<sup>26</sup>.
- Eine Datenbank, welche von TypeORM unterstützt wird (unterstützte Datenbanken <sup>27</sup>).
- Eine Datenbank Management Software um auf die Datenbank zuzugreifen.
- Ein **git**<sup>28</sup> Tool für die verteilte Software-Versionsverwaltung.

Für die lokale Ausführung der Frontend und Backend Applikation während der Entwicklung sind die selben Schritte aus Unterunterabschnitt 5.10.6 erforderlich.

### 5.11.1. Verwendete Tools

Wir haben für die lokale Entwicklung die Tools „Webstorm“ als IDE und „Datagrip“ als Datenbank Management Software verwendet, da wir diese bereits von vorherigen Arbeiten kannten.

Als Datenbank haben wir SQLite verwendet. SQLite kann über npm installiert werden. Dadurch mussten wir keine eigenständige Datenbank installieren.

Für die Versionierung mit git haben wir „Gitkraken“ verwendet, da es durch grafische Hilfsmittel eine hilfreiche Erweiterung zum Kommandozeilen basierten git ist.

---

<sup>25</sup><https://nodejs.org/en/>

<sup>26</sup><https://www.npmjs.com/>

<sup>27</sup>Unterstützte Datenbanken sind unter folgendem Link ersichtlich:<https://typeorm.io/>

<sup>28</sup><https://git-scm.com/>

## **5.12. Benutzerhandbuch (User Manual)**

### **5.12.1. Benutzer Verwaltung (Admin)**

Der Admin kann alle Benutzer einsehen.

1. Im Adminbereich das Register „Benutzer“ aufrufen.
2. Es werden alle Benutzer mit E-Mail Adresse, Berechtigung, dem zugeteilten Spiel und Gruppe angezeigt. Das Suchfeld ermöglicht es dem Admin, nach bestimmten Benutzern zu filtern.
3. Über den Button „Neuen Benutzer hinzufügen“ kann ein neuer Benutzer erstellt werden. Der Benutzer erhält eine E-Mail Einladung mit der Aufforderung, ein Passwort zu setzen.
4. In der Spalte E-Mail Override erscheint ein Button „E-Mail bestätigen“, sobald eine E-Mail Adressen Bestätigung erforderlich ist. Mit dem Klick auf den Button kann der Admin die Bestätigung der E-Mail Adresse manuell ausführen.

### **5.12.2. Anlegen eines neuen Spiels (Admin)**

1. Im Adminbereich das Register „Spiele“ aufrufen.
2. Auf den Button „Neues Spiel hinzufügen“ klicken, um ein neues Spiel zu erfassen.
3. Im geöffneten Menü können die Grunddaten eingetragen werden. Wichtige Grössen sind das Startbudget, Start-Scope und maximale Ausbildung. Diese beeinflussen das Spiel massgeblich.
4. Wenn man im Menü auf das Register „Zuweisung Spielelemente“ klickt, kann man die Projektleitertypen, Mitarbeitertypen und die Risiken auswählen, welche für das Spiel anwendbar sind.
5. Im Register Abschluss gibt man noch Eckdaten zum Verlauf an wie Gruppengrösse, Anzahl Runden und Wochenarbeitszeit. Man kann auch die Art der Zielsetzung festlegen, entweder so schnell wie möglich das Projekt abschliessen oder am meisten Budget beim Abschluss des Projektes übrighaben.
6. Mit dem Button „Speichern“ wird nun das Spiel erstellt. Die Spieler können diesem Spiel nun beitreten und Gruppen erfassen.

### **5.12.3. Spiel starten (Admin)**

Wurde ein Spiel angelegt, so können die Spieler selbständig Gruppen erstellen. Sobald alle Spieler sich in einer Gruppe befinden kann das Spiel gestartet werden. Der Admin hat jedoch nochmals die Option die Gruppeneinteilung anzupassen.

1. Im Adminbereich das Register „Spiele“ aufrufen.

2. Unter „Offene Spiele“ das gewünschte Spiel aussuchen und drauf klicken.
3. Prüfen, ob alle Spieler zugeteilt sind. An diesem Punkt kann die Zuteilung durch den Spielleiter noch angepasst werden.
  - Ein oder mehrere Spieler, welche sich noch nicht in einer Gruppe befinden, können einer Gruppe zugeteilt werden, indem die Spieler selektiert werden und danach auf das „hinzufügen“ Symbol bei der entsprechenden Gruppe geklickt wird.
  - Bestehende Spieler können aus einer Gruppe entfernt werden. Dazu muss beim Spieler, welcher einer Gruppe zugeteilt ist, in der Gruppenuflistung einfach auf den gewünschten Benutzernamen geklickt werden. So wird er wieder zu den nicht zugeteilten Benutzer verschoben und kann danach einer neuen Gruppe hinzugefügt werden.
  - Der Spielleiter kann direkt neue Gruppen erstellen, indem er bei Gruppen auf das „+“ Symbol klickt.
  - Gruppen können auch durch den Spielleiter entfernt werden, indem man über die gewünschte Gruppe fährt bis sie grau markiert wird, dann wird rechts ein „x“ Symbol ersichtlich, mit welchem die Gruppe gelöscht werden kann.
  - Ein oder mehrere Spieler, welcher sich im Pool mit den nicht zugeteilten Spielern befindet, können gänzlich aus dem Spiel entfernt werden, indem die gewünschten Benutzernamen durch Anklicken markiert werden und anschliessend auf „Spieler aus dem Spiel entfernen“ geklickt wird.
4. Neben den Gruppenmitglieder wird jeweils angezeigt, ob die angegebene Grösse eingehalten wird oder nicht. Das grüne „√“ zeigt an das die Gruppe im angegebenen Range ist, und das gelbe „Dreieck“ zeigt an, dass die Grösse unter- oder überschritten wurde.
5. Wenn die Zuteilung der Spieler komplett abgeschlossen ist, erscheint der „Spiel Starten“ Button, mit welchem das Spiel gestartet werden kann. So lange mindestens ein Spieler nicht im Spiel aber nicht einer Gruppe zugeteilt ist, kann das Spiel nicht gestartet werden.

#### 5.12.4. Spielereingabe (Spieler)

Solange eine Runde gestartet ist, können die Spieler Eingaben vornehmen. Folgende Möglichkeiten haben die Spieler

**Mitarbeiter Management** Die Spieler bekommen je einen fiktiven Anwärter zur Auswahl pro Mitarbeitertyp. Die Anwärter können entweder eingestellt oder abgewiesen werden. Bei der Einstellung können die Spieler mit einem Schieberegler das Pensum des Mitarbeiters festlegen. Nach der Einstellung oder Abweisung wird ein

neuer Anwärter angezeigt. Es sind max. so viele Stellenprozent möglich wie der Spielleiter für diese Runde definiert hat.

**Ausbildung** Im Register Mitarbeiter Management kann zusätzlich in die Ausbildung der Mitarbeiter investiert werden. Da jede Runde neue Mitarbeiter eingestellt werden müssen, wird die Ausbildung auf alle zukünftigen Mitarbeiter Auswirkung zeigen.

**Projektleiter Management** Die Spieler definieren mit Hilfe der jeweiligen Schieberegler, wie viel Prozent der Zeit des Projektleiters in die verschiedenen Projektleitertaufgaben investiert wird. Der Spielleiter kann diese Eingaben dann beispielsweise bei einem Change Request einsehen.

**Risikomanagement** Die Spieler erhalten eine Übersicht über alle Risiken mithilfe von zwei Diagrammen, welche die Eintrittswahrscheinlichkeit, den Schaden sowie das gewichtete Risiko zeigen. Anhand dieser Diagramme können die Spieler in die Risikoreduktion investieren. Pro Runde können maximal die vom Spielleiter definierten Funktion Points reduziert werden.

**Change Request** Wenn der Spielleiter für die betreffende Runde die Möglichkeit, einen Change Request zu stellen, freigeschaltet hat, können die Spieler einen Change Request definieren. Diese Angaben werden dem Spielleiter übermittelt.

#### 5.12.5. Change Request bearbeiten (Admin)

Sobald ein Change Request durch die Spieler erstellt wurde, erhält der Spielleiter in der Adminübersicht eine Notification. Er kann die offenen Change Requests nun bearbeiten.

1. Wenn der Spielleiter im entsprechenden Spiel drin ist, wird beim Button „Change Requests dieser Runde“ die Anzahl an unbearbeiteten Change Requests hinzugefügt. Durch Anwählen des Buttons werden alle offenen Change Requests angezeigt.
2. Den gewünschten Change Request auswählen.
3. Es wird nun der Change Request dem Spielleiter angezeigt. Auf der rechten Seite kann der Spielleiter einsehen, wie viel die Spieler in welche Projektleitertaufgaben investiert haben. Dank dieser Daten kann der Spielleiter beurteilen, ob der Change Request genehmigt oder abgelehnt wird. Der Spielleiter kann den Change Request auch nochmals zur Überarbeitung zurückweisen. In jedem Fall muss ein Kommentar angegeben werden.
4. Falls der Change Request angenommen wird, kann der Spielleiter die Werte auch vorgängig noch anpassen. Die Änderung wird für den Spieler am Ende der laufenden Runde wirksam.

#### 5.12.6. Runde beenden ankündigen (Admin)

Der Spielleiter kann das Ende einer Runde ankündigen.

1. Button „Runde beenden ankündigen“ anwählen.
2. Datum und Uhrzeit angeben.
3. Mit dem Klick auf „Ankündigung versenden“ wird eine E-Mail Benachrichtigung an alle Spieler des entsprechenden Spiels versendet. Die Benachrichtigung enthält jeweils eine Auflistung aller Eingaben der Gruppe, welche für die aktuelle Runde noch ausstehen.

### 5.12.7. Runde beenden (Admin)

Haben alle Spieler die Eingaben getätigt, kann der Spielleiter auf der Übersicht des laufenden Spiels die aktuelle Runde beenden und anschliessend eine neue Runde starten.

1. Über den Menüpunkt „Spiele“ die Seite „Spiele“ aufrufen.
2. Das gewünschte Spiel im Bereich „Aktive Spiele“ anwählen zum Öffnen.
3. Die Runde über den Button „Runde beenden“ beenden.
4. Die Spieler sehen nun eine Übersicht mit den vergangenen Eingaben.
5. Der Spielleiter kann nun eine neue Runde starten, in der die Spieler wieder Eingaben tätigen können.

## 5.13. Offene Issues

### 5.13.1. Zurückgestellt

**Spiel per Token beitreten** User, welche sich neu bei „PM Game“ registriert und angemeldet haben, können anhand eines vom Spielleiter zur Verfügung gestellten Token am korrekten Spiel teilnehmen. Auf diese Weise treten die Spieler nicht aus Versehen einem anderen offenen Spiel bei.

Dieses Issue entspricht der User Story 12 und wurde durch den Change Request zurückgestellt. Dies ist im Unterunterabschnitt 5.1.1 beschrieben.

### 5.13.2. Known Bugs

Dies sind bekannte Bugs, welche in der aktuellen Version der Software enthalten sind.

**Weiterleiten nach Gruppe beitreten funktioniert nicht** Nach dem ein Spieler einem Spiel und anschliessend einer Gruppe beigetreten ist, kommt es teilweise vor, dass der Spieler nicht automatisch zur Spielansicht weitergeleitet wird. Der Spieler muss in diesem Fall die Webseite neu laden, damit er das Spiel sieht.

Das Problem kommt nur einmal zum Beginn eines Spiels vor und tritt nicht immer auf. Da das Problem auch mit einem einfachen Neuladen der Seite umgangen werden kann, haben wir die Priorität tief angesetzt. Der Bug sollte dennoch in der nächsten Version von „PM Game“ behoben werden.

**Risiko Verteilung Diagramm Tooltip Size entfernen** Das Diagramm zur Risiko Verteilung, welches für die Spieler unter dem Register „Risiko Management“ ersichtlich ist, enthält jeweils auch den Wert „Size“ für die Grösse der Punkte. Die Grösse der Punkte ist jedoch einheitlich, da diese keiner Dimension im Diagramm zugewiesen ist. Deshalb ist der Wert „Size“ aus dem Tooltip zu entfernen.

Da keine falschen Informationen ausgegeben werden, sondern nur eine zusätzliche Konstante welche für den Spieler irrelevant ist, wurde die Priorität dafür tief angesetzt. Bisher konnte keine Lösung für diesen Bug gefunden werden. Das verwendete Framework Apex Chart bietet in der aktuellen Version keine Möglichkeit, diesen Wert in der Ausgabe zu unterbinden.

### 5.13.3. Nice to have

In diesem Abschnitt werden die offenen „Nice to have“ Issues aufgelistet. Diese Issues sind durch einen viel kleineren Umfang klar von den im Unterabschnitt 5.2 beschriebenen Erweiterungsmöglichkeiten zu unterscheiden. Sie wurden mehrheitlich während den Live Tests erfasst und umfassen hauptsächlich Erweiterungen sowie Verbesserungsmöglichkeiten im GUI, welche auf die Userexperience eine positive Auswirkung hätten. Damit diese Erkenntnisse nicht verloren gehen, listen wir diese hier auf.

**Benutzernamen ändern** Eine weitere Funktion in den Benutzereinstellungen, welche es dem Spieler ermöglicht seinen Benutzernamen zu ändern.

**Auswahl multipler Stammdaten auf einmal** Bei der Auswahl der Stammdaten wie Projektleiter Aufgaben, Mitarbeitertypen und Risiken kann im Dropdown Menü jeweils nur eine Option ausgewählt werden. Danach schliesst sich das Dropdown Menü und es muss erneut geöffnet werden, um eine weitere Option auszuwählen.

Dies sollte so angepasst werden, dass das Dropdown Menü geöffnet bleibt damit mehrere Optionen gleichzeitig ausgewählt werden können. Das Dropdown Menü sollte sich nur schliessen, wenn der User auf eine andere Stelle der Benutzeroberfläche klickt.

**Risiken nicht standardmässig auf 0 setzen** Beim Erfassen der Risiken in den Stammdaten wird die Eingabe für den „Verlust“ in Function Points standardmässig auf 0 gesetzt. Deshalb ist es möglich, das Risiko zu speichern, ohne dass der Standardwert ange-

passt wurde. Da es keine Möglichkeit gibt das Risiko im Nachhinein anzupassen, muss das Risiko in so einem Fall gelöscht und neu erstellt werden.

Hierzu haben wir uns bereits zwei unterschiedliche Lösungsansätze überlegt:

1. Den Standardwert von 0 FP Verlust entfernen und die Eingabe zwingend notwendig zu machen.
2. Eine nachträgliche Bearbeitung der Risiken ermöglichen.

**Scrolling für Nachrichten Konversation** Bei einer Nachrichtenkonversation, welche viele Nachrichten enthält, werden zuerst die ältesten Nachrichten angezeigt. Für die neuen Nachrichten muss der User nach unten scrollen. In dieser Ansicht sollte das Scrolling so eingerichtet werden, dass jeweils automatisch nach ganz unten gescrollt wird und der User bei Bedarf zu den älteren Nachrichten nach oben scrollen kann. Die Umsetzung würde damit eher den gängigen Chat Programmen entsprechen.

**Stammdaten direkt aus „Spiel erstellen“ Funktion hinzufügen** Wenn der Spielleiter ein neues Spiel erstellen möchte, bevor er die benötigten Stammdaten erfasst hat, muss er den Vorgang abbrechen und in das entsprechende Register wechseln, um dort die Stammdaten zu erfassen. Danach kann der Spielleiter erneut damit beginnen, ein Spiel zu erstellen.

Hier könnte man direkt im „Spiel erstellen“ Bereich die Möglichkeit bieten, über eine Schaltfläche die gewünschten Stammdaten zu erfassen, ohne dass ein Registerwechsel benötigt wird.

## 5.14. Screenshots

In diesem Kapitel präsentieren wir einige Screenshots der Software. Die Idee ist, dass wir dadurch auch visuelles Material zur Verfügung haben, falls wir jemandem die Software präsentieren wollen und gerade keinen Zugang zum Server haben oder der Server sich nicht auf die Schnelle starten lässt.<sup>29</sup>

---

<sup>29</sup>Diesen Tipp haben wir in der Vorlesung „SE2“ erhalten

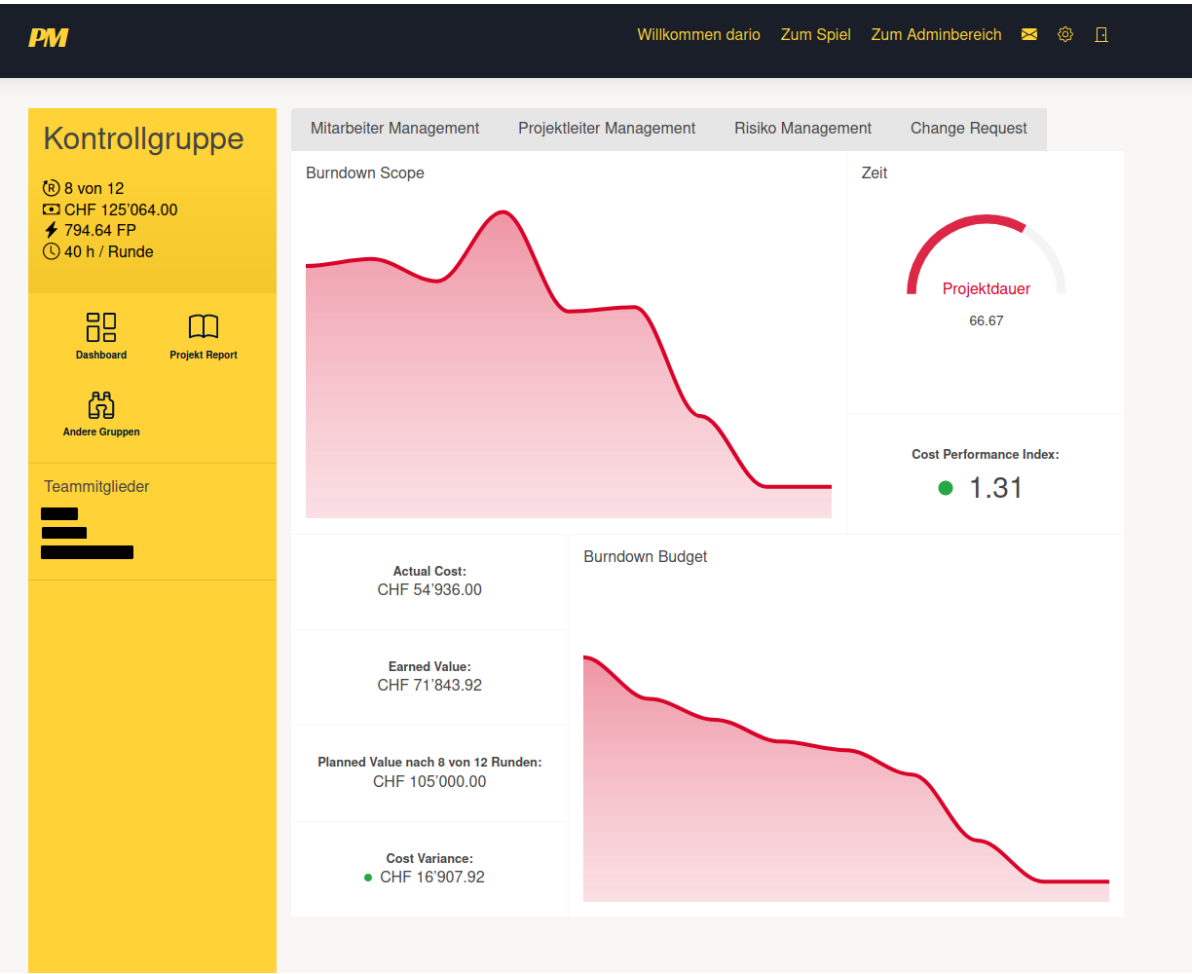





Abbildung 41: Spielansicht: Dashboard

## Kontrollgruppe

8 von 12  
CHF 125'064.00  
794.64 FP  
40 h / Runde

  
Dashboard

  
Projekt Report






  
Andere Gruppen

Teammitglieder

Mitarbeiter Management
Projektleiter Management
Risiko Management
Change Request

### Mitarbeiter Management

In dieser Runde erlaubt das Management eine max. Mitarbeiteranzahl von **500%**. Dies entspricht **5.00** Vollzeitstellen.  
Durch Investition in Ausbildung wurde die Performance um **15.50 %** gesteigert. In dieser Runde wird zusätzlich **5000 CHF** in die Ausbildung investiert.

<p><b>Kian Hahn</b> </p> <p style="font-size: x-small;">Senior</p> <p>CHF 44.00 0.73 FP Pensum: 100 %</p>	<p><b>Matteo Schmitt</b> </p> <p style="font-size: x-small;">Senior</p> <p>CHF 45.00 0.72 FP Pensum: 100 %</p>	<p><b>Sophia Krause</b> </p> <p style="font-size: x-small;">Senior</p> <p>CHF 48.00 0.73 FP Pensum: 100 %</p>
<p><b>Emma Möller</b> </p> <p style="font-size: x-small;">Professional</p> <p>CHF 54.00 0.40 FP Pensum: 70 %</p>	<p><b>Liam Koch</b> </p> <p style="font-size: x-small;">Junior</p> <p>CHF 28.00 0.43 FP Pensum: 100 %</p>	

Verwaltung Ausbildung

#### Verfügbare Mitarbeiter

Senior

**Jan Müller**

CHF 54.95 / Stunde  
0.49 / Stunde

Pensum:

AbweisenEinstellen

---

Professional

**Leo Lang**

CHF 51.87 / Stunde  
0.44 / Stunde

Pensum:

AbweisenEinstellen

---

Junior




**Maja Zimmermann**

CHF 25.5 / Stunde  
0.29 / Stunde

Pensum:

AbweisenEinstellen

Abbildung 42: Spielansicht: Mitarbeiter Management

**PM** Willkommen dario [Zum Spiel](#) [Zum Adminbereich](#)   

**Kontrollgruppe**

8 von 12  
 CHF 125'064.00  
 794.64 FP  
 40 h / Runde

Dashboard Projekt Report  
 Andere Gruppen

Teammitglieder

Mitarbeiter Management **Projektleiter Management** Risiko Management Change Request

### Projektleiterzeit

In dieser Runde arbeitet der Projektleiter **100%**. Dies bedeutet es können noch **0%** zugeteilt werden. Der Projektleiter hat einen Lohn von **60 CHF**.

Aufwand für Lenkungsausschuss: 33%

Aufwand für Projektführung: 17%


Aufwand für Stakeholder Kommunikation: 24%

Aufwand für Risikomanagement: 0%

Aufwand für Qualitätsmanagement: 14%

Aufwand für Motivation und Team-Spirit: 12%

Verteilung über alle Runden



- Aufwand für Lenkungsausschuss
- Aufwand für Projektführung
- Aufwand für Stakeholder Kommunikation
- Aufwand für Risikomanagement
- Aufwand für Qualitätsmanagement
- Aufwand für Motivation und Team-Spirit

Abbildung 43: Spielsicht: Projektleiter Management

**PM**
Willkommen dario [Zum Spiel](#) [Zum Adminbereich](#)

### Kontrollgruppe

8 von 12  
CHF 125'064.00  
794.64 FP  
40 h / Runde

Dashboard

Projekt Report

Andere Gruppen

Teammitglieder

Mitarbeiter Management
Projektleiter Management
Risiko Management
Change Request

## Risiko Management

Diese Runde erlaubt eine Risikoreduktion von max. **80.00 FP**. Bisher wurden in dieser Runde **18.00 FP** in die Risiko Reduktion investiert.

#### Risiko Verteilung

- Projektteammitglied fällt aus
- Technische Machbarkeit ist nicht gegeben
- Missverständnis bei den Anforderungen
- Schlechte Qualität der Nutzdaten
- Performance Probleme der Schnittstellen
- Zeitschätzung war nicht korrekt
- Sicherheitslücken bei der Software
- Kunde testet nicht gut
- Streit im Projektteam
- Spezifikation ist zu ungenau
- Absturz auf der Entwicklungsumgebung
- Probleme beim Go-Live

#### Gewichtetes Risiko

#### Unbehandelte Risiken

Bereits reduzierte oder ignorierte Risiken können in dieser Runde nicht erneut bearbeitet werden.

Bezeichnung	Risiko	%	Aktionen
Projektteammitglied fällt aus	28.00 FP	60%	<input type="button" value="Ignorieren"/> <input type="button" value="Reduzieren"/>
Performance Probleme der Schnittstellen	3.20 FP	40%	<input type="button" value="Ignorieren"/> <input type="button" value="Reduzieren"/>
Kunde testet nicht gut	80.00 FP	60%	<input type="button" value="Ignorieren"/> <input type="button" value="Reduzieren"/>
Absturz auf der Entwicklungsumgebung	12.00 FP	20%	<input type="button" value="Ignorieren"/> <input type="button" value="Reduzieren"/>
Probleme beim Go-Live	1.60 FP	40%	<input type="button" value="Ignorieren"/> <input type="button" value="Reduzieren"/>

#### Behandelte Risiken

Bezeichnung	Restrisiko (FP)	%	Status
Technische Machbarkeit ist nicht gegeben	1.20	20%	Ignoriert
Zeitschätzung war nicht korrekt	0.80	60%	Reduziert um 11.20 FP
Sicherheitslücken bei der Software	1.00	20%	Reduziert um 13.00 FP
Streit im Projektteam	25.70	10%	Ignoriert

Abbildung 44: Spielansicht: Risiko Management

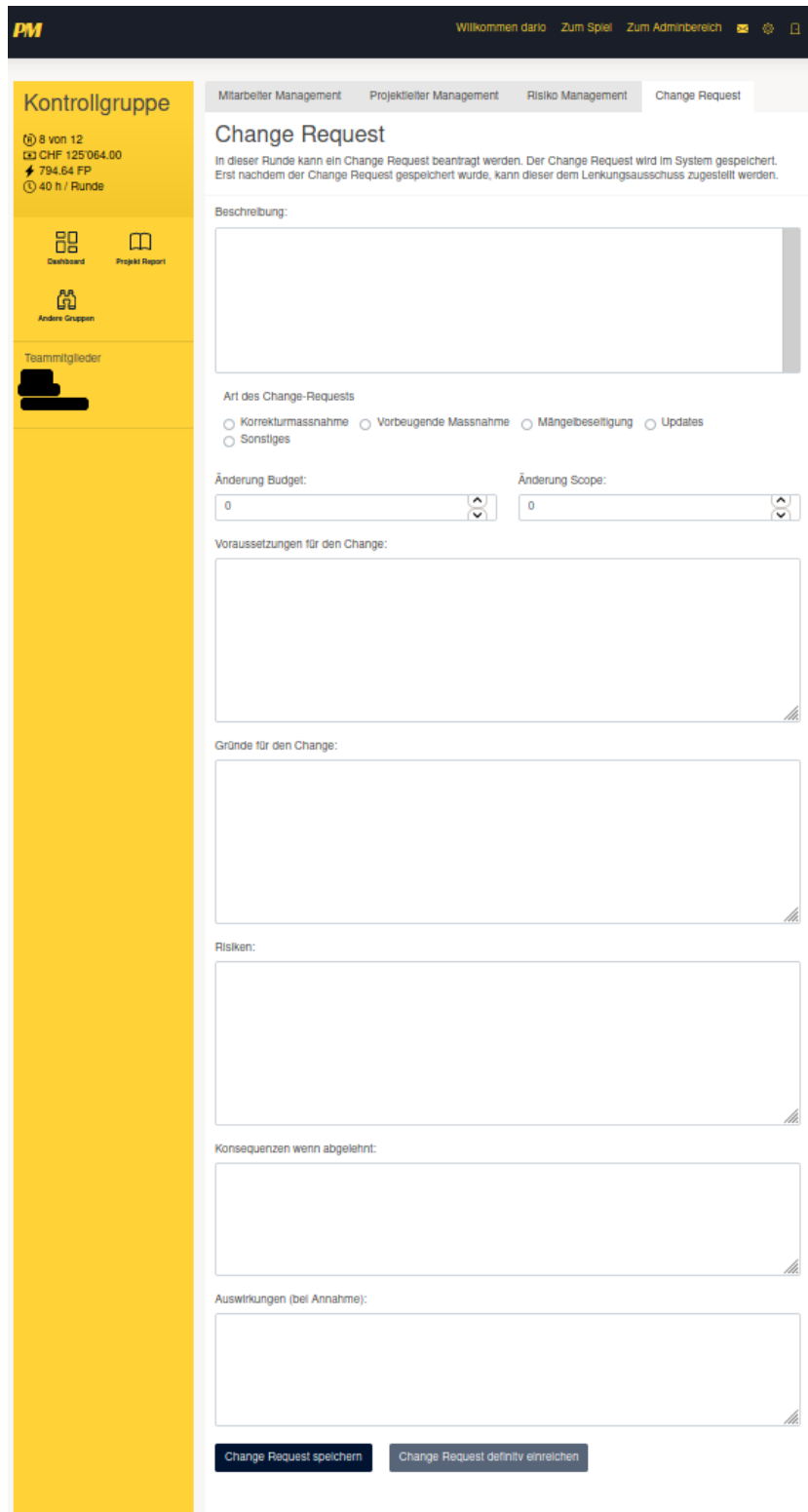


Abbildung 45: Spielansicht: Change Request Eingabe

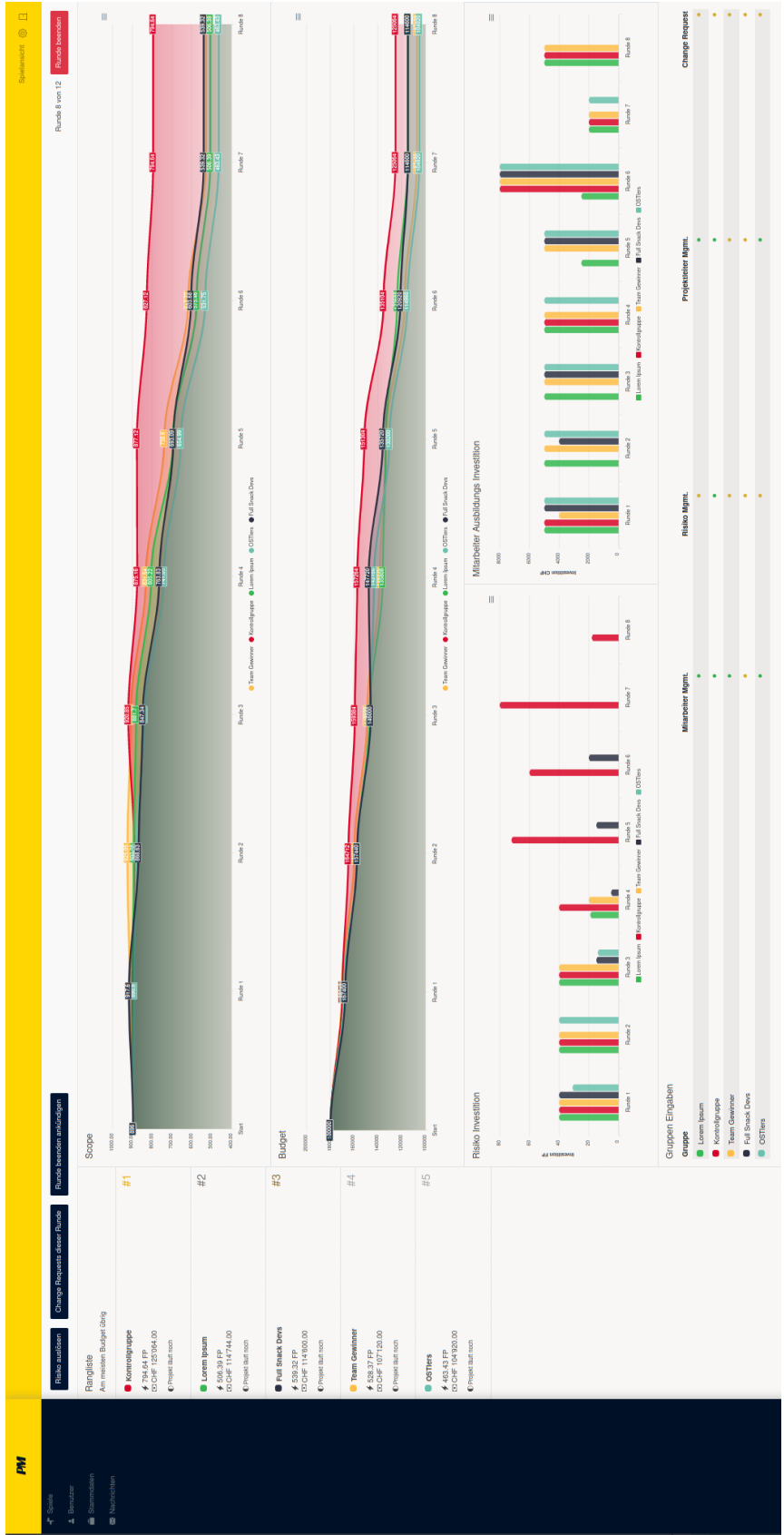


Abbildung 46: Adminbereich: Dashboard

Spieldansicht

Zurück zur Übersicht

## Change Request: Korrekturmassnahme

Erhalten am: 19.29.2020 - 03:29 **Gruppe: Gruppe 4** **Status:** akzeptiert

●	Gruppe 1 Vorbeugende Massnahme	03:23 Uhr <span style="color: red;">⊗</span>
●	Gruppe 2 Mängelbeseitigung	02:29 Uhr <span style="color: green;">⊙</span>
●	Gruppe 3 Vorbeugende Massnahme	03:23 Uhr <span style="color: red;">⊗</span>
●	Gruppe 4 Korrekturmassnahme	03:29 Uhr <span style="color: green;">⊙</span>
●	Gruppe 5 Mängelbeseitigung	03:05 Uhr <span style="color: green;">⊙</span>

### Verteilung über alle Runden

- Aufwand für Lenkungsausschuss
- Aufwand für Projektführung
- Aufwand für Stakeholder Kommunikation
- Aufwand für Risikomanagement
- Aufwand für Qualitätsmanagement

**Budget:** 8000 CHF **Scope:** 0 FP

**Beschreibung:**  
Um den termingerechten Abschluss des Projekts zu ermöglichen, schlagen wir eine Reduktion des Scopes um vor.

**Voraussetzungen für den Change:**  
Auch mit der Reduktion des Scopes kann ein funktionales Produkt abgeliefert werden.

**Gründe für den Change:**  
Durch das Eintreten der Risiken "ungenauere Spezifikation" und "Missverständnisse bei Anforderungen" wurde der Umfang grösser als ursprünglich budgetiert. Damit das Projekt trotzdem termingerecht abgegeben werden kann, beantragen wir eine Anpassung des zu erledigenden Scopes.

**Risiken:**  
Durch weitere Risiken könnte das Projekt in Zukunft weiter verzögert werden.

**Konsequenzen wenn abgelehnt:**  
Eine Ablehnung des Changes würde bedeuten, dass das Projekt nicht termingerecht abgeliefert werden kann.

**Projektleiter Pensum total:**

# CHF 9'336.00

Abbildung 47: Adminbereich: Change Request Kontrolle

## 5.15. Software-Metriken und Code Analyse

### 5.15.1. SLOC

Das Backend enthält 3'175 Source Lines of Code (SLOC). Durchschnittlich enthält eine Datei mit Sourcecode gut 99 SLOC. Zusätzlich umfassen die Tests 5'119 SLOC. Somit umfasst die das Backend inklusive der Tests 8'294 SLOC. Die Test sind jedoch oftmals repetitiv und konnten daher gut mit Copy & Paste umgesetzt werden.

Im Frontend kommen 9'782 SLOC zusammen. Dieser Wert muss jedoch relativiert werden, denn das Frontend enthält vergleichsweise wenig Logik. Der grosse Teil der SLOC wird durch semantischen Hypertext Markup Language (HTML) Code und CSS verursacht. (Gut 7'858 SLOC). Auf den JavaScript Teil entfallen 1'641 SLOC. Dieser Wert ist jedoch durch die Struktur von Vue.js und JavaScript aufgeblasen und lässt sich nicht direkt mit dem SLOC Wert des Backends (TypeScript) vergleichen.

Während dem Projekt haben wir für gut 3 Monate programmiert. Zählen wir die Anzahl SLOC des Backends (ohne Tests) und den JavaScript Teil des Frontends, so erreichen wir durchschnittlich knapp 802 SLOC pro Programmierer pro Monat. Dies liegt im normalen Rahmen.

### 5.15.2. Komplexität

Einige Controller sind etwas zu komplex geraten. Dies sind vor allem diejenigen Controller, welche am meisten Logik enthalten. Durch Refactoring könnte die Komplexität dieser Controller verbessert werden.

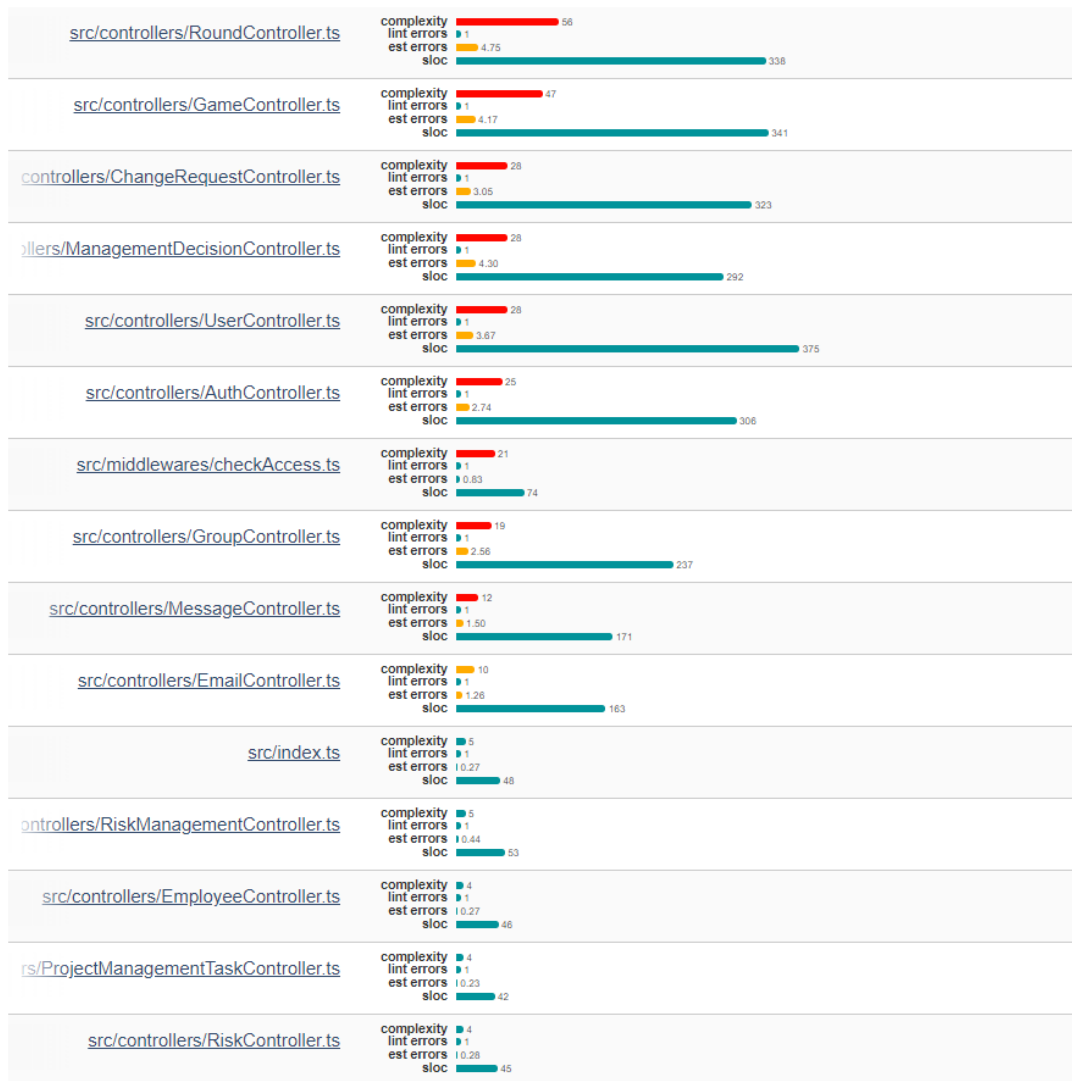


Abbildung 48: Code Komplexität Seite 1

<a href="#">src/middlewares/checkRole.ts</a>	complexity 4 lint errors 1 est errors 10.12 sloc 25
<a href="#">n/1549202832774-CreateAdminUser.ts</a>	complexity 3 lint errors 1 est errors 10.09 sloc 20
<a href="#">src/winston.ts</a>	complexity 1 lint errors 1 est errors 10.07 sloc 22
<a href="#">src/config/config.ts</a>	complexity 1 lint errors 1 est errors 0.00 sloc 13
<a href="#">src/routes/auth.ts</a>	complexity 1 lint errors 1 est errors 10.13 sloc 21
<a href="#">src/routes/changeRequest.ts</a>	complexity 1 lint errors 1 est errors 10.19 sloc 22
<a href="#">src/routes/employee.ts</a>	complexity 1 lint errors 1 est errors 10.03 sloc 12
<a href="#">src/routes/game.ts</a>	complexity 1 lint errors 1 est errors 10.28 sloc 25
<a href="#">src/routes/group.ts</a>	complexity 1 lint errors 1 est errors 10.21 sloc 23
<a href="#">src/routes/index.ts</a>	complexity 1 lint errors 1 est errors 10.13 sloc 31
<a href="#">src/routes/managementDecision.ts</a>	complexity 1 lint errors 1 est errors 10.21 sloc 22
<a href="#">src/routes/message.ts</a>	complexity 1 lint errors 1 est errors 0.01 sloc 6
<a href="#">src/routes/projectmanagertask.ts</a>	complexity 1 lint errors 1 est errors 10.04 sloc 13
<a href="#">src/routes/risk.ts</a>	complexity 1 lint errors 1 est errors 10.04 sloc 13
<a href="#">src/routes/riskManagement.ts</a>	complexity 1 lint errors 1 est errors 0.02 sloc 12

Abbildung 49: Code Komplexität Seite 2

<a href="#">src/routes/round.ts</a>	complexity 1 lint errors 1 est errors 10.14 sloc 18
<a href="#">src/routes/user.ts</a>	complexity 1 lint errors 1 est errors 10.41 sloc 33

Abbildung 50: Code Komplexität Seite 3

## 5.16. Qualität

Wir haben im Backend und Frontend statische Code Analyse angewandt und mit dem Dienst Codacy<sup>30</sup> die Werte getrackt.

<sup>30</sup><https://www.codacy.com>

Die Code Qualität im Frontend ist sehr hoch.

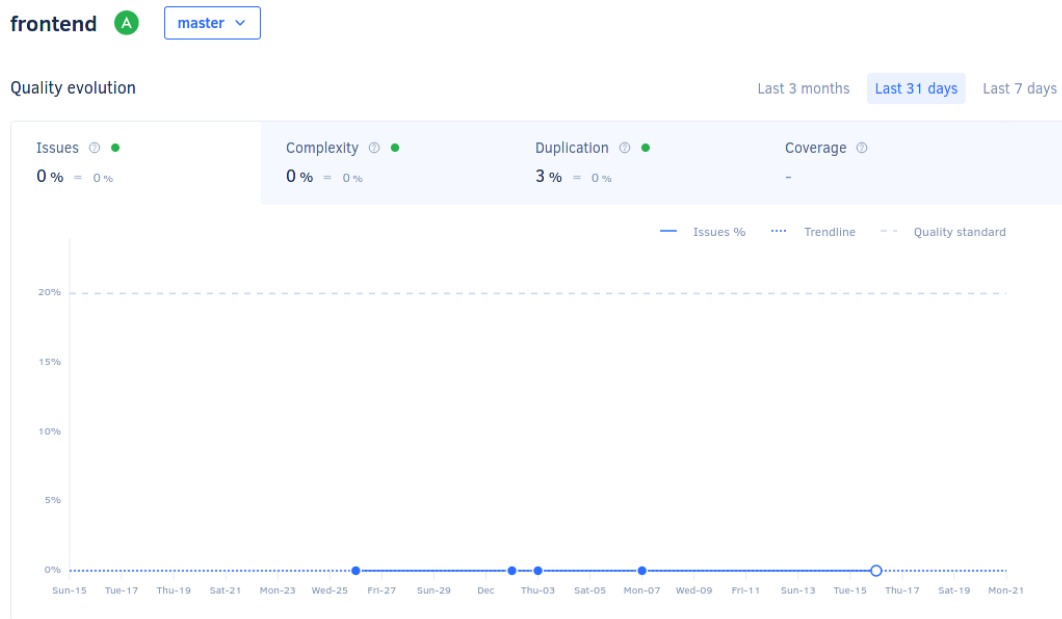


Abbildung 51: Codacy Auswertung Frontend

Im Backend konnte die Code Qualität stark verbessert werden. Das grösste Problem im Backend ist der „Duplication“ Wert. Dieser ist in der Applikation „PM Game“ merklich erhöht, da wir an vielen Stellen die Eingaben der Benutzer prüfen. Diese Prüfungen sind oftmals sehr ähnlich aufgebaut und wiederholen sich oft, daher werden diese als Duplicate erkannt. Wir konnten während der Programmierung der Applikation jedoch keine adäquate Lösung finden, um die Duplicates zu reduzieren.

Quality evolution

Last 3 months Last 31 days Last 7 days

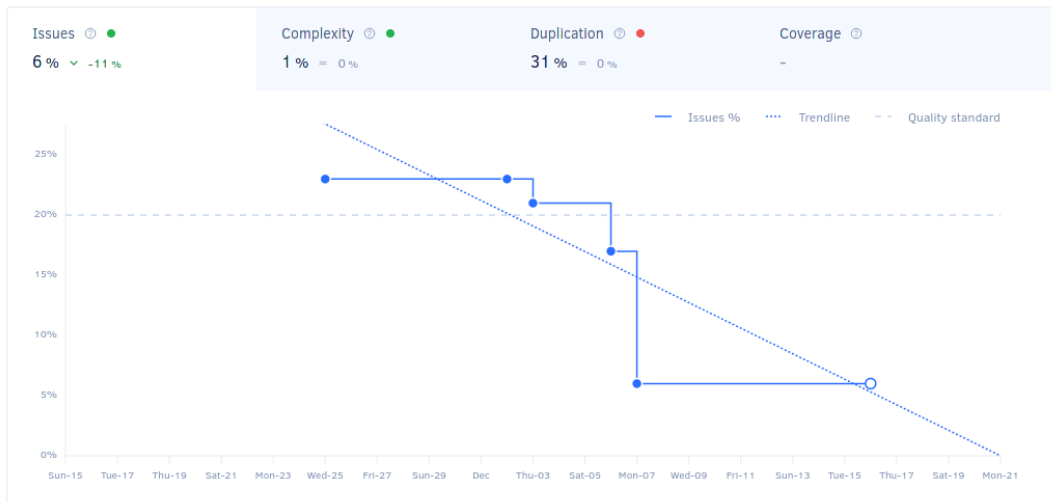



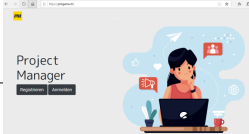
Abbildung 52: Codacy Auswertung Backend

## 6. Projektdokumentation

Dieser Teil enthält alle Dokumente, welche die Projektführung betreffen: Projektplanung, Zeitplanung, Projektcontrolling, Protokolle, etc.

### 6.1. Projektausschreibung

Das Projekt wurde von der OST als Bachelorarbeit ausgeschrieben.<sup>31</sup> Wir haben uns darauf beworben und die Zusage erhalten.

Projekt-Game programmieren	Dozent: Claudio Fuchs	Student(en): 2
<p>Diese Arbeit verbessert eine Simulation (Spiel) für das Modul PmQm, nachdem letztes Jahr die Version 1.0 geschrieben wurde. In dieser Simulation treten verschiedene Studententeams gegeneinander an und müssen rundenbasiert Entscheidungen treffen, um das Spiel zu gewinnen.</p> 	<p>Nebst den bereits umgesetzten Stories steht kommenden Herbst die Umsetzung der folgenden Anforderungen im Vordergrund:</p> <ul style="list-style-type: none"><li>■ Ich als Spielleiter habe die Möglichkeit, die Teams via Email zu informieren</li><li>■ Ich als Spielleiter kann einen Report über die Performance ziehen</li><li>■ Ich als Spielleiter kann einen Katalog an Projektrisiken vorgeben</li><li>■ Ich als Spielleiter kann festlegen, welche Risiken in welcher Runde auftreten</li><li>■ Ich als Spielleiter kann definieren, welcher Typ von Mitarbeitenden für das Projekt zur Verfügung stehen</li><li>■ Ich als Spieler habe einen groben Überblick, wie weit die anderen Teams sind (Details nur für Spielleiter)</li><li>■ Ich als Spieler werde per E-Mail informiert, wenn für meine Gruppe eine Eingabe ausstehend ist</li><li>■ Ich als Spieler muss entscheiden, welche Projektrisiken ich in welcher Runde reduziere</li><li>■ Ich als Spieler kann mir die Projektmitarbeitenden mit ihren Skills selbst zusammenstellen</li></ul>	

Das Team benötigt Erfahrung in der Software Programmierung mit NodeJS und Typescript. Auf der Tool Seite wurden Docker und Gitlab verwendet.

<https://pmgame.ch/>

HSR  
HOCHSCHULE FÜR TECHNIK  
RAPPERSWIL  
PHD Fachhochschule Ostschweiz

1  
Claudio Fuchs, PmQm, Rapperswil, Frühling 2020

Abbildung 53: Projektausschreibung

<sup>31</sup>Quelle: Claudio Fuchs, PmQm, Rapperswil, Frühling 2020

## 6.2. Aufgabenstellung



### Studienarbeit: Projekt-Game

<b>Einführung</b>	Im Zuge des Moduls Projekt- & Qualitätsmanagement soll das Erlernete spielerisch vertieft werden. Um das Interesse der Studierenden zu steigern und damit das Wissen nachhaltiger zu verankern, soll das Spiel den Charakter eines Team-Wettbewerbs haben.
<b>Aufgabe</b>	<p>Der Betreuer ist gleichzeitig der Dozent des Moduls und hat den Studierenden vorab eine Liste mit User Stories zur Verfügung gestellt. Die darin beschriebenen Anforderungen sollen in einer software-basierten Lösung als Spiel umgesetzt werden.</p> <p>Folgende (wissenschaftliche) Fragestellungen ergeben sich theoretisch, praktisch und betriebswirtschaftlich daraus:</p> <ol style="list-style-type: none"> <li>1. Verschafft der Einsatz einer Software in den Übungsstunden einen tatsächlichen Wissensaufbau, da sich die Studierenden damit auseinandersetzen müssen?</li> <li>2. Können die zusätzlichen funktionalen und nicht-funktionalen Anforderungen des Dozenten in einer Software umgesetzt werden? Begründung, wenn Anforderungen nicht wie gefordert umgesetzt werden konnten</li> <li>3. Was wäre notwendig, um diese Software aus Sicht Ostschweizer Fachhochschule als Trainingsdienstleistung für Externe anzubieten?</li> </ol>
<b>Hinweise</b>	Es besteht eine Version 1.0, welche den Studierenden bekannt sein sollte. In den Übungsstunden vom 12. November, 19. November und 26. November des Moduls PmQm besteht die Möglichkeit, eine Version des Games mit den Studierenden auszuprobieren.
<b>Erwartete Resultate</b>	<ul style="list-style-type: none"> <li>• Erwartet wird die Beantwortung der obengenannten Fragestellung in Form eines Berichts. Die Software wird einerseits als implementierte Lösung abgegeben, andererseits aber auch in Form von Source-Code und Dokumentation.</li> <li>• Die Auswahl der Vorgehensmethode ist den Studierenden überlassen. Die gewählte Methode muss jedoch zu Beginn kommuniziert und dann konsequent durchgezogen werden. Der Betreuer überwacht und bewertet die Effektivität und Einhaltung der Methode.</li> <li>• Die investierten Arbeitsstunden müssen personen- und datumsbezogen dokumentiert werden. Der Rapport wird im Anhang des Berichts beigelegt.</li> <li>• Es muss auf Basis des Stundenrapports oder der Kapitel im Bericht ersichtlich sein, wer an was gearbeitet hat.</li> </ul>

<b>Termine</b>	Es wird alle zwei Wochen eine Zwischenbesprechung angestrebt. Die Abgabe erfolgt bis spätestens Donnerstag, 17. Dezember 2020 um 12.00 Uhr.
<b>Betreuung</b>	Betreuer: Claudio Fuchs

Im Sinne der Akzeptanz unterzeichnen die Beteiligten diese Aufgabenstellung gemeinsam.

Ort, Datum:



Dario Caluzi, Student



Martin Iten, Student



Claudio Fuchs, Betreuer









## 6.4. Meilensteintrendanalyse

Um den Projektverlauf im Auge zu behalten, haben wir eine Meilensteintrendanalyse erstellt. Es ist ersichtlich, dass die Testing Phase aufgrund der noch umgesetzten Änderungen länger gedauert hat. Hierdurch wurde auch die Phase, welche wir zur Dokumentation der Arbeit eingeplant haben, verschoben und verlängert.

Projekt:

Aktualisierung:

Projektstart:

PM Game mit Stand vom 23. Dezember 2020

Meilenstein-Titel	Plan-Termin	Neue Abschätzung der Termine													
		14.09.20	28.09.20	12.10.20	26.10.20	09.11.20	23.11.20	07.12.20	21.12.20	04.01.21	18.01.21				
Anforderungsanalyse durchgeführt	01.10.20		01.10.20	01.10.20											
Systemdesign erstellt	06.10.20		06.10.20	06.10.20											
Programmierung umgesetzt	11.11.20		11.11.20	11.11.20	11.11.20	11.11.20	11.11.20	11.11.20							
Testing abgeschlossen	29.11.20		29.11.20	29.11.20	29.11.20	29.11.20	29.11.20	03.12.20	08.12.20						
Arbeit abgegeben	17.12.20		17.12.20	17.12.20	17.12.20	17.12.20	17.12.20	17.12.20	15.01.21	15.01.21	15.01.21	15.01.21	15.01.21	15.01.21	15.01.21

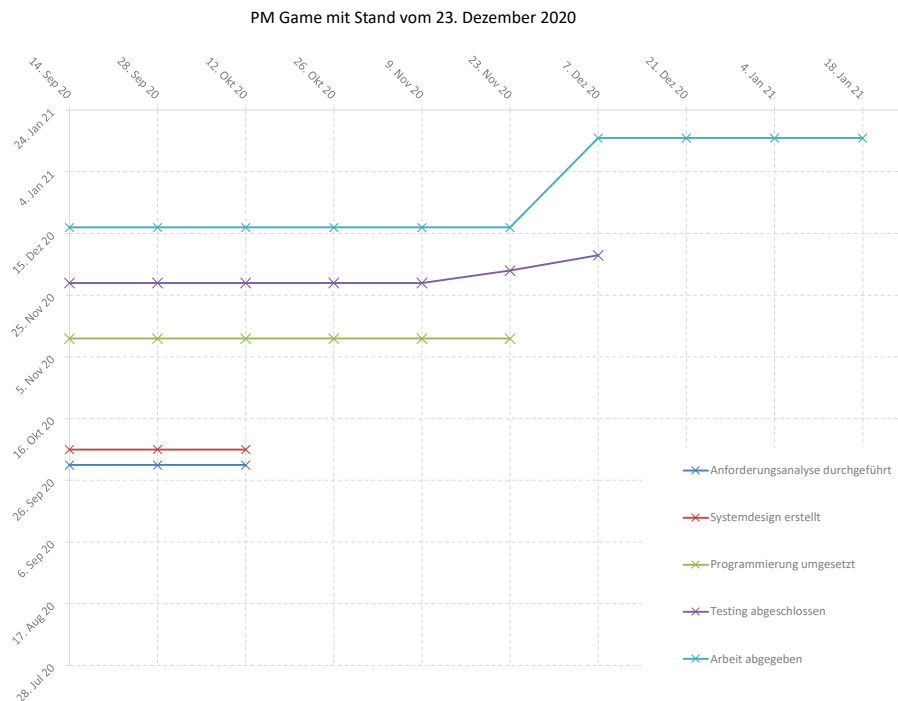


Abbildung 58: Meilensteintrendanalyse

## 6.5. Projektplan

### 6.5.1. Kritischer Rückblick: Vergleich Ist vs. Soll

Nachfolgend werfen wir einen kritischen Blick auf den Projektverlauf mit Bezug auf den Projektplan. Der Projektplan wurde am Anfang des Projekts von uns erstellt und ist im

Anhang A ersichtlich.

Im Projektplan haben wir festgehalten, wie das Projekt aus unserer Sicht verlaufen sollte. Nun prüfen wir, ob wir dies auch tatsächlich so umgesetzt haben.

- Projektübersicht: Ziele
  - ? Ziel 1 kann bei Abgabe dieser Dokumentation noch nicht beantwortet werden
  - ✓ Ziele 2-5 wurden aus unserer Sicht erfüllt
- Lieferumfang
  - ✓ Zum Zeitpunkt dieser Dokumentation wurde der Sourcecode, eine lauffähige Software sowie diese Dokumentation fertiggestellt.
- Projektorganisation
  - ✓ Microsoft Teams konnte gut genutzt werden. Der regelmässige Austausch zwischen den Projektmitgliedern erfolgte ohne nennenswerte Probleme.
  - ✓ Für die Ablage unserer Dokumente haben wir SharePoint verwendet. Dies hat gut geklappt.
  - ✓ Die Kommunikation mit dem Projektbetreuer klappte gut.
- Managementabläufe
  - ✓ Das Wasserfallmodell wurde gut eingehalten.
  - x Die geplanten Meilensteine „Testing“ und „Auslieferung“ mussten nach hinten geschoben werden, da die Phase „Testing“ mehr Zeit als geplant beanspruchte.
  - ✓ Die Builds für die Livetests liefen stabil ohne Ausfälle.
  - x Der zweiwöchige Rhythmus für die Besprechungen wurde nicht eingehalten. Bedingt durch die Remote-Arbeit wurden bei Bedarf Ad-Hoc Sitzungen abgehalten. Dies hat aus unserer Sicht jedoch der Qualität nicht geschadet.
- Risikomanagement
  - ✓ Die Liste mit den Risiken war Ende der Phase „Anforderungsanalyse“ erstellt.
  - ✓ Die Risiken wurden regelmässig neu beurteilt.
- Risikomanagement
  - ✓ Die CI/CD Pipeline hat gut funktioniert.
- Qualitätsmassnahmen
  - ✓ Die Definition of Done wurde jeweils eingehalten.

- x Anstelle eines Labels „Review“ wurden direkt die Merge-Requests verwendet, sobald ein Arbeitspaket bereit für den Review war. Der Ablauf unterscheidet sich hierdurch jedoch nicht.
- Generelle Qualitätsmassnahmen
  - ✓ Besprechungen wurden regelmässig Add-Hoc durchgeführt.
  - ✓ Wichtige Informationen wurden in Microsoft Teams bez. SharePoint abgelegt.
  - x Der Chat „Gitlab Notifications“ wurde nicht gross genutzt. Die betroffenen Teammitglieder haben von Gitlab direkt eine E-Mail erhalten. Dies hat sich als praktischer herausgestellt.
  - ✓ Die Codereviews wurden sorgfältig durchgeführt.
  - ✓ Wo nötig wurden Checklisten in Arbeitspaketen verwendet und abgearbeitet.
- Code Qualität
  - ✓ Das Frontend wurde vollumfänglich mit eslint abgedeckt.
  - x In Backend wurden die eslint Checks nicht automatisiert in der CI/CD Pipeline durchgeführt.
  - ✓ Die generelle Codequalität ist hoch → siehe Codacy
  - ✓ Die Coverage im Backend beträgt 85 %
  - ✓ Das Frontend wurde durch die Livetests und bei den Releasetests auf der Staging Plattform intensiv manuell getestet.
  - x Wir haben keine Merge-Request-Templates verwendet.
  - ✓ Das 4-Augen-Prinzip wurde ausnahmslos umgesetzt. Die Review Funktion im Gitlab wurde intensiv genutzt.
  - ✓ Es wurde regelmässig auf den Masterbranch gemerget. Die CI/CD Pipeline führt dabei Linter (Umgesetzt als pre commit hooks) und Unit Tests durch.
- Testing
  - ✓ Unit Tests wurden durchgeführt.
  - ✓ Integration Tests wurden manuell durchgeführt.
  - ✓ Lasttests wurden durchgeführt.

## 6.6. Risikoliste

In der Phase „Anforderungsanalyse“ haben wir eine Liste mit technischen Risiken zusammengestellt. Diese Liste wurde regelmässig neu überprüft.

## Risikomanagement: Technische Risiken

Projekt:	Project Manager
Erstellt am:	21.09.2020
Zuletzt aktualisiert am:	20.12.2020
Autor:	Dario Caluzi, Martin Iten
Gewichteter Schaden (in Stunden):	49.44

Risikoakzeptanzlinie	<p>Es werden nur Risiken akzeptiert, die:</p> <ul style="list-style-type: none"> <li>- einen gewichteten Schaden von maximal 1.5 Arbeitstagen (12 Stunden) haben</li> <li>- die Summe der gewichteten Schäden die Länge einer Woche (50h) nicht übersteigt.</li> </ul> <p>Beachte: Aus Darstellungsgründen enthält die Spalte "gewichtete Schaden" für die Risikoakzeptanzlinie nicht den gewichteten Schaden. Die Blasengrösse (12h) würde sonst die Risikomatrix unleserlich machen, da</p>
----------------------	---

Nr	Titel	Beschreibung	Max. Schaden	Eintrittswahrscheinlichkeit	Gewichteter Schaden	Risikostrategie (reduzieren, vermeiden, abwägen, selbst tragen)	Vorbeugung (Massnahme)	Verhalten beim Eintreten
R1	Scope creep	Wichtige Anforderungen werden in dem Anforderungsanalyse übersehen.	8	20%	1.6	reduzieren		
R2	Mailversand	Gegebenheiten durch OST IT Umgebung erschweren oder verhindert den Mailversand vom PM Game aus.	12	70%	8.4	reduzieren	Vorgehende Abklärung mit OST IT Dienste.	
R3	Gitlab offline	Gitlab.com ist nicht verfügbar, und der Zugang zu den Issues sowie Softwareversionisierung ist nicht möglich.	24	1%	0.24	selbst tragen	Lokaler Softwarestand stets aktuell behalten, damit lokal weiter gearbeitet werden kann.	
R4	CI/CD Build Minuten	Die kostenlosen Build Minuten auf GitLab sind aufgebraucht.	4	90%	3.6	vermeiden	Regelmässige Prüfung der verfügbaren Build Minuten	Aufsetzen eines eigenen Build Servers oder kaufen von Build Minuten
R5	Infrastruktur Probleme	Probleme beim Aufsetzen oder einrichten der gewählten Infrastrukturkomponenten.	16	30%	4.8	reduzieren	Genaues Studium der Dokumentationen und Funktionen der verwendeten Komponenten.	Ausweichen auf alternative Komponenten
R6	Webframework unzureichend	Das gewählte Webframework kann gewünschte Funktionalitäten nicht umsetzen.	8	10%	0.8	selbst tragen	Umfangreiches Webframework wählen.	Gewünschte Funktionalität auf eine andere Art umsetzen.
R7	Unzureichende Projektplanung	Durch unzureichende Planung können Meilensteine nicht eingehalten werden oder nicht alle geplanten Arbeitspakete können während der Programmierungsphase umgesetzt werden.	8	20%	1.6	selbst tragen	Sorgfältige und genaue Planung, priorisierung der Arbeitspakete, regelmässige Standortbestimmung und Absprache mit Projektleiter.	Durch Mehraufwand kompensieren. Issue identifizieren, welche nicht 100 prozentig benötigt werden.
R8	Projektdokumentation vernachlässigt	Es wird zu viel Zeit in die Umsetzung der Software investiert und die Projektdokumentation wird vernachlässigt.	32	20%	6.4	reduzieren	Stetige Überwachung des Fortschritts, regelmässige Besprechungen im Team.	
R9	Unzureichendes Klassendiagramm	Das definierte Klassendiagramm führt zu konzeptionellen Fehlern in der Applikation.	16	25%	4	selbst tragen	Klassendiagramm im Team und mit Projektleiter absprechen.	Nachträgliches Korrigieren des Klassendiagramms und der Software.
R10	Exportfunktion Komplexität	Die Komplexität der Exportfunktion in ein CSV ist höher als angenommen.	16	40%	6.4	selbst tragen	Einfacheren Export anbieten, beispielsweise als PNG	
R11	Nicht funktionale Anforderungen nicht erfüllt.	Nicht funktionale Anforderungen nicht erfüllt.	40	5%	2	reduzieren	Nicht funktionale Anforderungen klar definieren und überprüfbar machen.	
R12	Softwarestand Livetest	Der Build für den Livetest funktioniert nicht.	16	20%	3.2	reduzieren	Versionen der Software werden sauber getagged. Aktuelle Version wird auf Production deployed und vorgänger Version wird auf eine "old" Domain	Studenten werden auf die "old" Version verwiesen, damit sie dort testen können.
R13	Livetest identifiziert gravierender Fehler	Beim Livetest mit Studenten wird ein tiefgehender Fehler entdeckt, welche nicht in der geplanten Zeit für das Refactoring korrigiert werden kann.	64	10%	6.4	reduzieren	Verwendung automatisierter Tests und deren Einbindung in CI/CD. Bei Code Reviews wird auch ein manuelles Testen der neuen Funktionalität lokal durchgeführt.	

### 6.6.1. Eingetretene Risiken

Das Risiko „R13 - Livetest identifiziert gravierender Fehler“ ist teilweise eingetreten. Während den Livetests wurde zwar kein gravierender Fehler gefunden dessen Behebung sehr Zeitintensiv gewesen wäre. Jedoch wurden ein paar kleinere Bugs gefunden sowie diverse Verbesserungen vorgenommen. Dies hat dazu geführt, dass die eingeplante Zeit nicht ausgereicht hat und die Testing Phase entsprechend länger dauerte. Wir konnten dann mittels Change Request eine Verlängerung des Projekts beantragen, um die anschliessende Dokumentation der Arbeit sauber abschliessen zu können.

Das Risiko „R9 - Unzureichendes Klassendiagramm“ ist ebenfalls teilweise eingetreten. Kleinere Anpassungen wurden vorgenommen, jedoch hat sich das grundlegende Schema der Software hierdurch nicht verändert. Es handelte sich vor allem um Umbenennungen gewisser Properties sowie dem hinzufügen neuer Properties. Etwas angepasst wurde die Relation zu den Messages, da wir uns entschieden haben diese leicht verändert umzusetzen als es ursprünglich geplant war. Diese Anpassungen erfolgten jedoch allesamt im Vorhinein. Es musste daher kein Code umgeschrieben werden, weil sich unser Klassendiagramm als unzureichend herausstellte. Der Schaden beträgt somit weniger als zwei Stunden, welche für die Anpassung des Diagramms verwendet wurden.

## 6.7. Zeitauswertung

Wir haben mit PowerBi<sup>32</sup> eine grafische Auswertung erstellt, um den zeitlichen Verlauf des Projekts auswerten zu können.

In dieser Auswertung ist ersichtlich, dass wir die Arbeit sehr gut aufgeteilt haben und praktisch gleich viel gearbeitet wurde. Auch die Anzahl an Stunden über die gesamte Projektdauer ist relativ gleichmässig verteilt.

Am meisten Zeit hat die Umsetzung der Software in Anspruch genommen. Rechnen wir die Kategorien Coding, Review, Bug und Testing zusammen, erhalten wir einen Zeitanteil von gut 50 %. Weiter wurde ein grosser Teil der Zeit durch die Dokumentation der Arbeit (Documentation) beansprucht. Administrative Tasks und Arbeiten an der Architektur (Bereitstellung CI/CD, Deployment neuer Versionen etc.) sind weitere Kategorien, in der viel Zeit angefallen ist.

---

<sup>32</sup><https://powerbi.microsoft.com/de-de/>

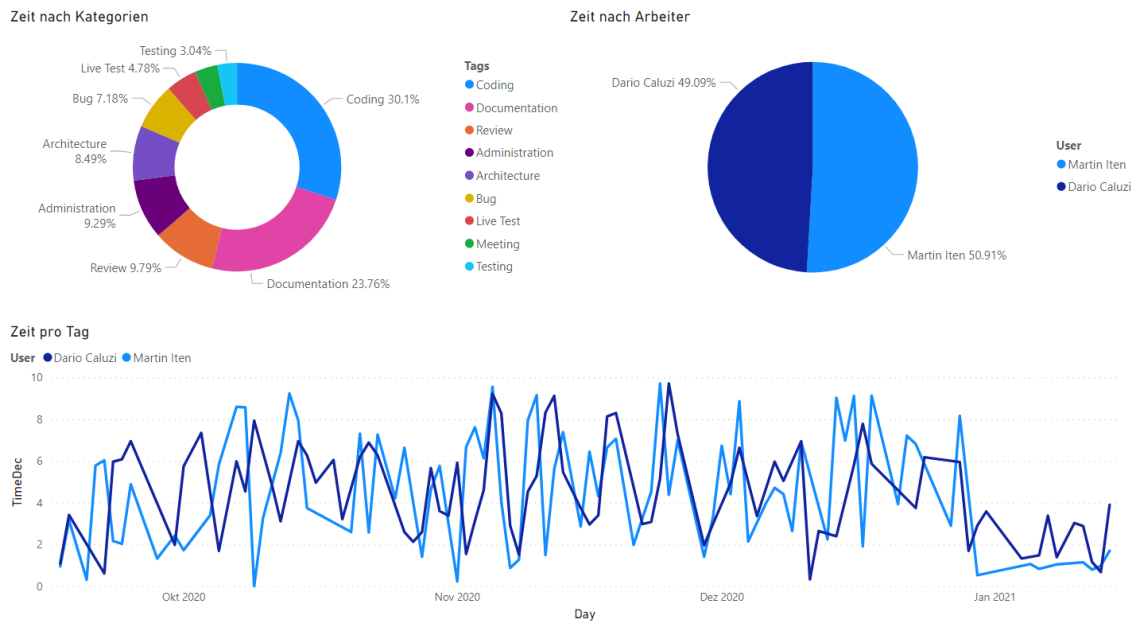


Abbildung 59: Zeitauswertung nach Stunden

Auch in der Verteilung der Stunden nach den Phasen ist ersichtlich, dass die Umsetzung am meisten Zeit beansprucht hat.

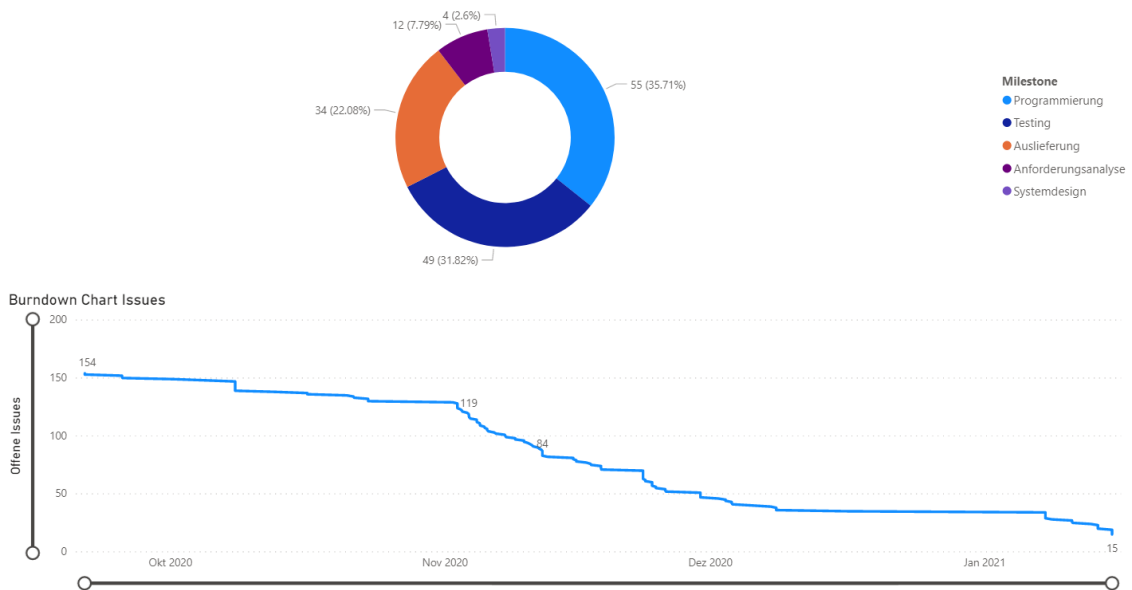


Abbildung 60: Zeitauswertung nach Issues

Im Burndown Chart kann abgelesen werden, wie die Issues über die Projektdauer abgearbeitet wurden. Das Burndown Chart bezieht sich dabei auf die totale Anzahl an

Issues im Projekt. Es kann somit nicht abgelesen werden, zu welchem Zeitpunkt neue Issues dazugekommen sind. Es ist jedoch ersichtlich, dass am Anfang des Projekts eher grössere Issues umgesetzt wurden. Diese waren eingeplant und hatten einen grösseren Umfang. Ab den Livetests sind viele kleinere Issues dazu gekommen, welche aber zügig wieder umgesetzt werden konnten.

## 7. Projektschlussberichte

In diesem Kapitel präsentieren wir unsere Erfahrungen, welche wir während dem Projekt gesammelt haben.

### 7.1. Martin Iten

Ich durfte bereits in der Studienarbeit die erste Version von „PM Game“ umsetzen. Daher hat es mich sehr gefreut, dass ich in der Bachelorarbeit dieses Projekt fortsetzen durfte. Es war mir ein Anliegen das Projekt auf einen Stand zu bringen, von dem aus es gut weiterentwickelt werden kann. Dies war in der ursprünglichen Version nicht gegeben. Der Code war nicht besonders sauber geschrieben und wies einige Schwachstellen auf. Einerseits hatte die erste Version der Applikation generell einen Proof of Concept Charakter, andererseits fehlte zur Zeit der Studienarbeit auch noch einige Fertigkeiten, welche ich in den letzten Semestern erwerben und nun in das Projekt einbringen konnte.

Im ersten Projekt hatten wir nur gut zwei Drittel der Anforderungen umsetzen können. Wir hatten die Anforderungen zu optimistisch eingeschätzt. In der Bachelorarbeit hatten wir nun einen angepassteren Scope, so dass wir nicht nur die neuen Anforderungen umsetzen, sondern auch die Software generell verbessern und optimieren konnten. Im Zuge dessen wurde die gesamte Applikation von Grund auf neu geschrieben und ich finde dies war die richtige Entscheidung. Der Code der neuen Version ist nun wesentlich schlanker und sauberer umgesetzt. Ich habe nun ein gutes Gefühl, was die Weiterentwicklung dieser Applikation betrifft.

Wir haben für dieses Projekt auf das Wasserfall Modell gesetzt. In der Studienarbeit war Scrum unser Model der Wahl. Meiner Meinung nach hat dies jedoch rückblickend auch zu Problemen geführt. Die organisatorischen Prozesse und Meetings haben viel wertvolle Zeit verschlungen. Die Vorzüge von Scrum sind dabei gar nicht gross zum Tragen gekommen, da das Projekt ein fixes Ziel, eine relativ kurze Laufzeit sowie ein kleines Team hatte. Ich muss dazu jedoch erwähnen, dass das Wasserfall Modell vermutlich nicht so reibungslos geklappt hätte, wenn ich nicht bereits Erfahrungen mit dem Projekt gehabt hätte. Diese wertvollen Erfahrungen haben enorm geholfen die Zeit am Anfang des Projekts gut einzuschätzen und die Phasen zu planen. Auch ohne die Zeiten der Tasks genau zu schätzen, haben wir die Planung erstaunlich gut getroffen. Es gab zwar in der Testingphase tatsächlich eine Verzögerung. Diese konnten wir jedoch mit einem Change Request gut abfedern.

Auch dass Claudio Fuchs bereits Betreuer der Studienarbeit war, hat meiner Meinung das Projekt positiv beeinflusst. Ich wusste bereits was ungefähr gefordert wird und welche Punkte das bei der Umsetzung wichtig sind. Dadurch mussten wir auch weniger Nachfragen und Abklären. Ich glaube, dass wir die Applikation so im Sinne von Claudio Fuchs umsetzen konnten.

An dieser Stelle muss erwähnt werden, dass dieses Projekt mitten in der Corona Pande-

mie umgesetzt wurde. Dies hat uns vor einige Herausforderungen gestellt. Einerseits was die Zusammenarbeit an der Schule betrifft, andererseits auch die Livetests sowie Präsentationen welche davon betroffen waren. Die Kommunikation in unserem Team hat trotz allem hervorragend funktioniert. Leider bietet Microsoft Teams keine Möglichkeit die totale Zeit auszulesen, welche wir zusammen telefoniert hatten, es kämen jedoch einige Stunden zusammen. Wir hatten jedoch das grosse Glück, dass wir beide im selben Dorf wohnen und so vor Ort miteinander Arbeiten konnten. Dies ist doch um einiges effizienter und angenehmer als nur Remote zu Arbeiten.

Wie bereits erwähnt, war für mich die Überarbeitung von „PM Game“ sehr wichtig. Die Trennung von Frontend und Backend hat sich meiner Meinung nach ausgezahlt. Wir konnten die Applikation dadurch wesentlich interaktiver umsetzen, davon hat meiner Meinung nach die User Experience stark profitiert. Die neue Version ist UI technisch auf einem ganz anderen Stand als die erste Version.

Die Live Tests hatten sich auch in diesem Projekt wieder als sehr wertvoll erwiesen. Wir haben noch einige Verbesserungsmöglichkeiten sowie Bugs gefunden, welche wir in unseren Lokalen Tests nicht entdecken konnten. Auch mit der Stabilität der Applikation bin ich zufrieden. In der Studienarbeit hatten wir beim zweiten Livetest grosse Probleme, weil ein Build nicht korrekt gestartet werden konnte. Dies war an diesem Tag nicht nur äusserts Mühsam, sondern hat auch erhebliche Mehrarbeit mit sich gebracht. Um diesem Problem zuvor zukommen, haben wir in dieser Arbeit die Build Pipeline anders aufgebaut. Auch haben wir die Planung des Projekts dahingehend geändert, dass solch ein Desaster schnell behoben werden kann.

- Wir hatten regelmässig auf den Master Branch gemerget. Diesen Masterbranch konnten wir stetig auf unserer Staging Umgebung prüfen.
- Wir haben in der Pipeline neu mit Releases gearbeitet. Selbst wenn ein Release korrupt sein sollte, hätten wir innerhalb weniger Minuten das letzte funktionierende Release starten können.
- Wir haben die Umsetzung der Applikation so geplant, dass ab den Livetests keine Änderung am Datenbankschema mehr vorgenommen werden muss. Funktionen welche keine Änderung am Schema erfordern, haben wir daher eher nach hinten geplant, damit diese ggf. in der Testing Phase noch umgesetzt werden können.

Wir hatten in jedem Livetest mit kleineren Kinderkrankheiten zu kämpfen. Es war uns jedoch in allen Fällen möglich, direkt auf der Applikation zu reagieren und so dennoch eine vollständige Durchführung der Simulation zu ermöglichen. Das Feedback der Studenten, des Betreuers sowie die entdeckten Bugs haben wir in der Zeit zum nächsten Livetest direkt umgesetzt. Dadurch konnte die Applikation in dieser Zeit nochmals stark verbessert werden. Ich würde daher auch bei künftigen Projekten, falls möglich, eine Pilotphase einplanen. Diese Livetests haben uns gezeigt, dass trotz guter Vorbereitung, einige Schwächen erst bei einer intensiveren Nutzung der Software ans Tageslicht kommen.

Das generelle Testing haben wir speziell im Backend wesentlich intensiver durchgeführt als damals bei der Studienarbeit. Die Logik der Applikation konnten wir fast gänzlich mit Unit Tests abdecken. Diese Tests wurden im Verlaufe des Projekts immer hilfreicher, da uns ein verändertes Verhalten der Applikation sofort angezeigt wurde. Im Frontend haben wir auf Unit Tests verzichtet, da keine grosse Logik im Frontend enthalten ist. Stattdessen haben wir manuelle Integrationstests bei jedem Merge auf den Master durchgeführt. Dies war zwar aufwendiger, hat den Zweck aber erfüllt. Spannend wären auch automatische End to End Tests in diesem Zusammenhang gewesen, hier hat uns aber die Erfahrung gefehlt um entsprechende Tests in einer adäquaten Zeit in unser Projekt zu integrieren.

Einige Stolpersteine sind während des Projekts auch aufgetreten.

**Verschiedenen Datenbanktechnologien** Während der Entwicklung hatten wir lokal SQLite verwendet, auf der produktiven Umgebung hingegen MariaDB. In der zweiten Woche der Livetests hatten wir dann Probleme mit dem Change Requests. Ursache war, dass wir in TypeORM keine Länge für die Felder der Change Requests angegeben hatten. MariaDB unterstützt bei Textfeldern standardmässig eine Länge von 255 Zeichen, SQLite hingegen hat keine standardmässige Begrenzung. Daher konnten während den Livetests am Anfang keine Change Requests gespeichert werden, da die meisten mehr als 255 Zeichen aufwiesen. Wir konnten diesen Umstand auf der Datenbank jedoch korrigieren und anschliessend auch im TypeORM den Wert setzen. Nach diesem Vorfall haben wir auch auf unseren lokalen Entwicklungsumgebungen MariaDB in einem Docker Container verwendet.

**Komplexität der Funktionen** In diesem Projekt haben wir bereits im Vorhinein auf die Qualität des Codes geachtet. Dennoch sind z. B. einige Funktionen komplexer geworden als empfohlen.

**Git Probleme** Wir hatten diverse Male, dass uns nach dem Mergen von einem Branch ein Teil des Codes gefehlt hat. Grundsätzlich erkennt Git automatisch, welche Teile des Codes gemerget werden können und welche z. B. Konflikte verursachen. In einigen Fällen im Frontend hat dies jedoch nicht geklappt und es wurden wieder Änderungen durch einen anderen Branch rückgängig gemacht. Wir haben daher angefangen nach einem Merge den Code nochmals genau zu überprüfen.

Im Grossen und Ganzen bin ich jedoch zufrieden mit der Arbeit. Die Probleme, welche während des Projekts auftraten, waren nie wirklich schwerwiegend. Die Verzögerung in der Test Phase konnten wir durch die Verlängerung des Projekts gut abfedern und hatten so dennoch genug Zeit für die Dokumentation. Die Zusammenarbeit mit allen Beteiligten des Projekts war jederzeit sehr angenehm und konstruktiv. Die definierten Ziele der Arbeit konnten wir allesamt erreichen.

## 7.2. Dario Caluzi

Wir konnten direkt sehr gut in die Bachelorarbeit starten. Da Martin bereits die Vorgängerversion unserer Arbeit mitentwickelt hat, hatten wir den Vorteil, dass wir die Verbesserungsmöglichkeiten bereits kannten. Deshalb haben wir auch sofort damit begonnen das Frontend und Backend aufzuteilen, sowie das Frontend mit VUE umzusetzen. Dadurch konnte ich auch die Applikation sehr gut kennenlernen und direkt meinen Input und Wünsche für unsere Version dazugeben. So konnten wir auch gleich die User Storys, welche vom Betreuer vorgegeben und aus der vorgängigen Arbeit übernommen wurden, ergänzen.

Die Entscheidung, das Projekt nach dem Wasserfallmodell zu organisieren hat sich sehr gelohnt. Somit hatten wir weniger administrativen Aufwand als bei vorherigen Arbeiten nach dem Scrum Modell. Trotz der ungewohnten Lage mit COVID-19 konnten wir uns im Team sehr gut organisieren und konnten auch mit dem Betreuer bei Bedarf in kurzer Zeit ein Meeting ansetzen. Der grösste Teil unserer Kommunikation im Team fand über Teams Meetings statt, was aber gut funktioniert hat. Nichtsdestotrotz hat sich gezeigt, dass die Kommunikation, wenn man zusammen an einem Ort ist, doch effizienter ist. Dies zeigte sich vor allem bei den Live Tests. Da mussten wir beide konstant den Studenten zur Verfügung stehen, damit wir Inputs aufnehmen konnten und bei Problemen unterstützten. Gleichzeitig mussten wir uns aber auch im Team absprechen wer, was macht, sowie dem Betreuer für Rückfragen zur Verfügung stehen. Deshalb war die optimale Lösung, dass ich und Martin uns trafen und jeweils den Übungsschat überwachten, wo die Studenten sich schriftlich meldeten bei Fragen und Anmerkungen, danach konnten wir uns jeweils telefonisch über einen privaten Teams Chat mit den jeweiligen Studenten austauschen. Mir ist es jedoch schon schwerer gefallen als wenn man sich persönlich trifft.

Wir konnten sehr rasch mit dem Entwickeln beginnen, nachdem der Projektplan stand. Ein grosser Teil davon ist natürlich, dass wir bereits zu Beginn der Arbeit vom Betreuer gewisse funktionale sowie auch nicht-funktionale Anforderungen bekommen haben. Bei früheren Arbeiten musste zuerst jeweils viel Zeit in die Ausarbeitung dieser investiert werden. Das war für mich eher einen Einblick in den Ablauf mit einem Kunden, welchen einen spezifischen Wunsch hat. Dies erfordert etwas weniger Aufwand, da man auch mit dem Kunden (in unserem Fall den Betreuer) Rücksprachen halten kann, ob die Umsetzung gut ist. Bei einer Arbeit mit Machbarkeitsstudie muss man sich zuerst in die Theorie einlesen und gut mit der Thematik befassen, um sich dann zu entscheiden, was man genau umsetzen möchte. Aus meiner Sicht ist der Lerneffekt bei einer Arbeit mit Machbarkeitsstudie höher als beispielsweise bei unserer Bachelorarbeit „Manager Game weiterentwickeln“. Im Gegensatz dazu ist der Lieferumfang der Applikation, was der Hauptteil der Arbeit ist, wesentlich grösser als wenn man zuerst eine Machbarkeitsstudie machen muss. Da finde ich es sehr schön zu sehen, was wir in dieser relativ kurzen Zeit alles umsetzen konnten im „PM Game“.

Bei der Umsetzung konnte ich sehr schnell produktiv mitarbeiten, nachdem ich das

Setup etwas kennengelernt hatte. Die Technologien wie TypeScript und VUE kannte ich bereits. Ich bin sehr froh, dass wir das typischere TypeScript verwendet haben, denn ich persönlich finde so so kann man bereits einige Vorteile gegenüber normalem JavaScript herausholen. Persönlich bevorzuge ich jedoch trotzdem andere Programmiersprachen, wie z. B. Python mit Flask<sup>33</sup> um application programming interface (API) Endpunkte zur Verfügung zu stellen.

Zu Beginn brauchte ich etwas Zeit mich an den OR Mapper TypeORM zu gewöhnen. Vor allem das Umdenken von einer rationalen Datenbank zum OR Mapper war etwas ungewohnt. Im speziellen die Abfragen, welche einen Join von mehreren Tabellen benötigen, waren mir nicht ganz klar. Mit der Unterstützung von Martin, der da schon mehr Erfahrungen hat, war die Umstellung jedoch schnell gemeistert.

Das ganze Setup mit den Technologien und Tools, welche wir verwendet haben, hat sehr gut funktioniert. Es gab dabei nur eine Ausnahme, wo uns aber bis heute nicht ganz klar ist, was das Problem war. Bei gewissen Merge Requests im Frontend hat git aus einem unbekanntem Grund teilweise gewisse Änderungen am Code nicht richtig erkannt und keine Merge Conflicts generiert. Dadurch hatten wir nicht die Möglichkeit, manuell die korrekten Änderungen auszusuchen, sondern git hat selbständig für uns entschieden. Leider hat git uns einige Mal einen kleinen Strich durch die Rechnung gemacht und wir mussten im Nachhinein mit grossem Schock feststellen, dass gewisse Änderungen fehlten. Da die Änderungen aber jeweils in den entsprechenden Branches vorhanden waren, konnten wir diese wieder holen und mussten sie neu in die Applikation eintragen. So mussten wir uns damit abfinden, dass wir im Frontend jeweils nach dem Mergen uns den Code nochmals genau ansehen, um zu kontrollieren, ob alle Änderungen korrekt vorhanden sind.

Die Live Tests waren sehr hilfreich und haben einmal mehr gezeigt, dass ein solcher Test mit richtigen Leuten und normaler Benutzung durch kein Tool ersetzbar ist. Einige Fehler sind erst bei der Verwendung im Test aufgetreten und uns trotz eingesetzter Test Tools durch die Lappen gegangen. Doch den Studenten entging nichts. Einige haben sich sogar die Zeit genommen die Webapp durch Absetzen eigener Requests mit extrahiertem Access Token zu testen. Die Bugs und Verbesserungsvorschläge die wir dadurch erhalten haben, hatten eine super Qualität. Das kann man auch daran erkennen, dass wir uns entschieden haben einen Change Request zu schreiben um mehr davon umzusetzen und dafür eine der Anforderungen (Dem Spiel per Token beizutreten) aus den User Storys dafür zurückzustellen. Der Nachmittag des ersten Live Test war jedoch sicherlich mitunter die intensivste und anstrengendste Zeit der Arbeit. Da mussten wir sehr schnell reagieren und gewisse Probleme sofort lösen (z. B. wurden gewisse Daten doppelt abgespeichert, diese mussten live in der Datenbank bereinigt werden) und von allen Studenten die Inputs aufnehmen, dass nichts untergeht. Bei den darauffolgenden Live Tests kamen nicht mehr ganz so viele Rückmeldungen gleichzeitig und hinzu kommt, dass ich und Martin auch schon besser darauf vorbereitet waren, da wir aus dem ersten Test gelernt haben.

---

<sup>33</sup><https://flask.palletsprojects.com/>

Aus meiner Sicht ist die Arbeit gut gelungen. Ich finde wir haben mit der zweiten Version von „PM Game“ einen guten Schritt nach Vorne gemacht mit dem Funktionsumfang, sowie der Usability. Es ist auch schön zu sehen, dass bei den Befragungen der Studenten, die Mehrheit angegeben hat, dass sie einen Lernvorteil daraus gewinnen konnten. Mir persönlich gefällt es auch, dass die Applikation im Unterricht für PmQm verwendet wird. Somit wird unsere Arbeit auch in Zukunft verwendet oder bildet eine gute Basis für die Weiterentwicklung durch andere Studenten.

## Glossar

- Change Request** Ein formalisierter Änderungswunsch im Change Management eines Projekts, nachdem die Anforderungen verbindlich festgelegt wurden. vi
- CI/CD** In der Softwareentwicklung steht CI/CD für „continuous integration“ und „continuous deployment“. CI/CD überbrückt die Lücken zwischen Entwicklung und dem Betrieb einer Applikation, indem es die Automatisierung beim Erstellen, Testen und Bereitstellen von Anwendungen erzwingt. 12
- Docker** Eine Softwareumgebung um mit Containern zu arbeiten. Container gewährleisten die Trennung und Verwaltung der auf einem Rechner genutzten Ressourcen. 59
- Function Points** Ein Wert welcher die Grösse einer Software weitgehend objektiv und reproduzierbar quantifiziert. 7
- Gantt Diagramm** Ein Balkendiagramm, welches die zeitliche Abfolge von Aktivitäten auf einer horizontalen Zeitachse darstellt. 29
- MariaDB** Ein relationales Datenbankmanagementsystem, welches aus einer Abspaltung von MySQL entstanden ist. 59
- OR-Mapper** Bildet ein objektorientiertes Datenschema auf eine relationale Datenbank ab. vi
- SLOC** Source lines of code ist eine Massangabe für die Grösse eines Softwareprojekts und umfasst die Anzahl der Zeilen des Quelltextes, aus denen das Programm besteht. 100
- SQL Injection** Eine Angriffsmethode bei der ein Angreifer versucht über die Applikation eigene SQL Befehle abzusetzen. 31
- Staging Umgebung** Eine Umgebung unter der die Applikation unter realen Bedingungen getestet werden kann. 12
- Traefik** Ein Open Source Reverse Proxy und Load Balancer. 59
- Use Cases** Beschreibt formell wie ein Akteur in einer Software ein bestimmtes Ziel erreichen kann. Es werden mögliche Abläufe und alternative Szenarien festgehalten. 42
- User Storys** Eine User Story ist eine in Alltagssprache formulierte Software-Anforderung. Sie ist bewusst kurz gehalten und umfasst in der Regel nicht mehr als zwei Sätze. v

**XSS** Cross-Site-Scripting (XSS; deutsch Webseitenübergreifendes Skripting) bezeichnet das Einschleusen z.B. eines Javascripts welches bei jedem Aufrufer der Seite ausgeführt wird. Dieses Javascript kann dann z.B Benutzerdaten abgreifen. 31

## **Akronyme**

**API** application programming interface. vi, 8, 122

**GUI** graphical user interface. 11

**IDE** Integrated Development Environment. 86

**OST** Ostschweizer Fachhochschule. iv, v, 15, 35

**PmQm** Projekt- und Qualitätsmanagement. v, 17, 123

## Literatur

- [1] B. Spinath, *Lernmotivation*, pp. 45–55. Wiesbaden: VS Verlag für Sozialwissenschaften, 2011.
- [2] J. Hattie, *Lernen sichtbar machen : überarbeitete deutschsprachige Ausgabe von Visible Learning*. Baltmannsweiler: Schneider Hohengehren, 4. erweiterte auflage mit index und glossar ed.
- [3] S. H. Josef Schrader, Reinhard Hohmann, *Mediengestützte Fallarbeit - Konzepte, Erfahrungen und Befunde zur Kompetenzentwicklung von Erwachsenenbildnern*. wbv Media.
- [4] U.-D. Ehlers, “Qualität beim e-learning: Der lernende als grundkategorie bei der qualitätssicherung,” *MedienPädagogik: Zeitschrift für Theorie und Praxis der Medienbildung*, vol. 5, pp. 1–20, März 2002.
- [5] A. Elliot and J. Harackiewicz, “Approach and avoidance achievement goals and intrinsic motivation: A mediational analysis,” *Journal of Personality and Social Psychology*, vol. 70, p. 461, 03 1996.
- [6] M. Iten and M. Keller, “Project manager.” December 2019.
- [7] T. Everts, *Time is money : the business value of web performance*. Beijing, [China: O’Reilly Media, first edition. ed.
- [8] C. Wigger, “Auswirkungen von blended-learning auf studierende und hochschulen - eine felduntersuchung.”

# Abbildungsverzeichnis

1.	Auswertung Frage 1 . . . . .	14
2.	Auswertung Frage 2 . . . . .	15
3.	Auswertung Frage 3 . . . . .	16
4.	Vergleich der drei Lernmethoden . . . . .	16
5.	Auswertung Frage 4 . . . . .	19
6.	Auswertung Frage 5 . . . . .	19
7.	Auswertung Frage 6 . . . . .	19
8.	Auswertung Frage 7 . . . . .	20
9.	Auswertung Frage 8 . . . . .	21
10.	Auswertung Frage 9 . . . . .	21
11.	Task zuteilung . . . . .	27
12.	Reduktion eines Risikos bei Simultrain . . . . .	28
13.	Auswertung der Qualität der geleisteten Arbeit bei Cesim Project . . . . .	28
14.	Auswertung Frage 12 . . . . .	33
15.	Auswertung Frage 13 . . . . .	34
16.	Auswertung Frage 14 . . . . .	35
17.	Auswertung Frage 15 . . . . .	36
18.	Auswertung Frage 15 . . . . .	37
19.	SWOT Analyse „PM Game“ . . . . .	38
20.	List of User Storys . . . . .	41
21.	List of Use Cases . . . . .	42
22.	Use Case Diagram . . . . .	43
23.	Domain Model . . . . .	52
24.	Schichten Diagramm . . . . .	57
25.	Deployment-Diagramm . . . . .	58
26.	Übersicht Game . . . . .	59
27.	Übersicht Admin . . . . .	60
28.	Spiel - Projektleiter Management . . . . .	61
29.	Spiel - Mitarbeiter Management . . . . .	62
30.	Black-Box Test Ergebnisse . . . . .	66
31.	Coverage aller Files . . . . .	67
32.	Coverage der Middleware . . . . .	67
33.	Coverage aller Controllers . . . . .	68
34.	Ergebnisse der Tests zu den nicht-funktionalen Anforderungen . . . . .	69
35.	Detaillierte Ergebnisse der Load Tests . . . . .	72
36.	Response Zeiten und deren Prozentsätze über die Zeit . . . . .	73
37.	Requests über Zeit . . . . .	73
38.	Responses über Zeit . . . . .	74
39.	Frontend Pipeline . . . . .	77
40.	Backend Pipeline . . . . .	77
41.	Spielansicht: Dashboard . . . . .	92
42.	Spielansicht: Mitarbeiter Management . . . . .	93

43.	Spielansicht: Projektleiter Management . . . . .	94
44.	Spielansicht: Risiko Management . . . . .	95
45.	Spielansicht: Change Request Eingabe . . . . .	96
46.	Adminbereich: Dashboard . . . . .	97
47.	Adminbereich: Change Request Kontrolle . . . . .	98
48.	Code Komplexität Seite 1 . . . . .	100
49.	Code Komplexität Seite 2 . . . . .	101
50.	Code Komplexität Seite 3 . . . . .	101
51.	Codacy Auswertung Frontend . . . . .	102
52.	Codacy Auswertung Backend . . . . .	103
53.	Projektausschreibung . . . . .	104
54.	Zeitplan - Seite 1 . . . . .	107
55.	Zeitplan - Seite 2 . . . . .	108
56.	Zeitplan - Seite 3 . . . . .	109
57.	Zeitplan - Seite 4 . . . . .	110
58.	Meilenstein Trend Analyse . . . . .	111
59.	Zeitauswertung nach Stunden . . . . .	116
60.	Zeitauswertung nach Issues . . . . .	116

## Tabellenverzeichnis

1.	Changelog . . . . .	ii
2.	Vergleich zwischen „PM Game“ Version 1.0 und 2.0 . . . . .	11
3.	Auswertung des Feedbacks „keinen Lernvorteil erhalten“ . . . . .	22
4.	Auswertung des Feedbacks „konnte Lernvorteil erzielen“ . . . . .	23
6.	Issues welche während den Livetests erfasst wurden. . . . .	70

# A. Projektplan

## Projekt-Plan „Project Manager“

HS20 OST Ostschweizer Fachhochschule

Dario Caluzi Martin Iten

23. September 2020

Version 1

### Inhaltsverzeichnis

<b>1. Änderungsverlauf</b>	<b>3</b>
<b>2. Einführung</b>	<b>4</b>
2.1. Zweck . . . . .	4
2.2. Gültigkeitsbereich . . . . .	4
<b>3. Projektübersicht</b>	<b>5</b>
3.1. Zweck Project Manager . . . . .	5
3.2. Aufgabenstellung . . . . .	5
3.3. Ziel . . . . .	5
3.4. Lieferumfang . . . . .	5
<b>4. Projektorganisation</b>	<b>7</b>
4.1. Übersicht . . . . .	7
4.2. Team . . . . .	7
4.3. Externe Schnittstellen . . . . .	7
<b>5. Managementabläufe</b>	<b>8</b>
5.1. Kostenvoranschlag . . . . .	8
5.2. Meilensteine . . . . .	8
5.3. Zeitliche Planung . . . . .	8
5.4. Besprechungen . . . . .	9
<b>6. Risikomanagement</b>	<b>10</b>
6.1. Risiken . . . . .	10
6.2. Umgang mit Risiken . . . . .	10

<b>7. Infrastruktur</b>	<b>11</b>
7.1. Gitlab	11
7.1.1. CI/CD Pipeline	11
<b>8. Qualitätsmassnahmen</b>	<b>12</b>
8.1. Definition of Done („Definition Fertigstellungsgrad der Arbeitspakete“)	12
8.2. Generelle Qualitätsmassnahmen	12
8.3. Code-Qualität	13
8.4. Testing	13
<b>A. BA Vereinbarung</b>	<b>14</b>
<b>B. BA Projektplan</b>	<b>16</b>

## 1. Änderungsverlauf

Datum	Version	Änderung	Autor
2020-09-20	1	Initiale Version des Dokumentes	Dario Caluzi, Martin Iten
2020-09-23	2	Überarbeitung	Dario Caluzi, Martin Iten

Tabelle 1: Changelog

## **2. Einführung**

### **2.1. Zweck**

Dieser Projektplan bildet die Grundlage für die Bachelorarbeit „Project Manager“.

### **2.2. Gültigkeitsbereich**

Dieses Dokument gilt in seiner ersten Phase als Grundlage für die Studienarbeit „Project Manager“. Es gilt jeweils die aktuellste Version des Dokuments.

### **3. Projektübersicht**

„Project Manager“ ist ein Spiel zum Thema Projektmanagement, welches im Rahmen einer Studienarbeit an der HSR als software-basierte Lösung erstellt wurde. In dieser Bachelor-Arbeit wird dieses Projekt nun weitergeführt. Einerseits werden Verbesserungen an der aktuellen Software vorgenommen und andererseits neue Funktionen hinzugefügt. Neben dieser praktischen Umsetzung wird auch eine theoretische und betriebswirtschaftliche Fragestellung untersucht.

#### **3.1. Zweck Project Manager**

Mit dem „Project Manager“ soll den im OST-Modul „Projekt- und Qualitätsmanagement“ erlernte Stoff spielerisch vertieft werden.

#### **3.2. Aufgabenstellung**

1. Verschafft der Einsatz einer Software in den Übungsstunden einen tatsächlichen Wissensaufbau, da sich die Studenten damit auseinandersetzen müssen?
2. Können die Anforderungen des Dozenten in einer Software umgesetzt werden? Begründung, wenn Anforderungen nicht wie gefordert umgesetzt werden konnten.
3. Was wäre notwendig, um diese Software aus Sicht Ostschweizer Fachhochschule als Trainingsdienstleistung für Externe anzubieten?

Die komplette Aufgabenstellung wurde diesem Projektplan als Anhang A beigelegt.

#### **3.3. Ziel**

Wir setzen uns als Projektteam folgende Ziele:

1. Bestehen der Bachelor-Arbeit.
2. Untersuchung und Beantwortung der theoretischen Fragestellung.
3. Untersuchung und Beantwortung der betriebswirtschaftlichen Fragestellung.
4. Saubere Anforderungsanalyse und solides Architekturdesign. Alle Dokumentationsrichtlinien der OST einhalten.
5. Bestehende Software „Project Manager“ erweitern und verbessern.

#### **3.4. Lieferumfang**

Am Ende des Projekts werden folgende Arbeitsprodukte abgegeben:

- Source Code
- Projektdokumentation gemäss OST-Richtlinien<sup>1</sup>
- Lauffähige Software auf dem OST-Server.

---

<sup>1</sup>Siehe dazu das Dokument [Anleitung: Dokumentation Studien- und Bachelorarbeiten](#). Dieser Link ist nur innerhalb des OST-Netzwerks erreichbar.

## **4. Projektorganisation**

### **4.1. Übersicht**

Der Austausch zwischen den einzelnen Projektmitgliedern ist sehr wichtig. Durch regelmässigen Austausch möchten wir verhindern, dass wir uns in den Weg kommen und Leerläufe entstehen.

Leider wird durch die aktuelle Situation mit Covid-19 und den damit verbundenen Vorschriften, die Möglichkeit vor Ort Meetings abzuhalten teilweise eingeschränkt. Daher werden Werkzeuge wie Microsoft Teams und Sharepoint einen grossen Stellenwert in diesem Projekt haben.

### **4.2. Team**

Das Team besteht aus Dario Caluzi und Martin Iten. Aufgrund der geringen Teamgrösse sowie dem breiten Umfang des Projekts verzichten wir auf die Verteilung konkreter Rollen sowie Verantwortlichkeiten.

- Dario Caluzi: [dario.caluzi@ost.ch](mailto:dario.caluzi@ost.ch)
- Martin Iten: [martin.iten@ost.ch](mailto:martin.iten@ost.ch)

### **4.3. Externe Schnittstellen**

- **Projektbetreuer:** Claudio Fuchs, [claudio.fuchs@ost.ch](mailto:claudio.fuchs@ost.ch)
- **Serverbetreuung:** ServiceDesk der Informatikdienste HSR Hochschule für Technik Rapperswil, [servicedesk-rj@ost.ch](mailto:servicedesk-rj@ost.ch)

## 5. Managementabläufe

Für dieses Projekt setzen wir auf das klassische Wasserfall Modell. Wir haben uns gegen ein agiles Vorgehen entschieden, da der Zeithorizont sowie der Umfang der Arbeit gut abgesteckt sind. Dies erlaubt uns eine solide Planung im Vorhinein sowie eine effiziente Umsetzung der Features während der Programmier-Phase.

### 5.1. Kostenvoranschlag

Für das Projekt stehen insgesamt 14 Wochen zur Verfügung. Jeder Student erhält 12 European Credit Transfer System-Punkte (ECTS) für die Studienarbeit. 1 ECTS entspricht einem Arbeitsaufwand von 30 Stunden<sup>2</sup> Total stehen somit 720 Arbeitsstunden zur Verfügung:

$$2 \text{ Projektmitglieder} \cdot 30 \text{ Stunden} \cdot 12 \text{ ECTS} = 720 \text{ Arbeitsstunden}$$

### 5.2. Meilensteine

Folgende Meilensteine wurden definiert:

- Anforderungsanalyse
- Systemdesign
- Programmierung
- Testing
- Auslieferung

### 5.3. Zeitliche Planung

Für die zeitliche Planung wurden die definierten Meilensteine als Eckpfeiler verwendet. Ab der Woche 40 soll die Umsetzung der Arbeitspakete erfolgen, so dass ab dem Meilenstein „Testing“ nur noch Verbesserungen aufgrund des erfolgten Feedbacks vorgenommen werden müssen. Dies soll ein stabiler Betrieb während der Testing-Phase garantieren.

Ein stetig aktueller Plan wird über die Share-Point Umgebung zur Verfügung gestellt und kann [hier](#)<sup>3</sup> eingesehen werden. Zusätzlich ist diesem Dokument als Anhang B der Plan vom Stand 21.09.2020 beigefügt.

<sup>2</sup>siehe OST-Modulbeschreibung Bachelor-Arbeit Informatik [https://studien.rj.ost.ch/allModules/public/24809\\_M\\_BAI14.html](https://studien.rj.ost.ch/allModules/public/24809_M_BAI14.html).

<sup>3</sup>[https://ostch.sharepoint.com/:x:/t/TS-PMGAmE/Eall5ht0o8hKjSxHCe\\_5SekBFJIiwC-5vMTTdttdNqiC4yw?e=EYBTAA](https://ostch.sharepoint.com/:x:/t/TS-PMGAmE/Eall5ht0o8hKjSxHCe_5SekBFJIiwC-5vMTTdttdNqiC4yw?e=EYBTAA)

#### **5.4. Besprechungen**

Alle zwei Wochen wird mit Claudio Fuchs als Projektbetreuer eine Besprechung abgehalten. Diese erfolgen entweder vor Ort an der OST oder per Remote-Sitzung über Microsoft Teams. Die genaue Festlegung der Termine erfolgt direkt mit dem Betreuer.

## 6. Risikomanagement

### 6.1. Risiken

Das Risikomanagement kann [hier](#)<sup>4</sup> in der Share-Point Ablage eingesehen werden.

### 6.2. Umgang mit Risiken

- Risiken werden vorgängig abgeschätzt und mit vorbeugenden Massnahmen dokumentiert. Ebenfalls wird ein mögliches Vorgehen beschrieben, sollte ein Risiko eintreten.
- Die Risiken werden alle zwei Wochen neu bewertet und wenn nötig aktualisiert.
- Für mögliche Risikopakete wird jeweils mehr Zeit eingeplant.
- Risikobehaftete Arbeitspakete werden mit einem Risikolabel versehen.
- Ob ein Arbeitspaket das Risikolabel zugewiesen bekommt wird im Plenum entschieden.

---

<sup>4</sup><https://ostch.sharepoint.com/:x:/t/TS-PMGAmE/EXWdMEwQcZ9PtQa8Jci7Y10Bs5hPZUVtLdobsgEsxGnrZw?e=ALtLq>

## 7. Infrastruktur

Folgende Werkzeuge setzen wir für die Umsetzung des Projekts ein.

- Gitlab
  - Versionsverwaltung
  - Continuous Integration
  - Deployment
  - Verwaltung der Arbeitspakete
- Office 365
  - Sharepoint für die Datenablage und Koordination
  - Word Online für die Protokollierungen der Sitzungen
  - Teams für Remote-Sitzungen
- TMetric
  - Zeiterfassung
  - Kontrolle und Auswertung der erfassten Zeiten
- Dedizierter Server OST
  - Betrieb der Software
- JetBrains Webstorm
  - Entwicklungsumgebung für Frontend sowie Backend

### 7.1. Gitlab

Gitlab spielt in diesem Projekt eine zentrale Rolle, denn darüber wird unser Code gehostet, die Arbeitspakete verwaltet sowie die Erstellung von Docker-Container für das Deployment durchgeführt. Wir haben uns für die öffentliche Version von GitLab entschieden, da diese im Gegensatz zur OST Internen GitLab Version weniger Port-Restriktionen unterworfen ist. Die Repositories sind privat und können nur durch die Teammitglieder eingesehen und bearbeitet werden.

#### 7.1.1. CI/CD Pipeline

Wir haben uns dazu entschieden die CI/CD Pipeline ebenfalls über GitLab zu betreiben. Diese Plattform hat sich bereits in der Studienarbeit bewährt. Zudem haben wir uns für Docker entschieden, da dadurch alle Mitglieder ohne aufwändige Infrastruktur an allen Teilen des Projekts arbeiten können. Auch eignet es sich gut zum Deployen. Dadurch ist es einfach auf andere Cloud Provider portierbar.

## 8. Qualitätsmassnahmen

Dieser Abschnitt erklärt, wie wir die Arbeitsqualität für die gesamte Dauer des Projekts sicherstellen möchten. Die Qualitätsanforderungen an die Software werden in diesem Abschnitt jedoch explizit **nicht** abgedeckt, denn diese werden in den nicht-Funktionalen Anforderungen beschrieben.

### 8.1. Definition of Done („Definition Fertigstellungsgrad der Arbeitspakete“)

Richtlinien für Arbeitspakete mit dem Label „Development“:

- Funktionalität ist vollständig implementiert.
- Die Dokumentation wurde ergänzt falls nötig.
- Die Style-Guidelines wurden eingehalten.
- Unit Tests wurden erstellt und laufen erfolgreich durch.

Wenn diese Vorgaben erfüllt sind, kann das Arbeitspakete nach Review verschoben werden.

Richtlinien für Arbeitspakete mit dem Label „Review“:

- Der Code wurde durch ein anderes Projektmitglied geprüft.

Wurden diese Vorgaben erfüllt, kann ein Arbeitspaket geschlossen werden.

Richtlinien für Arbeitspakete ohne eines dieser Label:

- Das Arbeitspaket wurde gemeinsam besprochen.
- Alle Projektmitglieder sind mit der Schliessung des Arbeitspakets einverstanden.

### 8.2. Generelle Qualitätsmassnahmen

- Regelmässige Besprechungen innerhalb des Teams sowie mit dem Projektbetreuer stellen sicher, dass der aktuelle Fortschritt wie auch Rückschläge zeitnahe ausgetauscht werden.
- Informationen werden über den zentralen Kommunikationskanal „Microsoft Teams“ ausgetauscht. Alle Projektmitglieder haben dadurch jederzeit Zugriff auf die Informationen.
- Wichtige Informationen von Gitlab, wie Merge Requests und Fehler bei der CI/CD werden automatisch als Nachricht im Teams Chat „Gitlab Notifications“ hinzugefügt, damit bei Bedarf schnell darauf reagiert werden kann.

- Alle Daten und Dokumente werden zentral auf dem Share-Point abgelegt. Alle Projektmitglieder haben dadurch jederzeit Zugriff auf die Informationen.
- Wir nehmen uns Zeit für das Code-Review eines Projektmitglieds.
- Arbeitspakete im GitLab können Checklisten enthalten um den Fortschritt eines Arbeitspakets zu dokumentieren.

### 8.3. Code-Qualität

- Für den Code verwenden wir verbindlich einen Code-Styleguide:
  - Vue Frontend** eslint-vue recommended rules
  - Backend** typescript-eslint recommended rules
- Wir führen Unittests und Integrationstests durch
  - Wir streben eine Test-Coverage von 65% an.
- Wir versuchen Kommentare im Code zu vermeiden, denn der Code sollte jeweils selbsterklärend sein.
- Wir verwenden Merge-Request Templates
- Wir verwenden das 4-Augen-Prinzip bei Merges
  - Wurde ein Arbeitspaket abgeschlossen, so wird das Review des Codes durch ein anderes Projektmitglied durchgeführt.
  - Der Ersteller eines Merge-Requests merged nie seinen eigenen Merge-Request.
  - Der Merge Button wird vom Reviewer als Zeichen eines durchgeführten Reviews gedrückt.
- Continuous Integration im GitLab stellt durch Linter und Tests sicher, dass gepushter Code den Code- und Testrichtlinien entspricht.

### 8.4. Testing

Folgende Tests werden im Verlauf des Projekts durchgeführt:

- Unit-Tests und Integration Tests
- Usability-Test: Live Testing der Applikation im PmQm Unterricht.

Für Tests welche nicht automatisiert durchgeführt werden können, werden konkrete Testspezifikationen sowie ein Testprotokoll erstellt.

## A. BA Vereinbarung



### Studienarbeit: Projekt-Game

<b>Einführung</b>	Im Zuge des Moduls Projekt- & Qualitätsmanagement soll das Erlernete spielerisch vertieft werden. Um das Interesse der Studierenden zu steigern und damit das Wissen nachhaltiger zu verankern, soll das Spiel den Charakter eines Team-Wettbewerbs haben.
<b>Aufgabe</b>	<p>Der Betreuer ist gleichzeitig der Dozent des Moduls und hat den Studierenden vorab eine Liste mit User Stories zur Verfügung gestellt. Die darin beschriebenen Anforderungen sollen in einer software-basierten Lösung als Spiel umgesetzt werden.</p> <p>Folgende (wissenschaftliche) Fragestellungen ergeben sich theoretisch, praktisch und betriebswirtschaftlich daraus:</p> <ol style="list-style-type: none"><li>1. Verschafft der Einsatz einer Software in den Übungsstunden einen tatsächlichen Wissensaufbau, da sich die Studierenden damit auseinandersetzen müssen?</li><li>2. Können die zusätzlichen funktionalen und nicht-funktionalen Anforderungen des Dozenten in einer Software umgesetzt werden? Begründung, wenn Anforderungen nicht wie gefordert umgesetzt werden konnten</li><li>3. Was wäre notwendig, um diese Software aus Sicht Ostschweizer Fachhochschule als Trainingsdienstleistung für Externe anzubieten?</li></ol>
<b>Hinweise</b>	Es besteht eine Version 1.0, welche den Studierenden bekannt sein sollte. In den Übungsstunden vom 12. November, 19. November und 26. November des Moduls PmQm besteht die Möglichkeit, eine Version des Games mit den Studierenden auszuprobieren.
<b>Erwartete Resultate</b>	<ul style="list-style-type: none"><li>• Erwartet wird die Beantwortung der obengenannten Fragestellung in Form eines Berichts. Die Software wird einerseits als implementierte Lösung abgegeben, andererseits aber auch in Form von Source-Code und Dokumentation.</li><li>• Die Auswahl der Vorgehensmethode ist den Studierenden überlassen. Die gewählte Methode muss jedoch zu Beginn kommuniziert und dann konsequent durchgezogen werden. Der Betreuer überwacht und bewertet die Effektivität und Einhaltung der Methode.</li><li>• Die investierten Arbeitsstunden müssen personen- und datumsbezogen dokumentiert werden. Der Rapport wird im Anhang des Berichts beigelegt.</li><li>• Es muss auf Basis des Stundenrapports oder der Kapitel im Bericht ersichtlich sein, wer an was gearbeitet hat.</li></ul>



Termine	Es wird alle zwei Wochen eine Zwischenbesprechung angestrebt. Die Abgabe erfolgt bis spätestens Donnerstag, 17. Dezember 2020 um 12.00 Uhr.
Betreuung	Betreuer: Claudio Fuchs

Im Sinne der Akzeptanz unterzeichnen die Beteiligten diese Aufgabenstellung gemeinsam.

Ort, Datum:

  
Dario Caluzi, Student

  
Martin Iten, Student

  
Claudio Fuchs, Betreuer

