



HSR

HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

FHO Fachhochschule Ostschweiz

Project Manager

Studienarbeit

ABTEILUNG INFORMATIK
HOCHSCHULE FÜR TECHNIK RAPPERSWIL

Herbstsemester 2019

Autoren: Martin Iten, Marco Keller

Betreuer: Claudio Fuchs

19. Dezember 2019

Änderungsverlauf

Datum	Version	Änderung	Autor
2019-10-16	1	Initiale Version des Dokumentes	Marco Keller, Martin Iten
2019-10-24	2	Bibliographie eingebunden; Einleitung geschrieben	Marco Keller, Martin Iten
2019-12-11	3	Struktur für Software Engineering- und Projektdokumentation erstellt gemäss HSR-Richtlinien	Marco Keller
2019-12-11	4	Einleitung sowie theoretischer Teil geschrieben	Martin Iten
2019-12-13	5	Dokumente aus Google Drive ins LaTeX übernommen	Marco Keller
2019-12-17	6	Dokumentation substantially complete	Marco Keller
2019-12-17	7	Dokumentation Layout überarbeitet	Martin Iten

Tabelle 1: Changelog

Abstract

„Project Manager“ ist eine Lernsimulation im Bereich IT-Projektmanagement. Sie richtet sich in erster Linie an Studenten des HSR-Moduls „Projekt- & Qualitätsmanagement“. Mit dem Project Manager können die in der Vorlesung erlernten Inhalte auf spielerische Weise an einem fiktiven IT-Projekt geübt werden.

Aus unserer Untersuchung zeigt sich, dass Lernspiele eine flexible und unabhängige Möglichkeit bieten, selbstständig oder in Gruppen zu lernen.

Als Ergebnis des Praxisteils entstand eine voll funktionsfähige Software inklusive Dokumentation. Von den erhobenen 40 User stories konnten 29 implementiert werden. Insbesondere konnten bis auf eine Ausnahme alle User stories mit der Priorität „MUSS“ umgesetzt werden. Die Alltagstauglichkeit der Software wurde mittels Live-Tests im Unterricht überprüft.

Des Weiteren untersuchten wir, ob die HSR diese Software im freien Markt als Trainingsdienstleistung für Unternehmen anbieten könnte. Mittels einer Konkurrenzanalyse fanden wir heraus, dass durchaus Potential besteht. Wir empfehlen aus heutiger Sicht, die Software in Form von Workshops anzubieten. Zudem empfehlen wir einen fachkundigen Spielleiter zur Verfügung zu stellen sowie den Detailgrad der Features noch zu erweitern.

Management Summary

Ausgangslage

Für die Vertiefung des Moduls „Projekt- & Qualitätsmanagement“ wird im Unterricht eine Managementsimulation verwendet. Bisher wurde diese weitestgehend offline mit einem Excel-Sheet durchgeführt. Im Rahmen dieser Studienarbeit wurde die Simulation als Applikation inklusive Dokumentation und Sourcecode umgesetzt. Als Grundlage dienten die bei der Projektausschreibung bereits konkret vorgegebenen Funktionen.

Zudem wurden in dieser Arbeit die beiden Fragen untersucht, ob es in der Literatur Hinweise gibt, ob spielerisches Lernen effizienter ist, und ob die Software auch im freien Markt als Trainingsdienstleistung für Unternehmen angeboten werden könnte.

Vorgehen und Technologie

Die theoretische Fragestellung haben wir mittels einer Literaturrecherche beantwortet. Zur Beantwortung der wirtschaftliche Fragestellung haben wir eine Konkurrenzanalyse durchgeführt und darauf basierend eine SWOT-Analyse erstellt.

Zur Erarbeitung der realisierten Software sind wir nach dem Vorgehensmodell Scrum+ vorgegangen. In der einwöchigen Inception-Phase verschafften wir uns zunächst einen Überblick über die Aufgabenstellung und den Umfang. In der dreiwöchigen Elaboration-Phase erstellten wir mehrere konzeptionelle Modelle, verfeinerten und erweiterten die vorgegebenen Funktionen. Ebenfalls spezifizierten wir die Qualitätsanforderungen an unsere Software.

Basierend auf den erhobenen Anforderungen haben wir die passenden Technologien gewählt. Die Software sollte auf möglichst vielen Endgeräten laufen. Entsprechend haben wir uns für eine Web-Client-Lösung entschieden. Dies bietet den Studenten einen einfachen Zugang zur Applikation und erlaubt dem Spielleiter eine einfache Verwaltung der Spiele.

In der achtwöchigen Construction-Phase setzten wir die Software gemäss den erhobenen Anforderungen um. Wir arbeiteten mit jeweils zweiwöchigen Sprints. In der knapp zweiwöchigen Transition-Phase lag der Fokus auf dem Zusammenstellen des Schlussberichts.

Ergebnisse und Ausblick

Aus dem Theorieteil zeigt sich, dass Lernspiele eine flexible und unabhängige Möglichkeit bieten, selbstständig oder in Gruppen zu lernen.

Als Ergebnis des Praxisteils entstand eine voll funktionsfähige Software inklusive Dokumentation. Wir konnten bis auf eine Ausnahme alle Funktionen, welche die Priorität

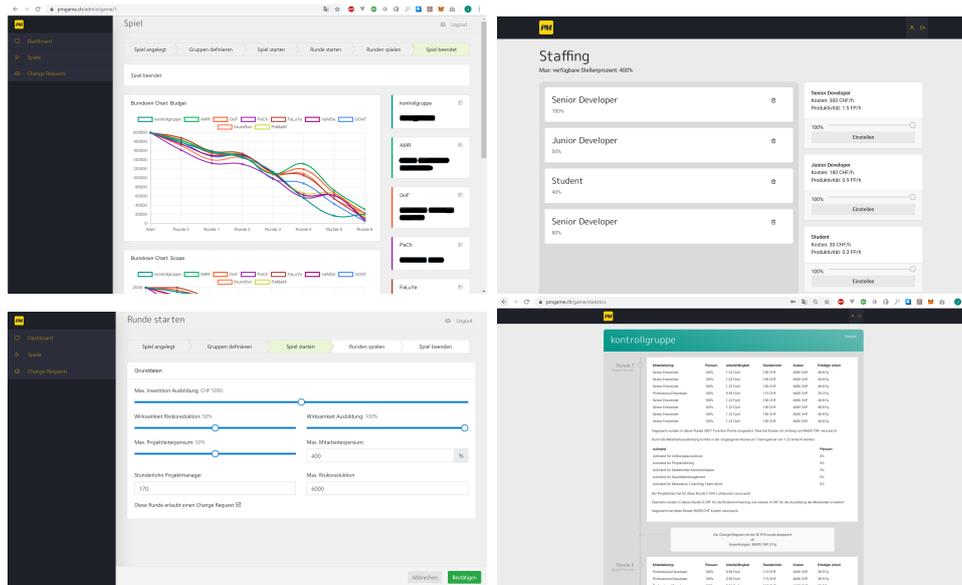


Abbildung 1: Screenshots aus „Project Manager“: Oben links ist der Burn-Down-Chart mit den Budgets der Gruppen zu sehen. Unten links ist die Eröffnung einer neuen Runde durch den Spielleiter zu sehen. Oben rechts ist die Spielersicht auf die Mitarbeiterplanung zu sehen. Unten rechts ist die Statistik der letzten Runde zu sehen.

„MUSS“ haben, vollständig einbauen.¹ Bei den nicht umgesetzten Funktionen handelt es sich vor allem um administrative Tasks und Notification-Features. Die Alltagstauglichkeit der Software wurde mittels Live-Tests im Unterricht überprüft. Es konnten erfolgreich mehrere Spiele durchgeführt werden.

Die Konkurrenzanalyse und die SWOT-Analyse in Abbildung 2 zeigen, dass durchaus Potential besteht, die Software im freien Markt als Trainingsdienstleistung anzubieten. Wir empfehlen aus heutiger Sicht, die Software in Form von Workshops anzubieten. Zudem empfehlen wir einen fachkundigen Spielleiter zur Verfügung zu stellen sowie den Detailgrad der Funktionen noch zu erweitern.

¹Die nicht umgesetzte Funktionalität kann jedoch hard-coded in der Datenbank gesetzt werden.

Strengths	Weaknesses
<ul style="list-style-type: none"> • Change Request-Feature: <ul style="list-style-type: none"> - hat im Live-Test den Austausch zwischen den Studierenden und dem Spielleiter stark gefördert. - hat auch den Austausch unter den Studierenden gefördert • flexibel ausbaubar • Event-Feature 	<ul style="list-style-type: none"> • Detailgrad beim Funktionsumfang ist kleiner als bei anderen Applikationen • Teilweise noch auf Zufall basierend (dies vor allem dadurch, da der Detailgrad der Funktionen klein ist und somit nicht alle Hintergrundberechnungen offengelegt werden können. Würden alle Berechnungen transparent gemacht, wäre die Simulation zu einfach berechenbar.)
Opportunities	Threats
<ul style="list-style-type: none"> • Unsere Applikation ist teamfähig: Die Gruppen spielen gegeneinander unter Aufsicht des Spielleiters und können sich somit direkt vergleichen. Es kann eine Diskussion stattfinden und gemeinsames Lernen wird unterstützt • Kompetitiver Aspekt: Es könnten Firmenteams gegeneinander antreten • Networking: Sollte diese Applikation als Workshop mit Teilnehmern aus verschiedenen Firmen durchgeführt werden, kann der Workshop auch nebenbei als Networking-Plattform dienen • Marketing: Die Simulation wurde an einer Hochschule entwickelt unter Aufsicht des Dozenten für Projekt- und Qualitätsmanagement 	<ul style="list-style-type: none"> • Benötigt einen mit der Domäne vertrauten Spielleiter zur Durchführung, da die Funktionen und Auswirkungen der einzelnen Parameter nicht intuitiv verständlich sind • Detailgrad der Funktionen muss erhöht werden (Stichwort: kein Zufallssieger)

Abbildung 2: SWOT-Analyse unserer Projektmanagementsimulation.

Inhaltsverzeichnis

Änderungsverlauf	ii
Abstract	iii
Management Summary	iv
1. Einleitung	4
1.1. Ziele	4
1.2. Vorgehen, Technologien	4
1.3. Aufbau	5
2. Theoretische Fragestellung	6
2.1. Ausgangslage	6
2.2. Positiver Lerneffekt durch Lernspiele	6
2.3. Fazit	7
3. Betriebswirtschaftliche Fragestellung	8
3.1. Vergleich Funktionsumfang	9
3.1.1. Tabelle Funktionsumfang	9
3.1.2. Schlussfolgerungen Funktionsumfang	9
3.2. Vergleich Kosten	10
3.2.1. SimulTrain	10
3.2.2. Cesim Project	11
3.2.3. Eduardo	11
3.2.4. thatpgame.com	11
3.2.5. Schlussfolgerungen Kosten	11
3.3. Fazit	12
4. Praktische Fragestellung	14
4.1. Ausgangslage & Aufgabenstellung	14
4.2. Lösungskonzept & Umsetzung	14
4.3. Ergebnis & Schlussfolgerungen	15
5. Software Engineering-Dokumentation	17
5.1. Funktionale Anforderungen	17
5.1.1. User Stories	17
5.1.2. Use Cases	17
5.1.3. User Stories: Abhängigkeitsgraph	20
5.2. Use Case-Beschreibungen	22
5.3. Erweiterungsmöglichkeiten	35
5.4. Domain Model	36
5.5. Nicht-funktionale Anforderungen	38
5.5.1. Zu unterstützende Szenarien	38

5.5.2.	Priorisierung	38
5.5.3.	Nicht-funktionale Anforderungen	40
5.5.4.	Erläuterungen	46
5.6.	Datenmodell	48
5.7.	Logische Architektur	49
5.8.	Deployment-Diagramm	50
5.9.	Gui-Skizzen	50
5.10.	Architekturentscheidungen: Diskussion	52
5.11.	Testing	54
5.11.1.	Testing der funktionalen Anforderungen	54
5.11.1.1.	Black-Box-Testing	54
5.11.1.2.	White-Box-Testing	55
5.11.2.	Testing der nicht-funktionalen Anforderungen	56
5.11.2.1.	Live-Tests	58
5.11.2.2.	Erkenntnisse	58
5.12.	Installationsanleitung	59
5.12.1.	Voraussetzungen	59
5.12.2.	Deployment auf HSR Server	59
5.12.3.	Installation auf einem neuen Host	60
5.12.4.	Lokales Ausführen der Applikation	60
5.12.5.	Datensicherung Datenbank	61
5.13.	Benutzerhandbuch (User Manual)	61
5.13.1.	Anlegen eines neuen Spiels (Admin)	61
5.13.2.	Spiel starten (Admin)	62
5.13.3.	Spielereingabe (Spieler)	63
5.13.4.	Change Request bearbeiten (Admin)	64
5.13.5.	Runde beenden (Admin)	64
5.14.	Known Bugs	64
5.15.	Screenshots	65
5.16.	Software-Metriken und Code Analyse	81
5.17.	Sonstiges	86
5.17.1.	Routing-Tabelle	86

6. Projektdokumentation 88

6.1.	Projektausschreibung	89
6.2.	Aufgabenstellung	90
6.3.	Zeitplan	91
6.4.	Projektplan	92
6.4.1.	Kritischer Rückblick: Vergleich Ist vs. Soll	92
6.5.	Projektcontrolling	94
6.5.1.	Erstellung des Controlling-Cockpits	94
6.5.2.	Projektbudget	96
6.6.	Risikoliste	96
6.6.1.	Eingetretene Risiken	96

Literatur	101
Abbildungsverzeichnis	102
Tabellenverzeichnis	104
A. E-Mail Verkehr	105
A.1. STS	105
A.2. EDUardo	108
B. Projektplan	109
C. Nicht-funktionale Anforderungen: Grafik Rohdaten	125
D. Einschätzung der Metrik Lines of Code	126

1. Einleitung

Für die Vertiefung des HSR Moduls „Projekt- & Qualitätsmanagement“ wird für den Unterricht eine Managementsimulation verwendet. Bisher wurde diese weitestgehend offline mit einem Excel-Sheet durchgeführt. Im Rahmen dieser Arbeit wird die Managementsimulation in eine Software übertragen, so dass die Studierenden auf spielerische Art und Weise das erlernte Wissen nachhaltig verankern zu können.

Die dabei soll eine Applikation entstehen, welche einen ausreichenden Funktionsumfang sowie die nötige Stabilität bietet, so dass diese im Unterricht verwendet werden kann. Die Studierenden haben die Möglichkeit, während den Übungsstunden oder selbständig zu Hause Spielzüge vorzunehmen.

1.1. Ziele

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

1. Gibt es in der Literatur oder bei anderen Instituten Hinweise darauf, um wie viel effizienter spielerisches Lernen ist?
2. Können die Anforderungen des Dozenten in einer Software umgesetzt werden? Begründen, wenn Anforderungen nicht wie gefordert umgesetzt werden konnten.
3. Könnte die HSR diese Software in Form einer Trainingsdienstleistung im freien Markt anbieten, um Software-Unternehmen im Bereich Projektmanagement fitter zu machen?

1.2. Vorgehen, Technologien

Durch die Vorgaben an das Projekt war recht schnell klar, dass die Managementsimulation als Webapplikation umgesetzt werden sollte. Dies bietet den Studenten einen einfachen Zugang zu der Applikation und erlaubt dem Betreuer eine einfache Verwaltung der Spiele. Da die Applikation flexibel und unabhängig eines spezifischen Anbieters betrieben werden können sollte, war die Auswahl der zu verwendenden Technologien auf PHP und Node.js beschränkt. Diese haben eine genug grosse Verbreitung, so dass für eine entsprechende Applikation vergleichsweise einfach eine Hosting-Plattform gefunden werden kann. Wir haben uns schlussendlich für Node.js entschieden, da wir uns darin besser gefestigt fühlten.

Die Entwicklung der Software war so geplant, dass in regelmässigen Abständen eine lauffähige Version der Applikation auf einer Staging-Umgebung verfügbar gemacht wird, welche vom Betreuer verwendet werden kann. Gegen Ende des Semester sollte die Applikation dann den Umfang und die Stabilität erreicht haben, dass dieser lauffähige Prototyp bereits im Unterricht eingesetzt werden kann.

Um diese Vorgaben erreichen zu können, haben wir uns für den Entwicklungsprozess entschieden auf SCRUM+ zu setzen. Bei einer längeren Projektlaufzeit mit flexibleren Testing Terminen wäre eine voll agile Umsetzung attraktiver gewesen, da die Projektdauer jedoch zeitlich stark begrenzt sowie mit einigen fixen Deadlines versehen war, setzten wir auf ein agiles Vorgehen mit einigen fixen Terminen.

1.3. Aufbau

In dieser Dokumentation beantworten wir die im Abschnitt „Ziele“ genannten Fragestellungen. Die Resultate werden in verschiedenen Abschnitten präsentiert:

- Die theoretische Fragestellung wird im Kapitel 2 „Theoretische Fragestellung“ - durch eine Literaturrecherche innerhalb dieser Dokumentation beantwortet.
- Für die betriebswirtschaftliche Fragestellung wurde eine Konkurrenzanalyse durchgeführt. Die Ergebnisse dieser Analyse liegen im Kapitel 3 „Betriebswirtschaftliche Fragestellung“ vor.
- Die praktische Umsetzung der Arbeit erfolgte mittels lauffähiger Applikation, welche der Dozent im Unterricht vollumfänglich verwenden kann. Der Sourcecode dieser Applikation sowie die dazugehörige Software-Dokumentation sind daher ebenfalls Teil der Projektabgabe.

2. Theoretische Fragestellung

In diesem Kapitel fassen wir unsere Resultate zusammen, welche wir aus der Recherche diverser Studien erhalten haben. Ziel der Untersuchung war herauszufinden, ob spielerisches Lernen effizienter ist als klassisches Lernen. Die Thematik wurde bereits in zahlreichen Studien eingehend untersucht. Wir haben uns daher rein auf Studien konzentriert, welche sich mit computergestützten Lernen beschäftigen, da dies für unsere Arbeit am relevantesten schien.

2.1. Ausgangslage

Der Computer als Unterrichtsmittel ist in unserem Bildungswesen auf allen Stufen bereits breit vertreten[1]. Dies umfasst beispielsweise den Einsatz des Computers zu Recherchezwecken sowie die Office-Anwendungen zur Textverarbeitung.

Seit einigen Jahren kommen jedoch, speziell in den unteren Schulstufen, vermehrt Software-Programme zum Einsatz, welche computergestütztes Lernen anbieten. Als Vertreter sind hierbei „Cluefinders (Dt. Die Schlaue Bande)“[2] oder „Globi 123“[3] zu nennen. Schüler können damit auch Zuhause selbständig Inhalte erarbeiten. Mit der Verfügbarkeit von Mobiltelefonen, Tablets und Apps wurde die Verwendung solcher Tools sogar noch intensiviert. [4]

Neben diesen Spielen, welche sich vorwiegend an ein jüngeres Lernpublikum richten, sind auch eine Vielzahl an Applikationen auf dem Markt, welche sich an ältere Studenten oder sogar Erwachsene richten. So hat der Anbieter Cesim diverse Managementsimulationen im Angebot, welche jeweils unterschiedliche Managementbereiche abdecken und für Managementausbildungen verwendet werden. [5]

2.2. Positiver Lerneffekt durch Lernspiele

Der förderliche Effekt von Lernspielen auf Schülerinnen und Schüler ist durch mehrere Studien belegt. [6] [7] Schüler werden jedoch nicht automatisch besser durch das Spielen von Lernspielen. Tatsächlich nehmen verschiedene Faktoren Einfluss auf den kognitiven Lerneffekt. [8]

So können Elemente wie Punkte, Spielgrafik oder herausfordernde Spielsteuerung dazu verwendet werden, um Inhalte in spielerischer Art und Weise verpackt zu vermitteln. Die Inhalte können jedoch auch eher schlicht daher kommen, dafür mehr Komplexität bieten. Die Ausgestaltung der Applikation hat einen direkten Einfluss auf die Lerneffektivität. [9]

So konnte bewiesen werden, dass der erlebte Spielspass zwar Kinder dazu motiviert häufiger und länger ein entsprechendes Lernangebot in Anspruch zu nehmen, der kognitive Lernzuwachs jedoch nicht in Relation mit dem erlebten Spielspass steht. [7] Merkmale wie Punkte, Spielgrafik oder herausfordernde Spielsteuerung tragen gemäss der Studie

weder zu einer Steigerung des motivationalen noch des kognitiven Lernzuwachs bei. Der eigentliche Lerneffekt tritt in Zusammenhang mit den vermittelten Informationen sowie dem interaktiven Handeln auf.[8]

Wir schliessen daraus, dass unsere Applikation eine ausgewogene Mischung zwischen Spielspass und Komplexität bieten muss. Der Spielspass sorgt dafür, dass die Studenten motiviert sind und die Simulation seriös spielen. Die Komplexität des Spiels sorgt hingegen dafür, dass über die getätigten Spieleingaben nachgedacht werden muss um Zusammenhänge und Auswirkungen zu erkennen und dadurch die gelernten Inhalte vertieft und gefestigt werden.

Für uns ist daher klar, dass die Managementsimulation eine ansprechende Oberfläche mit einer guten Usability bieten muss. Der Spieler muss die Möglichkeit haben, sich mit anderen Spielern zu vergleichen. Wir werden jedoch kein User Interface bieten, welches zu stark gamifiziert wurde, da dies gemäss der Studie von „Nina Imlig-Iten“ keinen besonderen Einfluss auf die Lerneffektivität hat. [8]

2.3. Fazit

Wir sehen aufgrund der Aussagen der bereits zahlreich durchgeführten Studien einen positiven Effekt auf Lernende. Die gesteigerte Lerneffektivität der Probanden wurde nachgewiesen und es wurde auch untersucht, welche Faktoren bei den Spielen für den Lernerfolg verantwortlich sind.

Unsere Applikation dürfte sich daher positiv auf den Lernerfolg der Studenten auswirken, welche damit eine Managementsimulation durchspielen können. Die Erkenntnisse über die verschiedenen Faktoren zum Lernerfolg helfen uns zudem die Managementsimulation so zu gestalten, dass diese einen möglichst hohen Lerneffekt erzielt und für den Dozenten im Unterricht ein sinnvolles Lehrmittel ist.

3. Betriebswirtschaftliche Fragestellung

Wir haben verschiedene Anbieter untersucht, um zu eruieren ob das Potential vorhanden ist, die Applikation „Project Manager“ als Learning-Plattform kommerziell anzubieten.

Dazu haben wir ebenfalls die Preisstrukturen analysiert, um eine ungefähre Einschätzung über eine mögliche Preisgestaltung für „Project Manager“ abgeben zu können.

Dazu haben wir nachfolgende Produkte genauer untersucht:

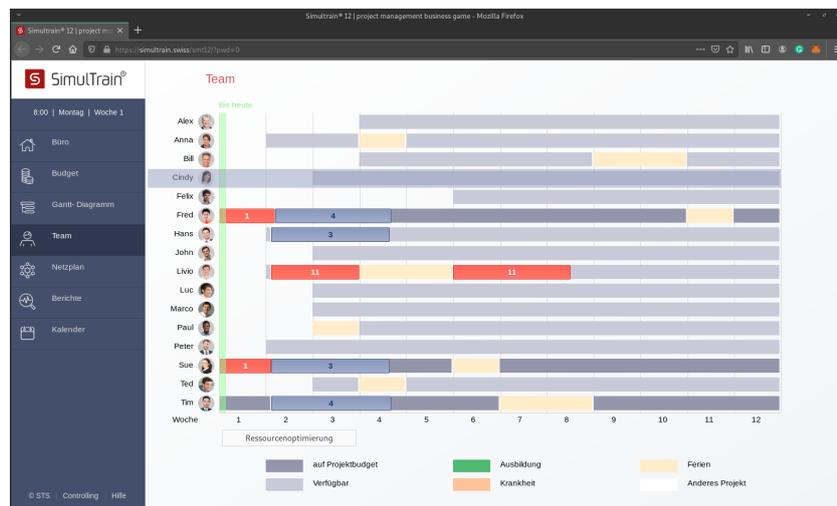


Abbildung 3: Ansicht der Mitarbeiterzuweisung in SimulTrain

SimulTrain Die Software des Schweizer Anbieters STS 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.[10]

Cesim Project Der Schwedische Anbieter Cesim 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. [5]

thatpmgame.com Die Webseite thatpmgame.com ist eine sehr einfach gehaltene, aber kostenlose Simulation. Die Spieler können verschiedene Projektszenarien durchspielen und dabei eine simple Einsatzplanung vornehmen. Es werden jeweils zufällig Projektszenarien generiert, welche der Spieler mit den optimalen Mitarbeiter besetzen muss. Es können zufällige Events eintreten, welche Auswirkung auf den Spielverlauf haben. [11]

Eduardo Eduardo ist eine Plattform, welche sechs verschiedene Module anbietet, um verschiedene Managementaufgaben zu trainieren. Diese Module sind nicht spezifisch dafür ausgelegt ein Software-Projekt zu simulieren, dennoch ist die thematische Schnittmenge gross, weshalb wir Eduardo ebenfalls in unseren Vergleich aufgenommen haben. Die Module umfassen Unternehmertum, Business Development, Kommunikationsfähigkeiten, Strategisches Denken, Anpassungsfähigkeiten sowie Change Management.[12]

3.1. Vergleich Funktionsumfang

Um das Marktpotential unserer Applikation abschätzen zu können, haben wir den Funktionsumfang der entsprechenden Applikationen mit unserer Applikation „Project Manager“ verglichen. Es geht hierbei in erster Linie um eine grobe Übersicht, da ein direkter Vergleich aufgrund der unterschiedlichen Ausrichtungen und Funktionen nicht ohne weiteres möglich ist. Einen detaillierteren Vergleich auszuarbeiten, hätten den Rahmen dieser Arbeit gesprengt.

3.1.1. Tabelle Funktionsumfang

Die Daten über den Funktionsumfang der betreffenden Applikationen haben wir anhand der offiziellen Dokumentationen der Anbieter, deren Demos der Software sowie per E-Mail-Austausch und Telefonkonferenzen mit den Anbietern gesammelt. Bei sämtlichen Anbietern war eine Demo möglich. Die Resultate haben wir in der Tabelle 2 aufbereitet.

3.1.2. Schlussfolgerungen Funktionsumfang

Es existieren diverse Applikationen für Projektmanagementsimulationen von verschiedenen Anbietern. Einige decken auch ganz spezifisch das Themengebiet Software Entwicklung ab. Es ist daher anzunehmen, dass ein Markt für entsprechende Applikationen besteht.

Vom Funktionsumfang und Detailgrad der Simulationen her bietet unsere Applikation ein gutes Mittelmass. Einige Simulationen bieten einen extrem hohen Detailgrad für gewissen Funktionen, so müssen beispielsweise bei „SimulTrain“ einzelne Mitarbeiter mit unterschiedlichen Skills einem spezifischen Projekttask zugewiesen werden.[13] Andere Simulationen wie das kostenlose „thatpmpgame.com“ wiederum bieten einen sehr eingeschränkten Funktionsumfang.

Unsere Applikation bietet im Vergleich mit den bestehenden Lösungen einen soliden Funktionsumfang. Zur Zeit ist „Project Manager“ für den Einsatz im „Projekt- & Qualitätsmanagement“ Unterricht optimiert. Es wären daher noch diverse Änderungen nötig, damit die Applikation für eine generische Schulung eingesetzt werden kann (Benutzerverwaltung, Mitteilungen, Verfeinerung bestehender Funktionen). Wir sehen jedoch klar ein Potential,

	Project Manager	SimulTrain	Cesim Project	thatpgame.com	Eduardo
Kooperativer Modus	✓		✓		
Mitarbeiterplanung	✓	✓	✓	✓	✓
Events	✓	✓		✓	
Terminplanung		✓			
Risikoreduktion	✓	✓			
Projektleiterplanung	✓				
Eigene Szenarien möglich	✓				✓
Online-Forum			✓		✓
Quiz			✓		
Change Requests	✓				✓
Mehrsprachig		✓	✓		

Tabelle 2: Vergleich Funktionsumfang

dass die Applikation kostenpflichtig für externe Schulungen durch die HSR bereitgestellt werden kann.

3.2. Vergleich Kosten

In folgendem Abschnitt haben wir die Kosten der betreffenden Applikationen zusammengetragen. Eine Empfehlung für den Preis von „Project Manager“ lässt sich daraus jedoch nicht ableiten. Dennoch bieten die Daten eine ungefähre Übersicht, in welchem Bereich der Preis von „Project Manager“ liegen könnte.

3.2.1. SimulTrain

Die Kosten für SimulTrain belaufen sich auf CHF 905 für eine Gruppe von vier Spielern. Die Lizenz gilt für eine einmalige Durchführung.² Öffentlichen Schulen und Hochschulen wird ein Rabatt von 75% gewährt. Bei einer weiteren Durchführung wird eine neue Lizenz benötigt.

²Gilt für ein Durchlauf einer Simulation, die Resultate sind anschliessend für 3 Monate noch einsehbar.
Siehe Anhang

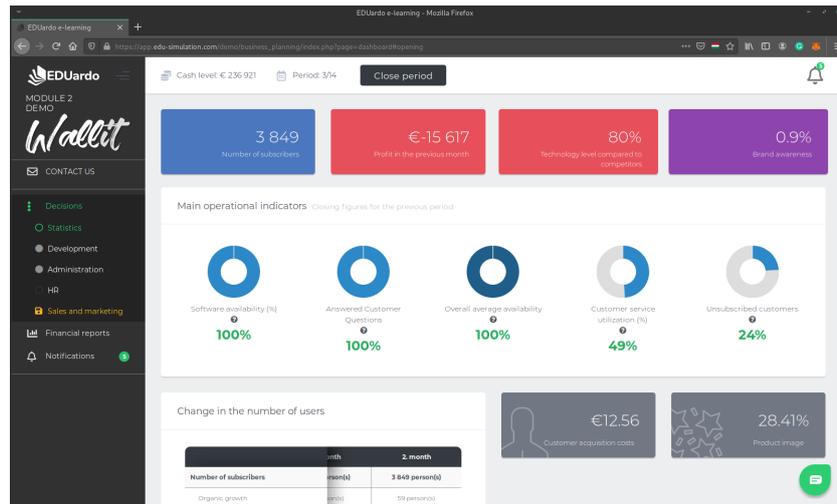


Abbildung 4: Dashboard von Eduardo

3.2.2. Cesim Project

Bei Cesim Project wird eine Lizenz pro Durchführung einer Simulation benötigt. Diese kostet pro Student EUR 45 (CHF ~49.30³). Werden grössere Lizenzpakete gekauft, so wird ein Rabatt gewährt. Öffentliche Schulen erhalten keinen speziellen Rabatt.⁴

3.2.3. Eduardo

Bei Eduardo müssen die einzelnen Module separat lizenziert werden. Dies kostet EUR 35 pro Modul (CHF ~38.30⁵) und gilt jeweils pro Semester. Werden alle 6 Module benötigt so kostet dies CHF 210 pro Studenten. Für Durchführungen mit über 100 Studenten wird ein Rabatt gewährt.

3.2.4. thatpmsgame.com

Diese Applikation ist kostenlos verfügbar, kann jedoch in keinster Weise mit dem Detailgrad und Komplexität der anderen Produkte verglichen werden.

3.2.5. Schlussfolgerungen Kosten

Der Vertrieb von „Project Manager“ als Applikation ist aus unserer Sicht eine finanziell attraktive Option. Die bestehenden Lösungen positionieren sich ganz klar als Spezial-

³Stand 16.12.2019

⁴Diese Angaben wurden während einer Telefonkonferenz mit Cesim genannt. Schriftliche Informationen haben wir bis zum Zeitpunkt dieser Arbeit leider noch nicht erhalten.

⁵Stand 15.12.2019

anwendungen im Markt und kosten auch dementsprechend. Auch „Project Manager“ könnte sich preislich in diesem Segment ansiedeln.

Ein direkter Vergleich zwischen den verschiedenen Applikationen ist nicht aussagekräftig, da diese sich teilweise erheblich vom Funktionsumfang, Laufzeit der Lizenz und dem dazu mitgelieferten Schulungsmaterial unterscheiden. Dennoch haben wir in der Tabelle 3 einen rein finanziellen Vergleich aufgestellt, welcher sämtliche anderen Parameter ignoriert. Die berechneten Kosten gelten für 40 Teilnehmer während einem Semester (14 Wochen).

Produkt	Laufzeit Lizenz	Kosten für 40 Studenten
SimulTrain	Eine Durchführung	CHF 2262.5 ⁶
Eduardo	6 Monate	CHF 9192
thatpmgame.com	-	CHF 0
Cesim Project	Eine Durchführung	CHF 1972

Tabelle 3: Vergleich Kosten für 40 Teilnehmer über ein Semester

3.3. Fazit

Unsere Untersuchung hat ergeben, dass die Applikation „Project-Manager“ sich gut eigenen würde, kostenpflichtig für Managementschulungen vertrieben zu werden. Die Applikation bietet einen guten Funktionsumfang und kann mit den etablierten Anbietern durchaus mithalten. Einige Funktionen wie das Change-Request Feature, die Events oder die durchgehende kompetitive Komponente sind bei „Project-Manager“ im Vergleich zur Konkurrenz stark ausgeprägt.

Vorgängig wären jedoch diverse Anpassungen nötig. So ist derzeit die Benutzerverwaltung nur äusserst rudimentär implementiert. Ebenfalls muss der Spielablauf an einigen Stellen optimiert werden, damit ein positives Spielerlebnis sichergestellt werden kann. Auch sind gewisse Funktionen derzeit stark vereinfacht umgesetzt und bedürften einer Überarbeitung, damit komplexere Abläufe möglich sind.

Die Investition in diese Funktionalitäten könnte sich jedoch aus unserer Sicht lohnen, da diese Art von Applikation als Spezialsoftware im eher hochpreisigem Segment angesiedelt ist. Zudem ist die Anzahl der Konkurrenten überschaubar, was den Einsteig in diesen Markt attraktiv macht.

Unsere Erkenntnisse lassen sich auch aus der nachfolgenden SWOT-Analyse in der Abbildung 5 ablesen:

Strengths	Weaknesses
<ul style="list-style-type: none"> • Change Request-Feature: <ul style="list-style-type: none"> - hat im Live-Test den Austausch zwischen den Studierenden und dem Spielleiter stark gefördert. - hat auch den Austausch unter den Studierenden gefördert • flexibel ausbaubar • Event-Feature 	<ul style="list-style-type: none"> • Detailgrad beim Funktionsumfang ist kleiner als bei anderen Applikationen • Teilweise noch auf Zufall basierend (dies vor allem dadurch, da der Detailgrad der Funktionen klein ist und somit nicht alle Hintergrundberechnungen offengelegt werden können. Würden alle Berechnungen transparent gemacht, wäre die Simulation zu einfach berechenbar.)
Opportunities	Threats
<ul style="list-style-type: none"> • Unsere Applikation ist teamfähig: Die Gruppen spielen gegeneinander unter Aufsicht des Spielleiters und können sich somit direkt vergleichen. Es kann eine Diskussion stattfinden und gemeinsames Lernen wird unterstützt • Kompetitiver Aspekt: Es könnten Firmenteams gegeneinander antreten • Networking: Sollte diese Applikation als Workshop mit Teilnehmern aus verschiedenen Firmen durchgeführt werden, kann der Workshop auch nebenbei als Networking-Plattform dienen • Marketing: Die Simulation wurde an einer Hochschule entwickelt unter Aufsicht des Dozenten für Projekt- und Qualitätsmanagement 	<ul style="list-style-type: none"> • Benötigt einen mit der Domäne vertrauten Spielleiter zur Durchführung, da die Funktionen und Auswirkungen der einzelnen Parameter nicht intuitiv verständlich sind • Detailgrad der Funktionen muss erhöht werden (Stichwort: kein Zufallssieger)

Abbildung 5: SWOT-Analyse unserer Projektmanagementsimulation.

4. Praktische Fragestellung

Wir halten diesen Teil bewusst kurz. Für Details zum Software Engineering sei auf das Kapitel 5 „Software Engineering-Dokumentation“ verwiesen. Für Details zum Projektverlauf sei auf das Kapitel 6 „ Projektdokumentation“ verwiesen.

4.1. Ausgangslage & Aufgabenstellung

In der Aufgabenstellung⁷ wurden wir beauftragt, eine Lernsimulation respektive ein Lernspiel für das Modul „Projekt- und Qualitätsmanagement“ (PmQm) als Software umzusetzen. Beim Lernspiel handelt es sich um ein fiktives IT-Projekt, welches von einer Gruppe über mehrere Runden betreut und gesteuert werden muss. Die Stellschrauben sind dabei die Wahl der Mitarbeiter, die Investition in Ausbildung der Mitarbeiter, Investitionen in risikoreduzierende Massnahmen sowie die Zuteilung des Projektleiters auf Aufgabengebiete. Es treten jeweils mehrere Gruppen gegeneinander an. Es gibt einen Spielleiter, welcher durch das Spiel führt und die Umgebungsvariablen des Spiels beeinflussen kann.

Die Abgabe der Software erfolgt sowohl als implementierte Lösung, Source-Code als auch als Software-Engineering-Dokumentation. Bisher wurde diese Lernsimulation in „Projekt- und Qualitätsmanagement“ manuell via Emails und Excel gespielt.

Zusätzlich haben wir durch den Ausschreibungstext⁸ bereits vorgefertigte User Stories erhalten. Diese dienten uns als Vorlage.

4.2. Lösungskonzept & Umsetzung

Wir haben uns für das Vorgehensmodell⁹ Scrum+ entschieden. Darauf basierend haben wir den Zeitplan¹⁰ mit den vier Phasen Inception, Elaboration, Construction und Transition aufgestellt.

In der einwöchigen Inception-Phase verschafften wir uns zunächst einen Überblick über die Aufgabenstellung und den Umfang. In der dreiwöchigen Elaboration-Phase erstellten wir relativ früh ein erstes Domain Model. Ebenfalls verfeinerten und erweiterten wir die User Stories. Dadurch hatten wir bis End of Elaboration ein gutes Verständnis, was für eine Lösung von uns gefordert wird. Ebenfalls sind in der Elaboration-Phase diverse weitere Skizzen, Diagramme und Templates entstanden, u.a. die nicht-funktionalen Anforderungen im „SMART“-Stil.

Basierend auf den erhobenen Anforderungen haben wir die Technologien gewählt. Die Software sollte auf möglichst vielen Endgeräten laufen. Entsprechend haben wir uns für

⁷Siehe Abschnitt 6.2 für die Aufgabenstellung.

⁸Siehe Abbildung 52 für den Ausschreibungstext.

⁹Siehe im Projektplan in Abschnitt 6.4 für weitere Details.

¹⁰Siehe Abschnitt 6.3 für den Zeitplan.

eine Web-Client-Lösung entschieden. Aus den nicht-funktionalen Anforderungen kam des weiteren heraus, dass die Software möglichst einfach zu warten und bei möglichst vielen Hosting-Anbietern lauffähig sein sollte. Daher setzen wir auf NodeJS in Verbindung mit TypeScript. Wir verwenden zudem handlebars.js, um die ausgelieferten Seiten serverseitig zu rendern.

Die Verbindung zwischen Backend und Datenbank haben wir mit dem SQL Query builder „knexjs“¹¹ hergestellt. Die Idee war, dass wir so einfach von einer Datenbank-Technologie (z.B. SQLite) auf eine andere Datenbank-Technologie (z.B. MySQL) wechseln können. Als Datenbank haben wir eine SQLite-Datenbank eingesetzt. Es war geplant, diese SQLite-Datenbank in der Folge durch eine MySQL-Datenbank abzulösen, indem wir die knexjs-Konfiguration ändern. Jedoch hatten wir beim Aufbau der MySQL Umgebung Schwierigkeiten, sodass die Software auf unserem Server nach wie vor auf einer SQLite-Datenbank läuft. Der Betrieb auf einer anderen Datenbanktechnologie wäre jedoch aufgrund der Verwendung von knexjs möglich.

In der achtwöchigen Construction-Phase setzten wir die Software gemäss den erhobenen Anforderungen um. Wir arbeiteten mit jeweils zweiwöchigen Sprints. In der knapp zweiwöchigen Transition-Phase lag der Fokus auf dem Zusammenstellen des Schlussberichts.

4.3. Ergebnis & Schlussfolgerungen

Die umgesetzten User Stories erlauben es, ein komplettes Spiel im Unterricht durchzuführen. Von den erhobenen 40 User Stories konnten 29 implementiert werden. Insbesondere konnten bis auf eine Ausnahme alle User Stories mit der Priorität „MUSS“ umgesetzt werden.¹² Bei den nicht umgesetzten Features handelt es sich einerseits um administrative Tasks wie Spieler löschen, Spiel löschen sowie Daten exportieren und andererseits um Notification-Features.

Die nicht-funktionalen Anforderungen haben wir nur teilweise erreicht. Insbesondere das automatisierte Testing kam zu kurz.

Im Nachhinein würden wir folgende Dinge anders handhaben:

- Datenbank bereits frühzeitig auch auf dem Livesystem testen.
- Mehr Zeit auch für einfache Tätigkeiten einplanen. Wir haben viele der Arbeitspakete unterschätzt. Insbesondere der Aufwand für das Testing haben wir unterschätzt.
- Qualitätsmassnahmen besser einhalten. Bedingt aber, dass dazu auch mehr Zeit einplant wird.

Folgende Punkte würden wir wieder gleich machen:

¹¹Siehe Webseite <https://knexjs.org/> für eine Beschreibung von knexjs.

¹²Bei der Ausnahme handelt es sich um CRUD Aufwand, welche jedoch hard-coded in der Datenbank eingegeben werden kann.

- Das Vorgehensmodell Scrum+ hat unserer Ansicht gut auf die relativ kurze Projektlaufzeit von 14 Wochen gepasst. Durch den Fixpunkt End of Elaboration hat es hat uns zu Beginn genügend Druck gemacht diszipliniert zu arbeiten.
- Wie bereits im Engineering-Projekt erlebt hat uns die CI/CD-Pipeline viel Aufwand erspart.
- Gitflow mit (mindestens) den 3 Branches „develop“, „staging“ und „master“

Zum Schluss sei noch auf die aus unserer Sicht besonders interessanten Abschnitte aus der Software Engineering- und Projektdokumentation verwiesen.

- Funktionale Anforderungen: Abschnitt Erweiterungsmöglichkeiten
- Nicht-funktionale Anforderungen: Abschnitt Priorisierung sowie Abschnitt Erläuterungen
- Testing der nicht-funktionalen Anforderungen: Abschnitt Erkenntnisse
- Projektplan: Abschnitt Kritischer Rückblick: Vergleich Ist vs. Soll
- Risikoliste: Abschnitt Eingetretene Risiken

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 Stories

Die funktionalen Anforderungen wurden teilweise bereits bei der Projektausschreibung¹³ vom Product Owner in Form von User Stories vorgegeben. Darauf aufbauend haben wir in der Elaboration-Phase die vorgegebenen User Stories kritisch geprüft sowie in Abstimmung mit dem Product Owner einige wenige User Stories abgeändert und diverse neue User Stories hinzugefügt. Bei End of Elaboration hatten wir rund 35 User Stories zusammen. In Verlauf der Construction-Phase haben wir gemerkt, dass wir einige User Stories noch nicht abgedeckt hatten bei End of Elaboration, sodass schlussendlich 40 User Stories (siehe Abbildung 6) resultierten.

Die User Stories haben wir in enger Abstimmung mit dem Domain Model (siehe Abbildung 9) erarbeitet. Des Weiteren haben wir darauf geachtet, dass alle User Stories im gleichen Schema erfasst werden: „Ich als [Aktor] [Verb] [Objekt]“. Zudem haben wir explizit vermieden, die Verben „muss“, „soll“ und „kann“ zu verwenden. Denn diese Verben haben wir bereits in der Gliederung der User Stories nach Wichtigkeit angewendet.

5.1.2. Use Cases

Aus den User Stories haben wir in der Elaborationsphase Use Cases geformt (siehe Abbildung 7). Dazu haben wir die 40 User Stories in sinnvoll zusammenhängende Gruppen geordnet.

Die Idee ist, dass wir so ein höheres Abstraktionsniveau für den Product Owner erreichen können. Der Product Owner kann dann die Items im Product Backlog besser priorisieren. In dieser Arbeit wäre dies nicht zwingend notwendig gewesen, da der Product Owner die Domäne sehr gut kennt und das Spiel im Vorjahr selbst durchgeführt hat. Dennoch haben wir dies als gute Übung empfunden.

¹³Siehe Abbildung 52 im Abschnitt Projektausschreibung.

ID	Beschreibung	Priorität	Abhängigkeiten	Kommentar	Use Case	Use Case-Name	Use Case-Priorität	Estimate-Total	Estimate-Development	Estimate-Review/Testing
3	Ich als Spieler gebe mir Spiel Vorgaben an Budget, Zeit, Function Points, Projektziele, Risiken, Ausbildung, Anzahl Runden und Stakeholder vor	MUSS	1		UC01	Spiel erstellen	MUSS	13	8	5
4	Ich als Spieler definiere die Größe der Gruppen (1+n Spieler pro Team)	MUSS	1		UC01	Spiel erstellen	MUSS	5	3	2
9	Ich als Spieler lege fest, was die Gruppen genau pro Runde entgegen haben	SOLL	15, 17		UC01	Spiel erstellen	MUSS	7	4	3
36	Ich als Spieler teile einen oder mehrere Aufwände einem Spiel zu	MUSS	1		UC01	Spiel erstellen	MUSS	9	5	4
15	Ich als Spieler erstelle eine Gruppe	MUSS	15, 21		UC02	CRUD Gruppe	MUSS	6	4	2
16	Ich als Spieler weise Spieler einer Gruppe zu	MUSS	15, 21		UC02	CRUD Gruppe	MUSS	4	3	1
20	Ich als Spieler entferne ein Spieler aus einer Gruppe	SOLL	16, 18		UC02	CRUD Gruppe	MUSS	5	4	1
21	Ich als Spieler trete einem Spiel bei	MUSS	1		UC03	Spiel beitreten	MUSS	6	4	2
1	Ich als Spieler lanciere ein Spiel mit 1+n Runden	MUSS	1		UC04	Spiel starten	MUSS	6	5	2
10	Ich als Spieler lege fest, wie fest die Ausbildung pro Runde wirksam ist	SOLL	2		UC05	Runde starten	MUSS	7	5	2
11	Ich als Spieler lege fest, wann welche Risiken eintreten	SOLL	2		UC05	Runde starten	MUSS	9	7	2
36	Ich als Spieler starte die Runden jeweils	MUSS	9, 10, 11, 12		UC05	Runde starten	MUSS	7	4	3
24	Ich als Spieler gebe jeder Runde die vom Spieler verlangten Parameter ein	MUSS	35		UC06	Runde spielen	MUSS	7	6	1
25	Ich als Spieler definiere, wie viel pro Runde in Reinereduktion investiert wird	MUSS	35		UC06	Runde spielen	MUSS	5	3	2
26	Ich als Spieler definiere, wie viele Personentage pro Runde gebucht werden	MUSS	35	ist nicht von User Story 33 abhängig, weil wir mit Defaultwerten arbeiten für die Mitarbeiter.	UC06	Runde spielen	MUSS	6	5	2
27	Ich als Spieler definiere, wie viel pro Runde in Ausbildung investiert wird	MUSS	35		UC06	Runde spielen	MUSS	8	3	2
28	Ich als Spieler definiere, für was der Projektleiter seine Zeit aufwendet	MUSS	35		UC06	Runde spielen	MUSS	6	7	1
32	Ich als Spieler modifiziere meine Eingaben, solange der Spieler die Runde noch nicht terminiert hat.	SOLL	35		UC06	Runde spielen	MUSS	6	5	1
29	Ich als Spieler definiere, wann ein Change Request gestellt wird	KANN	35		UC07	Change request erstellen	KANN	7	5	2
12	Ich als Spieler lege fest, wann der Lenkungsabschluss stattfindet	SOLL	2		UC08	Change Request erhalten	SOLL	5	3	2
13	Ich als Spieler lege fest, ob die Chingas vom Lenkungsabschluss genehmigt werden	KANN	29		UC09 / UC10	Change Request bestätigen / Change Request ablehnen	KANN	9	6	3
2	Ich als Spieler terminiere die Runden jeweils	MUSS	1		UC11	Runde beenden	MUSS	28	23	5
30	Ich als Spieler habe einen groben Überblick, wie weit die anderen Teams sind (Details nur für Spieler)	KANN	19		UC12	Spielübersicht	KANN	9	6	3
5	Ich als Spieler habe ein Dashboard, um die Eingabe aller Teams zu überwachen	SOLL	9, 24, 26, 27, 28		UC13	Dashboard aufrufen	SOLL	9	6	3
36	Ich als Spieler habe die Möglichkeit, Reports hinsichtlich Burn Down Chart oder Kostenentwicklung anzusehen	SOLL	9, 24, 26, 27, 28		UC14	Reports einsehen	SOLL	6	5	3
8	Ich als Spieler habe die Möglichkeit, Reports hinsichtlich Burn Down Chart oder Kostenentwicklung zu exportieren	SOLL	36	benötigt sinvollweise, dass alle Eingaben von den Gruppen gemacht wurden	UC15	Reports exportieren	SOLL	12	8	4
14	Ich als Spieler definiere Events, welche eintreten können	SOLL			UC16	CRUD Event	SOLL	10	8	3
6	Ich als Spieler habe die Möglichkeit, die Gruppen via Email zu informieren	SOLL	15, 17		UC17	Benachrichtigung versenden (E-Mail und Notification)	SOLL	8	6	3
7	Ich als Spieler informiere jederzeit die Gruppen (keine Spiele) via Notifications	KANN	15, 17		UC17	Benachrichtigung versenden (E-Mail und Notification)	SOLL	6	5	2
31	Ich als Spieler werde per E-Mail informiert, wenn für meine Gruppe eine Eingabe ausstehend ist; ersetzt durch Ich als Spieler informiere andere Spieler, welche noch keine Eingabe gemacht haben, via E-Mail.	KANN	19, 35		UC18	Benachrichtigung bei ausstehenden Eingaben	KANN	8	5	3
34	Ich als Spieler informiere meine E-Mailadresse	KANN			UC19	CRUD Spieler	MUSS	3	2	1
40	Ich als Spieler lege ein Benutzerkonto an	MUSS			UC19	CRUD Spieler	MUSS	6	7	2
17	Ich als Spieler eröffne eine Gruppe.	KANN	21		UC20	Spieler eröffnet eine Gruppe	SOLL	5	3	2
19	Ich als Spieler bestätige eine Gruppe, sobald die vom Spieler definierte Anzahl Spieler der Gruppe beigetragen sind	KANN	4, 16, 18		UC20	Spieler eröffnet eine Gruppe	SOLL	4	3	1
23	Ich als Spieler gebe meiner Gruppe einen Friseurnamen	SOLL	19		UC20	Spieler eröffnet eine Gruppe	SOLL	2	1	1
18	Ich als Spieler trete einer Gruppe bei.	KANN	15, 17		UC21	Spieler tritt einer Gruppe bei	KANN	5	4	2
39	Ich als Spieler erfasse einen neuen Spieler an anhand dessen E-Mail Adresse	KANN			UC22	CRUD Spieler	KANN	7	6	2
22	Ich als Spieler entferne ein Spieler aus einem Spiel	SOLL	1		UC23	Spiel editieren oder löschen	SOLL	7	5	2
33	Ich als Spieler definiere Mitarbeitertypen hinsichtlich Produktivität und Kosten	KANN		KANN, weil wir mit Defaultwerten arbeiten können	UC24	CRUD MitarbeiterTyp	KANN	9	7	2
37	Ich als Spieler erfasse einen oder mehrere Aufwände	MUSS			UC25	CRUD Aufwand	MUSS	9	6	3
Summe								289,75	206,75	84

Abbildung 6: Liste der User Stories. Die grün markierten User Stories konnten im Rahmen der Arbeit umgesetzt werden.

Use Case	Use Case-Name	Priorität	Estimate: Total	Priorität Product Owner	Bemerkung Product Owner
UC01	Spiel erstellen	MUSS	33	1	
UC02	CRUD Gruppe	MUSS	14	1	
UC04	Spiel starten	MUSS	6	1	
UC05	Runde starten	MUSS	22	1	
UC06	Runde spielen	MUSS	36	1	
UC11	Runde beenden	MUSS	28	1	
UC19	CRUD Spieler	MUSS	11	1	Login wichtig
UC03	Spiel beitreten	MUSS	6	2	
UC25	CRUD Aufwand	MUSS	9	2	
UC13	Dashboard aufrufen	SOLL	9	3	
UC20	Spieler eröffnet eine Gruppe	SOLL	10	3	
UC23	Spiel editieren oder löschen	SOLL	7	3	
UC08	Change Request erhalten	SOLL	5	4	
UC14	Reports einsehen	SOLL	8	4	
UC15	Reports exportieren	SOLL	12	4	
UC16	CRUD Event	SOLL	10	4	
UC17	Benachrichtigung versenden (E-Mail und Notifik)	SOLL	14	5	
UC07	Change request erstellen	KANN	7	6	
UC09 / UC10	Change Request bestätigen / Change Request	KANN	9	6	
UC22	CRUD Spielleiter	KANN	7	6	
UC12	Spielübersicht	KANN	9	7	
UC18	Benachrichtigung bei ausstehenden Eingaben	KANN	8	7	
UC21	Spieler tritt einer Gruppe bei	KANN	5	7	
UC24	CRUD Mitarbeitertyp	KANN	9	8	
Summe			291		

Abbildung 7: Liste der Use Cases. Die grün markierten Use Cases konnten im Rahmen der Arbeit umgesetzt werden. Die blau markierten Use Cases konnten im Rahmen der Arbeit teilweise umgesetzt werden.

5.1.3. User Stories: Abhängigkeitsgraph

Wir haben einen Abhängigkeitsgraphen der User Stories erstellt, um auf komplexe Arbeitsschritte vorbereitet zu sein (siehe Abbildung 8). Das Ziel des Graphen ist, aufzuzeigen, welche User Stories miteinander gekoppelt sind. Zudem erlaubte es uns, einen Hinweis zu geben, welche User Stories zuerst implementiert werden sollten.

Aus dem Graphen zeigte sich vor allem, dass die Gruppenadministration¹⁴ diverse Abhängigkeiten untereinander hat. Entsprechend haben wir beim Schätzen der User Stories für die Gruppenadministration mehr Zeit eingeplant.

¹⁴In der Abbildung 8 in der unteren Mitte links zu sehen.

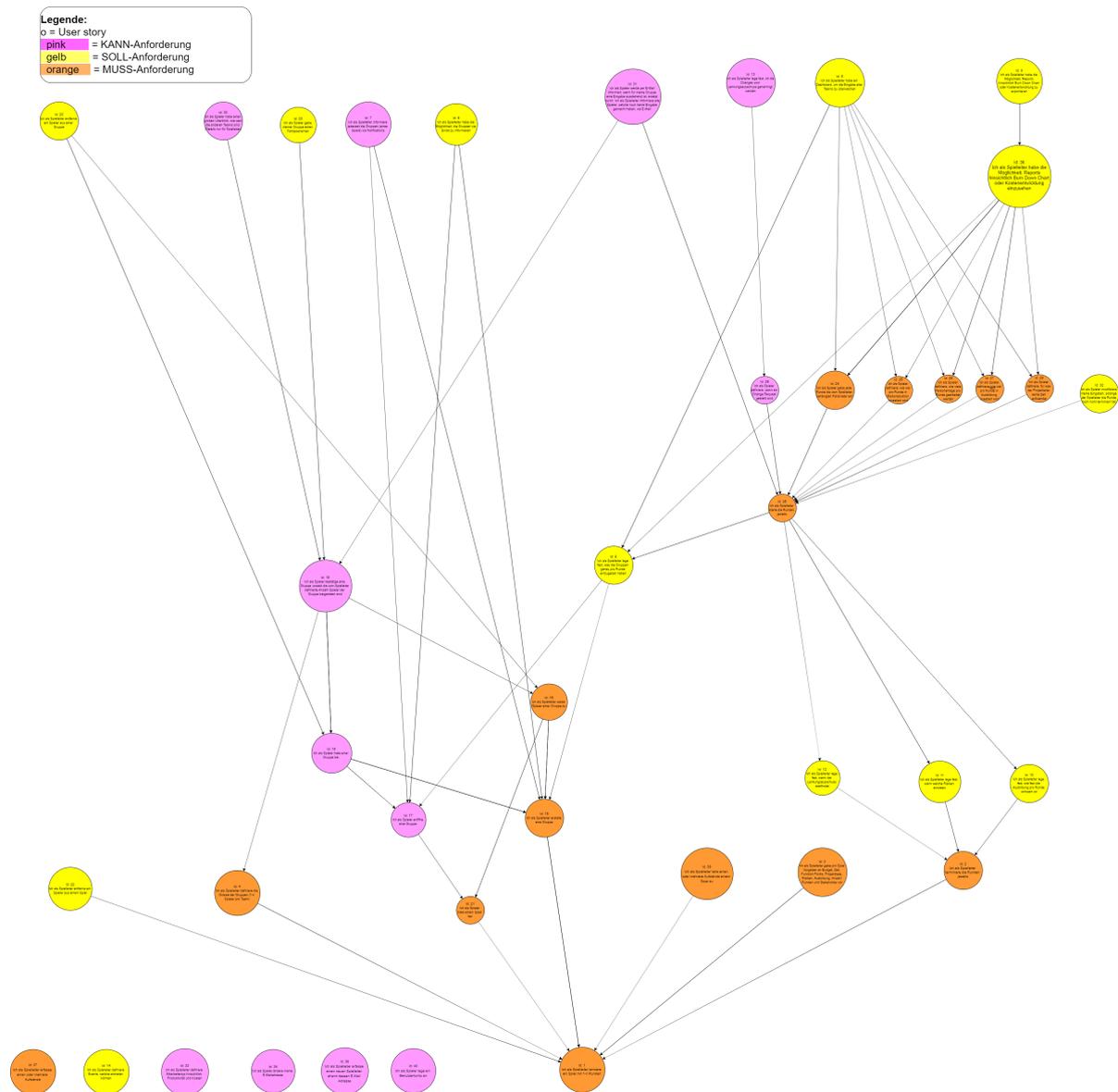


Abbildung 8: Die Abbildung zeigt die Abhängigkeiten der User Stories untereinander.

5.2. Use Case-Beschreibungen

Dieser Abschnitt zeigt die vollständig beschriebenen Use cases.

ID	UC1
Name	Spiel erstellen
Ziel im Kontext	Der Spielleiter erstellt ein neues Spiel
Akteure	Spielleiter
Trigger	
Essentielle Schritte	<ol style="list-style-type: none"> 1. Der Spielleiter klickt auf "Spiel erstellen"-Button 2. Der Spielleiter trägt einen Namen ein 3. Der Spielleiter trägt folgende Vorgaben ein: <ol style="list-style-type: none"> a. Startbudget b. Anzahl Runden c. Start Function Points d. Ausbildung e. Stakeholder f. Projektziele g. Gruppengrösse h. Aufwände 4. Der Spielleiter bestätigt das Spiel
Erweiterungen	<ol style="list-style-type: none"> 2a. Der Name existiert schon 2a1. Der Spielleiter erhält eine Warnmeldung 2a2. Der Spielleiter wählt einen neuen Namen
Abhängigkeiten	-
User Stories	3, 4, 9, 38 → Priorität: MUSS

ID	UC2
Name	CRUD Gruppe
Ziel im Kontext	Der Spielleiter erstellt, mutiert oder löscht eine Gruppe
Akteure	Spielleiter
Trigger	Alle Spieler sind registriert.
Essentielle Schritte	<ol style="list-style-type: none"> 1. Der Spielleiter wählt ein existierendes Spiel 2. Der Spielleiter klickt auf den "Team Erstellen"-Button 3. Der Spielleiter trägt den Gruppennamen ein 4. Der Spielleiter fügt n Spieler der Gruppe hinzu 5. Der Spielleiter bestätigt die Gruppe
Erweiterungen	<ol style="list-style-type: none"> 2a. Der Spielleiter möchte eine bestehende Gruppe löschen. 2a1. Der Spielleiter klickt auf das Papierkorb-Symbol bei der betreffenden Gruppe

	<p>2b. Der Spielleiter möchte eine bestehende Gruppe mutieren. 2b1. Der Spielleiter klickt bei der betreffenden Gruppe auf das Stift-Symbol 2b2. Weiter bei Schritt 3 mit vorausgefüllten Werten</p> <hr/> <p>3a. Der Gruppenname existiert bereits 3a1. Der Spielleiter erhält eine Warnmeldung 3a2. Der Spielleiter trägt einen neuen Gruppennamen ein</p> <hr/> <p>4a Spielleiter möchte Spieler entfernen 4a1. Spielleiter entfernt den gewünschten Spieler über das Papierkorb-Symbol.</p> <hr/> <p>5a. Ein Spieler wurde in der Zwischenzeit gelöscht 5a1. Der Spielleiter erhält eine Warnmeldung 5a2. Der Spielleiter führt Schritt 4 erneut aus</p>
Abhängigkeiten	UC1 / UC3
User Stories	15, 16, 20 → Priorität: MUSS

ID	UC3
Name	Spiel beitreten
Ziel im Kontext	Der Spieler tritt einem existierenden Spiel bei
Akteure	Spieler
Trigger	Der Spieler hat sich erfolgreich registriert.
Essentielle Schritte	<ol style="list-style-type: none"> 1. Der Spieler wählt auf der Spieleübersicht das gewünschte Spiel. 2. Der Spieler klickt auf den "Spiel beitreten"-Button
Erweiterungen	-
Abhängigkeiten	UC1
User Stories	21 → Priorität: MUSS

ID	UC4
Name	Spiel starten
Ziel im Kontext	Der Spielleiter startet das Spiel mit der ersten Runde
Akteure	Spielleiter
Trigger	Alle Gruppen sind organisiert.

Essentielle Schritte	1. Der Spielleiter klickt auf den "Spiel starten"-Button
Erweiterungen	-
Abhängigkeiten	UC2
User Stories	1 → Priorität: MUSS

ID	UC5
Name	Runde starten
Ziel im Kontext	Der Spielleiter startet eine neue Runde
Akteure	Spielleiter
Trigger	Das Spiel wurde gestartet oder die vorherige Runde beendet.
Essentielle Schritte	<ol style="list-style-type: none"> 1. Der Spielleiter trägt ein, wie viel die Ausbildung in dieser Runde wirksam ist. 2. Der Spielleiter legt fest, welches Risiko in der Runde eintritt 3. Der Spielleiter legt fest, was das Maximum an Personentage pro Runde ist 4. Der Spielleiter klickt auf den "Runde starten"-Button
Erweiterungen	
Abhängigkeiten	UC4
User Stories	10, 11, 35 → Priorität: MUSS

ID	UC6
Name	Runde spielen
Ziel im Kontext	Der Spieler spielt eine gestartete Runde
Akteure	Spieler
Trigger	Die Runde wurde gestartet
Essentielle Schritte	<ol style="list-style-type: none"> 1. Der Spieler erhält auf der Startseite das Rundenformular 2. Der Spieler trägt die geforderten Aufwände für diese Runde ein 3. Der Spieler trägt die Anzahl Personentage für diese Runde ein 4. Der Spieler trägt die Höhe der Ausbildungsinvestition für diese Runde ein 5. Der Spieler trägt die Höhe der Risikoreduktion ein 6. Der Spieler bestätigt die Eingabe

Erweiterungen	1a. Der Spieler möchte nachträglich Änderungen vornehmen 1a1. Der Spieler klickt auf den Button "Daten editieren" 1a2. Falls die Runde noch nicht beendet ist, startet Schritt 1.) mit den bereits definierten Werten im Formular erneut.
Abhängigkeiten	UC5
User Stories	24, 25, 26, 27, 28, 32 → Priorität: MUSS

ID	UC7
Name	Change request erstellen
Ziel im Kontext	Der Spieler erstellt einen Change Request
Akteure	Spieler
Trigger	Runde wurde gestartet
Essentielle Schritte	<ol style="list-style-type: none"> 1. Der Spieler klickt auf den "Change Request erstellen"-Button 2. Der Spieler definiert die Werte, welche mutiert werden sollen in einem Formular. 3. Der Spieler trägt in einem Freitextfeld seine Begründung für den Change Request ein. 4. Der Spieler sendet seinen Change Request ab.
Erweiterungen	-
Abhängigkeiten	UC5
User Stories	27 → Priorität: KANN

ID	UC8
Name	Change Request erhalten
Ziel im Kontext	Der Spielleiter erhält einen neuen Change Request
Akteure	Spielleiter
Trigger	Eine Gruppe hat einen Change Request erstellt
Essentielle Schritte	<ol style="list-style-type: none"> 1. Der Spielleiter erhält eine Benachrichtigung über einen neuen Change Request. 2. Der Spielleiter navigiert auf die Übersichtsseite mit den Change Requests in diesem Spiel 3. Der Spielleiter kann den Change Request öffnen
Erweiterungen	-

Abhängigkeiten	-
User Stories	12 → Priorität: SOLL

ID	UC9
Name	Change Request bestätigen
Ziel im Kontext	Der Spielleiter bestätigt einen offenen Change Request
Akteure	Spielleiter
Trigger	Spielleiter hat einen Change Request geöffnet
Essentielle Schritte	<ol style="list-style-type: none"> 1. Spielleiter bestätigt den Change Request mittels "Bestätigen"-Button 2. Die gewünschten Budget-Erhöhungen/Verlängerungen werden berechnet
Erweiterungen	<p>1a. Der Spielleiter möchte die Werte vor dem Bestätigen noch anpassen.</p> <p>1a1. Der Spielleiter mutiert die geforderten Werte</p> <p>1a2. Der Spielleiter erfasst eine Begründung.</p> <p>1a3. Der Spielleiter bestätigt den Change Request mittels "Bestätigen"-Button</p>
Abhängigkeiten	Extends UC 8
User Stories	13 → Priorität: KANN

ID	UC10
Name	Change Request ablehnen
Ziel im Kontext	Der Spielleiter lehnt einen offenen Change Request ab
Akteure	Spielleiter
Trigger	Spielleiter hat eine Change Request geöffnet
Essentielle Schritte	<ol style="list-style-type: none"> 1. Der Spielleiter erfasst eine Begründung 2. Der Spielleiter klickt auf den "Ablehnen" Button
Erweiterungen	-
Abhängigkeiten	Extends UC 8
User Stories	13 → Priorität: KANN

ID	UC11
----	------

Name	Runde beenden
Ziel im Kontext	Der Spielleiter beendet eine Runde
Akteure	Spielleiter
Trigger	Spielleiter möchte die Runde beenden
Essentielle Schritte	1. Der Spielleiter klickt auf den "Runde beenden"-Button in der Rundenübersicht
Erweiterungen	1a. Nicht alle Spieler haben die Eingaben bestätigt. 1a1. Der Spielleiter erhält eine Meldung. 1a2. Der Spielleiter kann die Gruppen per E-Mail informieren, dass Eingaben noch ausstehend sind 1a3. Der Spielleiter kann das Beenden forcieren, dadurch werden die Werte der Vorrunde übernommen und der Gruppe eine Strafe angerechnet.
Abhängigkeiten	UC5
User Stories	2 → Priorität: MUSS

ID	UC12
Name	Spielübersicht
Ziel im Kontext	Der Spieler erhält eine grobe Übersicht über alle Teams
Akteure	Spieler
Trigger	Der Spieler möchte sehen, wie die anderen Teams ungefähr stehen
Essentielle Schritte	1. Der Spieler klickt auf den "Statistik"-Button 2. Der Spieler sieht eine grobe Übersicht über alle Teams
Erweiterungen	-
Abhängigkeiten	UC4
User Stories	30 → Priorität: KANN

ID	UC13
Name	Dashboard aufrufen
Ziel im Kontext	Der Spielleiter erhält ein detailliertes Dashboard, welches die Eingaben der aktuellen sowie der vergangenen Runden der Spieler zeigt

Akteure	Spielleiter
Trigger	Der Spielleiter möchte genaue Informationen über die aktuellen Eingaben der Spieler sehen.
Essentielle Schritte	<ol style="list-style-type: none"> 1. Der Spielleiter klickt auf den "Dashboard"-Button 2. Der Spielleiter sieht die Eingaben der aktuellen Runde
Erweiterungen	<p>2a. Der Spielleiter möchte ältere Daten sehen</p> <p>2a1. Der Spielleiter klickt auf den "Vorgängige Runden"-Button</p> <p>2a2. Der Spielleiter sieht die Daten der betreffenden Runde</p>
Abhängigkeiten	UC4
User Stories	5 → Priorität: SOLL

ID	UC14
Name	Reports einsehen
Ziel im Kontext	Der Spielleiter kann detaillierte Reports der Gruppen einsehen
Akteure	Spielleiter
Trigger	Der Spielleiter möchte Reports einsehen
Essentielle Schritte	<ol style="list-style-type: none"> 1. Der Spielleiter klickt auf den "Reports"-Button 2. Der Spielleiter sieht folgende Charts <ol style="list-style-type: none"> a. Kostenentwicklung b. Burn-Down-Chart
Erweiterungen	-
Abhängigkeiten	UC4
User Stories	36 → Priorität: SOLL

ID	UC15
Name	Reports exportieren
Ziel im Kontext	Der Spielleiter kann einen oder mehrere Reports exportieren
Akteure	Spielleiter
Trigger	Der Spielleiter ist auf der "Reports"-Seite und möchte die Daten exportieren.
Essentielle Schritte	<ol style="list-style-type: none"> 1. Der Spielleiter klickt auf den "Export"-Button 2. Der Spielleiter wählt die gewünschten Daten und Charts aus.

Erweiterungen	-
Abhängigkeiten	UC14
User Stories	8 → Priorität: SOLL

ID	UC16
Name	CRUD Event
Ziel im Kontext	Der Spielleiter definiert oder mutiert ein Event, welches in einer Spielrunde eintreten kann.
Akteure	Spielleiter
Trigger	Der Spielleiter möchte ein Event definieren oder mutieren
Essentielle Schritte	<ol style="list-style-type: none"> 1. Der Spielleiter klickt auf den "Events"-Button 2. Der Spielleiter klickt auf den "Neues Event erstellen"-Button 3. Der Spielleiter definiert einen Namen 4. Der Spielleiter definiert eine Beschreibung 5. Der Spielleiter definiert die Auswirkung des Events. 6. Der Spielleiter speichert das Event
Erweiterungen	<p>2a. Der Spielleiter möchte ein Event löschen 2a1. Der Spielleiter klickt bei dem betreffenden Event auf das Papierkorb-Symbol</p> <hr/> <p>2b. Der Spielleiter möchte ein bestehendes Event bearbeiten. 2b1. Der Spielleiter klickt beim betreffenden Event auf das Stift-Symbol 2b2. Weiter bei Schritt 3 mit vorausgefüllten Werten</p>
Abhängigkeiten	-
User Stories	14 → Priorität: SOLL

ID	UC17
Name	Benachrichtigung versenden (E-Mail und Notification)
Ziel im Kontext	Der Spielleiter versendet einer oder mehreren Gruppen eine Benachrichtigung
Akteure	Spielleiter
Trigger	Spielleiter möchte Benachrichtigung versenden
Essentielle Schritte	<ol style="list-style-type: none"> 1. Der Spielleiter klickt auf den "Benachrichtigungen"-Button 2. Der Spielleiter trägt den Betreff ein

	<ol style="list-style-type: none"> 3. Der Spielleiter trägt im Freitextfeld die gewünschte Benachrichtigung ein 4. Der Spielleiter wählt die gewünschten Gruppen aus 5. Der Spielleiter wählt die gewünschte Benachrichtigungsart "E-Mail" und/oder "Notifications" aus.
Erweiterungen	-
Abhängigkeiten	-
User Stories	6, 7 → Priorität: SOLL

ID	UC18
Name	Benachrichtigung bei ausstehenden Eingaben
Ziel im Kontext	Der Spielleiter informiert alle Gruppen, welche noch keine Eingabe bestätigt haben.
Akteure	Spielleiter
Trigger	Es gibt noch Gruppen, welche die Eingaben nicht bestätigt haben.
Essentielle Schritte	<ol style="list-style-type: none"> 1. Der Spielleiter klickt in der Rundenübersicht auf den "Gruppen mit ausstehenden Eingaben benachrichtigen"-Button
Erweiterungen	1a. Der Spielleiter klickt den "Gruppen mit ausstehenden Eingaben benachrichtigen"-Button in der "Runde beenden"-Übersicht
Abhängigkeiten	UC5
User Stories	31 → Priorität: KANN

ID	UC19
Name	CRUD Spieler
Ziel im Kontext	Einen Spieler erstellen, mutieren oder löschen
Akteure	Spielleiter, Spieler
Trigger	
Essentielle Schritte	<p>Registrieren</p> <ol style="list-style-type: none"> 1. Der Spieler registriert sich über die Hauptseite mit einer E-Mail-Adresse. 2. Er definiert einen Nickname <hr/> <p>Erfassen</p>

	<ol style="list-style-type: none"> 1. Der Spielleiter erfasst einen Spieler mit E-Mail-Adresse 2. Der Spieler erhält eine E-Mail mit einem "Bestätigen"-Link, darüber kann er das Passwort setzen. <hr/> <p>Löschen durch Spielleiter</p> <ol style="list-style-type: none"> 1. Der Spielleiter löscht einen Spieler über die Benutzerübersicht. 2. Der Spieler wird aus <u>sämtlichen</u> Spielen entfernt und der Account des Spielers wird gelöscht. <hr/> <p>Löschen durch Spieler</p> <ol style="list-style-type: none"> 1. Der Spieler löscht seinen Account über die Einstellungen 2. Der Spieler wird aus sämtlichen Spielen entfernt und der Account des Spielers wird gelöscht. <hr/> <p>Bearbeiten</p> <ol style="list-style-type: none"> 1. Der Spieler mutiert seine E-Mail-Adresse, Passwort und/oder Nickname über die Einstellungen
Erweiterungen	-
Abhängigkeiten	-
User Stories	34, 40 → Priorität: MUSS

ID	UC20
Name	Spieler eröffnet eine Gruppe
Ziel im Kontext	Der Spieler erstellt selbständig eine neue Gruppe
Akteure	Spieler
Trigger	Der Spieler möchte eine Gruppe erstellen
Essentielle Schritte	<ol style="list-style-type: none"> 1. Der Spieler klickt auf den "Gruppe erstellen"-Button 2. Der Spieler trägt einen Gruppennamen ein 3. Andere Spieler treten bei, bis die erforderliche Anzahl an Gruppenmitglieder erreicht ist 4. Beliebiger Spieler in der Gruppe bestätigt Gruppe
Erweiterungen	2a. Gruppenname existiert bereits 2a1. Der Spieler erhält eine Warnmeldung
Abhängigkeiten	UC4
User Stories	17, 19, 23 → Priorität: SOLL

ID	UC21
----	------

Name	Spieler tritt einer Gruppe bei
Ziel im Kontext	Der Spieler tritt einer bestehenden Gruppe bei
Akteure	Spieler
Trigger	Spieler möchte einer Gruppe beitreten
Essentielle Schritte	1. Der Spieler wählt in der Gruppenübersicht die gewünschte Gruppe aus und drückt auf den "Beitreten"-Button
Erweiterungen	1a. Gruppe ist bereits voll 1a1. Der "Beitreten"-Button ist deaktiviert
Abhängigkeiten	Extends UC20
User Stories	18 → Priorität: KANN

ID	UC22
Name	CRUD Spielleiter
Ziel im Kontext	Ein bestehender Spielleiter erstellt einen neuen Spielleiter
Akteure	Spielleiter
Trigger	Spielleiter möchte einen neuen Spielleiter erfassen
Essentielle Schritte	1. Der Spielleiter klickt in den Einstellungen auf den "Weiteren Spielleiter hinzufügen"-Button 2. Der Spielleiter trägt die E-Mail Adresse des neuen Spielleiters ein. 3. Der Spielleiter bestätigt mit dem "Bestätigen"-Button
Erweiterungen	1a. Der Spielleiter möchte einen bestehenden Spielleiter löschen 1a1. Der Spielleiter klickt auf den
Abhängigkeiten	
User Stories	39 → Priorität: KANN

ID	UC23
Name	Spiel editieren oder löschen
Ziel im Kontext	Das Spiel wird editiert oder gelöscht
Akteure	Spielleiter

Trigger	Der Spielleiter möchte ein noch nicht gestartetes Spiel bearbeiten oder löschen
Essentielle Schritte	<ol style="list-style-type: none"> 1. Der Spielleiter wählt aus der Spieleübersicht das gewünschte Spiel aus 2. Der Spielleiter editiert die gewünschten Werte 3. Der Spielleiter speichert das Spiel mit den neuen Werten
Erweiterungen	<p>2a. Der Spielleiter möchte das Spiel löschen</p> <p>2a1. Der Spielleiter klickt auf das Papierkorb-Symbol</p>
Abhängigkeiten	UC1
User Stories	22 → Priorität: SOLL

ID	UC24
Name	CRUD Mitarbeitertyp
Ziel im Kontext	Ein Mitarbeitertyp wird erstellt, bearbeitet oder gelöscht
Akteure	Spielleiter
Trigger	Der Spielleiter möchte einen Mitarbeitertyp erstellen, bearbeiten oder löschen.
Essentielle Schritte	<ol style="list-style-type: none"> 1. Der Spielleiter öffnet die Einstellungen. 2. Der Spielleiter öffnet den Mitarbeiter-Abschnitt 3. Der Spielleiter klickt auf den "Hinzufügen"-Button 4. Der Spielleiter trägt folgende Angaben ein: <ol style="list-style-type: none"> a. Namen b. Funktionsbeschreibung c. Effizienz 5. Der Spielleiter speichert die Eingabe
Erweiterungen	<p>3a. Der Spielleiter möchte einen bestehenden Mitarbeitertyp mutieren</p> <p>3a1. Der Spielleiter klickt beim entsprechenden Mitarbeitertyp auf das "Stift"-Symbol.</p> <p>3a2. Der Spielleiter aktualisiert folgende Werte:</p> <ol style="list-style-type: none"> a. Namen b. Funktionsbeschreibung c. Effizienz <p>3a3. Der Spielleiter speichert die Eingaben.</p> <hr/> <p>3b. Der Spielleiter möchte einen Mitarbeiter löschen.</p> <p>3b1. Der Spielleiter klickt bei dem entsprechenden Mitarbeiter auf das "Papierkorb"-Symbol</p>
Abhängigkeiten	UC1

User Stories	33 → Priorität: KANN
--------------	----------------------

ID	UC25
Name	CRUD Aufwand
Ziel im Kontext	Ein Aufwand wird erstellt, bearbeitet oder gelöscht.
Akteure	Spielleiter
Trigger	Der Spielleiter möchte einen Aufwand erfassen oder einen bestehenden bearbeiten.
Essentielle Schritte	<ol style="list-style-type: none"> 1. Der Spielleiter öffnet die Einstellungen 2. Der Spielleiter öffnet die Aufwandsübersicht 3. Der Spielleiter klickt auf den "Aufwand hinzufügen"-Button 4. Der Spielleiter trägt die Bezeichnung des Aufwands hinzu. 5. Der Spielleiter speichert den Aufwand.
Erweiterungen	<p>3a. Der Spielleiter möchte einen Aufwand löschen</p> <p>3a1. Der Spielleiter klickt auf das Papierkorb-Symbol bei einem Aufwand in der Liste.</p> <hr/> <p>3b. Der Spielleiter möchte einen Aufwand bearbeiten.</p> <p>3b1. Der Spielleiter klickt auf einen Aufwand in der Liste.</p> <p>3b2. Der Spielleiter bearbeitet den Text.</p> <p>3b3. Der Spielleiter speichert den Aufwand.</p>
Abhängigkeiten	UC1
User Stories	33, 37 → Priorität: MUSS

5.3. Erweiterungsmöglichkeiten

Im Rahmen der Live-Tests im „Projekt- & Qualitätsmanagement“ Unterricht haben wir diverse Erweiterungsmöglichkeiten festgestellt. Nachfolgend eine Liste mit den Überlegungen. Diese Überlegungen können in einem nächsten Schritt in User Stories umgeschrieben werden und in den Product Backlog aufgenommen werden.

Core Features:

- Nicht nur Mitarbeitertypen, sondern einzelne Mitarbeiter einbauen (Hans, Max, Franz, Bruno, etc.). Benötigt Änderungen am Domain Model.
- Transfermarkt für Mitarbeiter
- Trainingslevel: Abzug für jeden Mitarbeiter mehr, den man einstellt (Koordinationsaufwand / Kommunikationsaufwand).
- Trainingslevel: Zusätzlichen Abzug für jeden 10. Mitarbeiter mehr den man einstellt. Hintergrund ist die Idee, dass ein Development Team maximal 10 Leute umfassen sollte. Jedes mal wenn man diese Grenze überschreitet, muss man ein neues Team bilden, was zu Kosten für Büro, Bürogebäude, evt. gar Auslagerung in eine andere Stadt oder Land nachsichzieht. Alternativ: statt Effekt auf Scope, die Kostenstruktur dieser Gruppe ändern.
- Man könnte es sich überlegen, ob interdisziplinäre Teams einen Bonus erhalten sollten
- Es sollte auch möglich sein, eine Low-cost/Outsourcing-Strategie mit Erfolg zu fahren, sofern das Projekt hervorragend gemanaget wird.
- Das Konzept der Risikoreduktion ist sehr abstrakt¹⁵. Man könnte sich überlegen, ob der Spielleiter zu Beginn eine Liste mit konkreten Risiken anlegen soll. Danach können die Gruppen selbst Risiken erfassen und in diese Risiken investieren. Dadurch hätte man ein zusätzliches Lernziel „Wir als Gruppe erkennen Risiken eines spezifischen IT-Projekts“.
- Eine Gruppe wählt ihr Vorgehensmodell für das IT-Projekt selbst (z.B. Scrum, KanBan, Wasserfall)
- Eine Gruppe ordnet für eine Runde Überstunden an: Die Produktivität wäre dann höher, aber dafür sinkt die Motivation und die Fehleranfälligkeit steigt (Risiko steigt wieder).
- Ausbildungslevel nur für eingesetzte Mitarbeiter erhöhen (i.e. es soll verhindert werden, dass man zu Beginn keine Mitarbeiter einsetzt, aber in die Ausbildung investiert und trotzdem den Ausbildungseffekt erhält).
- Bonus/Malus für lebenslanges Lernen vs. 1 Mal sehr viel ausbilden

¹⁵Dies hat sich in den Live-Tests durch diverse Rückfragen gezeigt.

- Als Spieler aus einem Spiel austreten, so lange das Spiel noch nicht gestartet ist
- Reale IT-Projekte, welche schiefgelaufen oder zumindest nicht optimal verlaufen sind, als Szenarien vorgeben können. Optimalerweise könnten diese dann den verantwortlichen Personen eines Projekts vorgegeben werden, sodass man prüfen kann, ob man aus den Fehlern gelernt hat.

Weitere Features und kleinere Anpassungen:

- Mobile-GUI beziehungsweise kleinere Desktop-Screen Grössen unterstützen.
- Für Gruppen: Burndown-Chart analog wie Admin (einfach nur für die eigene Gruppe)
- Admin: Projektleiterpensum aggregiert über alle Runden pro Gruppe vergleichen können
- Passwort-Reset-Funktion
- Passwort-Change-Funktion für Admin einbauen
- Usability: Rückmeldung, ob ein Wert übernommen wurde (gespeichert wurde) analog den Fehlermeldungen
- Admin: Slider für Wirksamkeit Risikoreduktion default auf 100% stellen und evt. mehr als 100% erlauben (z.B. 0-200%)
- Admin-Dashboard: Alle Parameter des Games anzeigen während dem Spiel
- Header: Titel des Spiels: wenn der Titel lange ist, gibt es einen Zeilenumbruch: Beispiel Titel: „Spiel: HS 2019 Übungsgruppe 1 - Test“
- Beschreibung/Spielinformationen des Spiels anzeigen - entweder während man auf das Spielstart wartet, oder als separate Infoseite. Zudem Infos angeben, wie sich das Training auf die Mitarbeiter auswirkt.

5.4. Domain Model

Während der Elaboration-Phase sind wir mehrmals systematisch die User Stories durchgegangen, um zu prüfen, ob das aktuelle Domain Model die funktionalen Anforderungen erfüllen kann. Das wir dies diszipliniert gemacht haben, war sehr wichtig. Bei einigen Durchläufen hatten wir festgestellt, dass das Domain Model noch nicht alle User Stories abbilden kann und Änderungen notwendig sind.

Ausserdem haben wir intensiv diskutiert, wie viel Komplexität wir uns mit dem Domain Model in der 14-wöchigen Studienarbeit zumuten können. So haben wir uns zum Beispiel explizit dagegen entschieden, eine eigene Klasse für Mitarbeiterinstanzen umzusetzen - im Wissen darum, dass dann später für die Umsetzung eines Transfermarkts für Mitarbeiter eine Änderung am Domain Model notwendig sein wird.

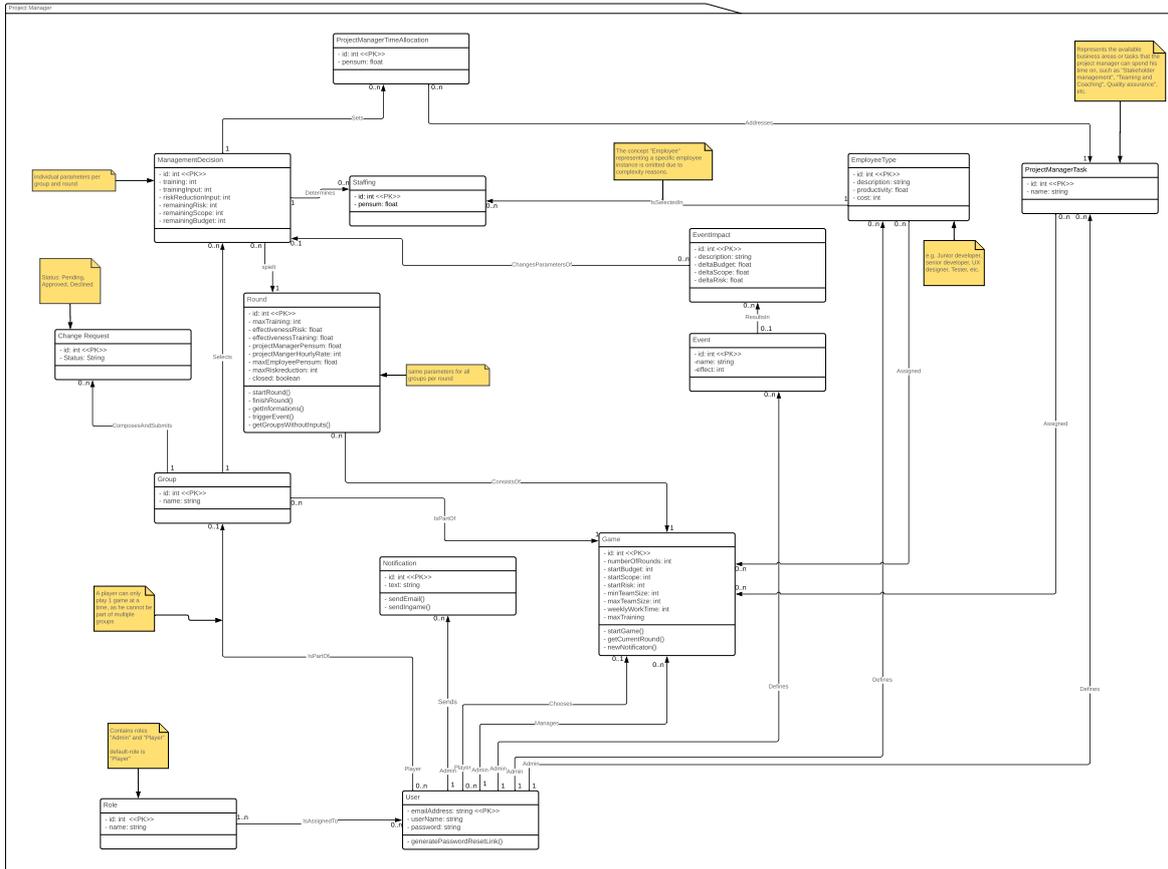


Abbildung 9: Domain Model. Zu beachten sind auch die UML-Kommentare direkt in der Abbildung. Bis auf die Tabellen „Notification“ und „Event“ konnten alle Tabellen implementiert werden.

Andererseits haben wir vorausschauend auf Hinweis unseres Betreuers eine Tabelle „roles“ eingebaut, in der später neben den aktuellen Rollen „admin“ und „user“ noch weitere Rollen (z. B. „systemadmin“) hinzugefügt werden können.

Insgesamt haben wir uns in der Elaboration-Phase intensiv und lange mit dem Domain Model beschäftigt. Entsprechend mussten wir in der Construction-Phase nur noch marginale Änderungen am Domain Model vornehmen. Beispielsweise haben wir je eine Beziehung von „EmployeeType“ respektive „ProjectManagerTask“ zu „Game“ hinzugefügt. Zudem war das Konzept des Change Requests und der Eventauswirkungen noch nicht ganz ausgereift bei End of Elaboration, sodass wir noch 2 Tabellen hinzufügen mussten.

Bei der Darstellung des Domain Model haben wir darauf geachtet, dass (soweit möglich) die Tabelle mit der 1er-Beziehung unterhalb der n-Beziehung steht, m-zu-n-Beziehungen und 1-zu-1-Beziehungen auf der selben Ebene stehen. Dies hat die Darstellung wesentlich vereinfacht.¹⁶ Das dabei entstandene Domain Model ist in der Abbildung 9 ersichtlich.

5.5. Nicht-funktionale Anforderungen

5.5.1. Zu unterstützende Szenarien

Um die nicht-funktionalen Anforderungen besser definieren zu können, haben wir uns zwei Szenarien überlegt, welche wir mindestens unterstützen möchten:

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 Freitagabend, Spielleiter beendet Runde am Samstag, Resultate am Montag verfügbar)

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

Zielgruppe

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

Spieler Haben mindestens rudimentäre EDV-Anwenderkenntnisse

5.5.2. Priorisierung

Basierend auf diesem Hintergrund haben wir die Anforderungskategorien gemäss ISO/IEC 25010 aufgelistet. Jedes Projektmitglied hat den einzelnen Kategorien eine Priorität ver-

¹⁶Dies ist ein Tipp aus der Vorlesung „SE1“.

geben.¹⁷ Daraus konnten wir ein Spider-Diagramm erstellen (siehe Abbildung 10). Das Spider-Diagramm hat uns geholfen, uns auf die wesentlichen Punkte bei der Spezifikation der nicht-funktionalen Anforderungen zu konzentrieren.

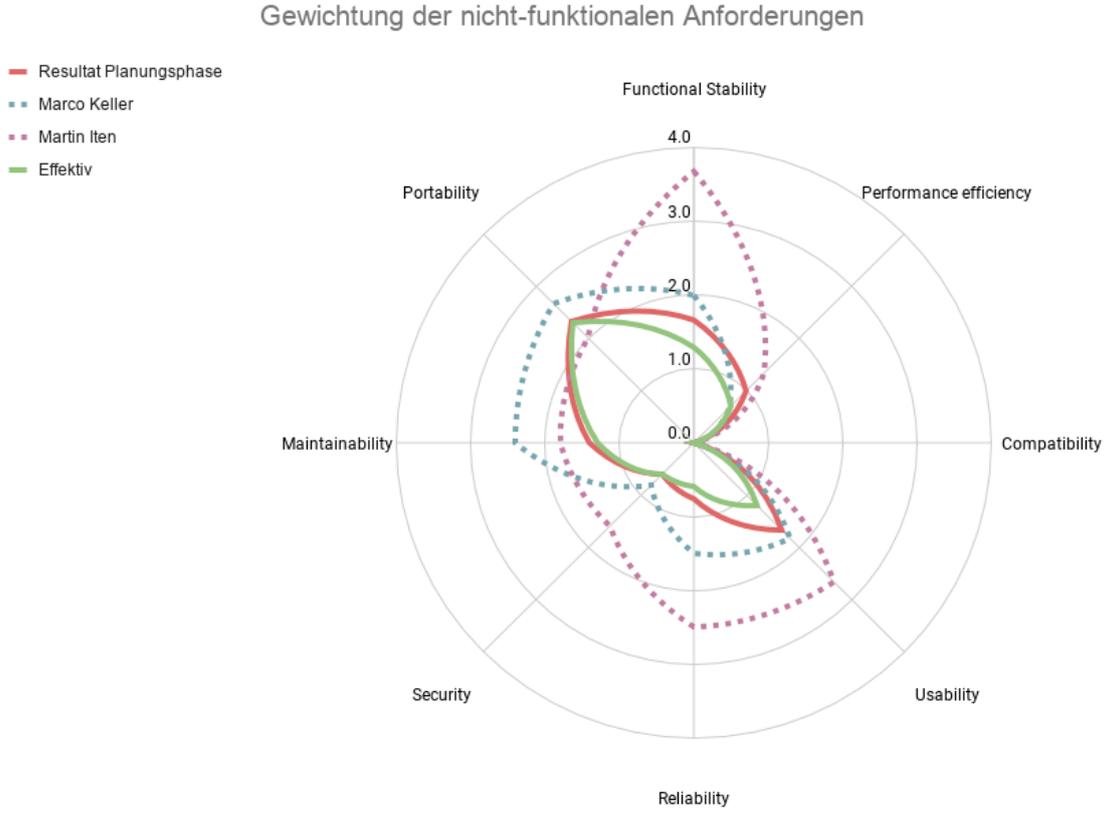


Abbildung 10: Die Gewichtung der nicht-funktionalen Anforderungen der Projektmitglieder, dem Resultat der Planungsphase und der effektiv implementierten Lösung.

¹⁷Siehe die Rohdaten im Anhang auf Abbildung C

5.5.3. Nicht-funktionale Anforderungen

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

Merkmal	Beschreibung
Functional Stability	<p><i>Functional Completeness:</i></p> <ul style="list-style-type: none"> • Alle User stories, die mit Priorität "MUSS" gekennzeichnet sind, sind implementiert. <p><i>Functional Correctness:</i></p> <ul style="list-style-type: none"> • Das Budget, der Scope und die Restrisiken einer Runde eines Teams werden basierend auf den Stellschrauben (Aufwände, Events, Wirksamkeiten, etc.) korrekt berechnet. • Die Rangliste am Ende des Spiels zeigt die korrekte Reihenfolge der Teams basierend auf dem Projektziel des Spiels. <p><i>Functional Appropriateness (de: Angemessenheit):</i></p> <p>-</p>
Performance efficiency	<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"> • Antwortzeiten auf alle "CRUD Spielleiter"-Aktionen betragen maximal 1 Minute (Szenario 1 und Szenario 2) (bei 1 User im System online) • 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) • "CRUD Spieler": Nachdem der Spielleiter einen Spieler erfasst hat, wird das E-Mail innert 10 Sekunden (Szenario 1; bei 19 User im System online) respektive 20 Sekunden (Szenario 2; bei 127 User im System online) verschickt. • Nachdem der Spielleiter eine "CRUD Gruppe"-Aktion bestätigt hat, beträgt die Antwortzeit maximal 20 Sekunden (Szenario 1; bei 19 User im System online) respektive 30 Sekunden (Szenario 2; bei 127 User im System online) • Nachdem der Spielleiter das Spiel gestartet hat, beträgt die Antwortzeit maximal 1 Minute (Szenario 1; bei 19 User im System online) respektive 2 Minuten (Szenario 2; bei 127 User im System online)

	<ul style="list-style-type: none"> • Nachdem der Spielleiter eine Runde 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) • 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) • Nachdem der Spielleiter eine Runde beendet hat, dauert das Berechnen der Rundenresultate maximal 1 Minute (Szenario 1; bei 19 User im System online) respektive 10 Minuten (Szenario 2; bei 127 User im System online) <p><i>Resource utilization:</i> -</p> <p><i>Capacity:</i></p> <ul style="list-style-type: none"> • Es können mindestens 126 Spieler (maximale Teilnehmerzahl des HSR-Moduls "Projekt- und Qualitätsmanagement") und 1 Spielleiter gleichzeitig im System eingeloggt sein.
Compatibility	<p><i>Co-existence:</i> -</p> <p><i>Interoperability:</i> -</p>
Usability	<p><i>Appropriateness recognizability:</i></p> <ul style="list-style-type: none"> • Für noch nicht registrierte Benutzer ist eine Kurz-Demo mit Screenshots aus einem Spiel auf der Startseite verfügbar. <p><i>Learnability:</i></p> <ul style="list-style-type: none"> • Das GUI ist von der Zielgruppe ohne besondere Schulung verwendbar. • Die Applikation verfügt über ein Benutzerhandbuch. <p><i>Operability:</i></p> <ul style="list-style-type: none"> • Es ist kein vertikales Scrollen nötig bei Zoom-Level 100% für die optimierten Geräte und Auflösungen (siehe "Look & Feel") - Ein User erfasst (maximal) 5 benutzerdefinierte Aufwände in unter 10 Eingabeaktionen.

	<ul style="list-style-type: none"> - Ein User erfasst 1 benutzerdefiniertes Event in unter 5 Eingabeaktionen. - Ein User 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) - Spielzugeingabe: Ein User hat Einsicht in die Eingaben seiner Gruppe aus den vorherigen Runden in unter 1 Eingabeaktion - Spielzugeingabe: Ein User hat Einsicht in den Zwischenstand der anderen Teams in unter 1 Eingabeaktion - Spielzugeingabe, Spielzugeingaben aus vorherigen Runden und Zwischenstand der Teams sind zusammen auf 1 Screen einsehbar für einen User. <p><i>User error protection:</i></p> <ul style="list-style-type: none"> • Alle Eingabefelder werden entsprechend der Art des Transaktionsfeldes, das sie repräsentieren, validiert. <p><i>User interface aesthetics ("Look & Feel"):</i></p> <ul style="list-style-type: none"> • Es wird ein spielerisches GUI verwendet, um dem Spieler einen einfachen Zugang zum Spiel zu ermöglichen. Es zeichnet sich durch Grafiken und klare Strukturen aus. • "Project Manager" wird für die folgenden Geräte und Auflösungen optimiert und getestet: <ul style="list-style-type: none"> ○ Desktop: Auflösung 1920 x 1080 • Für alle die optimierten Geräte und Auflösung sind folgende Punkte erfüllt: <ul style="list-style-type: none"> ○ wirkt vertrauenserweckend (Grammatik, Rechtschreibung, Farben (siehe Material-Design-Anforderung), keine verpixelten Bilder, keine blinkenden Elemente) ○ Controls sind intuitiv angeordnet ○ Zweck (Hauptfunktion) der Seite ist klar • Material Design-Anforderung: <ul style="list-style-type: none"> ○ Eine Seite hat 1 Primary color (und Schattierung davon) und 1 Accent color (und Schattierung davon) • <p><i>Accessibility:</i> -</p>
--	---

<p>Reliability</p>	<p><i>Maturity:</i> -</p> <p><i>Availability:</i></p> <ul style="list-style-type: none"> • Das System ist zu 90% der operativen Zeit verfügbar. Dies entspricht einer maximalen Downtime von 72 Stunden (3 Tagen) bei 30 Tagen pro Monat. <p><i>Fault tolerance:</i> -</p> <p><i>Recoverability:</i></p> <ul style="list-style-type: none"> • Täglich wird 1 Backup der Datenbank erstellt. • Wiederherstellung bei Systemausfall: Da wir nur 1 tägliches Backup machen, ist nur der Stand vom Vortag wiederherstellbar. Änderungen des laufenden Tages sind verloren. • Auf ein Logging der Datenbanktransaktionen wird verzichtet. Angefangene Transaktionen sind somit bei einem Systemausfall nicht wiederherstellbar..
<p>Security</p>	<p><i>Confidentiality:</i></p> <ul style="list-style-type: none"> • Die Applikation speichert keine Personendaten. • Die Authentifizierung erfolgt via E-Mail-Adresse und Passwort. • Das Passwort ist in der Datenbank als Salted Hash abgelegt. • Benutzerrechte: Die Spieler haben keinen Zugriff auf Bereiche, die nur der Spielleiter verwalten darf. <p><i>Integrity:</i></p> <ul style="list-style-type: none"> • XSS: Sämtliche Formulare sind gegen XSS-Angriffe abgesichert • Konfigurationen: Die Konfigurationsdateien sind nicht öffentlich erreichbar. • 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)). • Passwortstärke: Ein Passwort besteht mindestens aus 8 Zeichen • Passwortstärke: Ein Passwort besteht mindestens aus 1 Grossbuchstaben, 1 Kleinbuchstaben, 1 Zahl und 1 Sonderzeichen.

	<p><i>Non-repudiation (de: Nicht-Abstreitbarkeit): -</i></p> <p><i>Accountability: -</i></p> <p><i>Authenticity: -</i></p>
<p>Maintainability</p>	<p><i>Modularity:</i></p> <ul style="list-style-type: none"> ● Das User-Handling (Spieler registrieren, Gruppen registrieren, Passwort ändern, etc.) ist vom Game-Handling (Runde spielen) getrennt. ● Die Architektur besteht aus mindestens 3 getrennten logischen Schichten: Presentation (UI), Business Logic und Persistence (Datenbank) <p><i>Reusability: -</i></p> <p><i>Analysability: -</i></p> <p><i>Modifiability:</i></p> <ul style="list-style-type: none"> ● Die Applikation verfügt über eine Infrastruktur, um mindestens die Stages “build”, “test”, “deploy” (entweder auf Staging- oder Production-Server) zusammen automatisch auszuführen. ● Die Applikation verfügt über eine Infrastruktur, um automatisch von der Staging- auf die Production-Umgebung zu deployen. ● Die Software wird in einem Paket auf den Host deployed und gestartet. ● Beim Betrieb der Applikation in der ausgelieferten Version fällt kein Wartungsaufwand an. <p><i>Testability:</i></p> <ul style="list-style-type: none"> ● Das Main Success-Szenario aller implementierten Use Cases wird mittels Tests überprüft. ● Zu den Tests werden Testspezifikationen und Testprotokolle erstellt.

Portability	<p><i>Adaptability:</i></p> <ul style="list-style-type: none">• Die Applikation ist anbieterunabhängig: Das heisst, dass die Applikation auf jedem Server lauffähig ist, auf dem Node.js ausgeführt werden kann. <p><i>Installability:</i></p> <ul style="list-style-type: none">• Die Applikation ist mit weniger als 10 Kommandozeilenaktionen installierbar.• Die Applikation verfügt über ein Readme, indem die Installationsschritte detailliert beschrieben werden. <p><i>Replaceability:</i> -</p>
--------------------	--

5.5.4. Erläuterungen

Einige Entscheidungen möchten wir noch näher erläutern.

- Performance
 - Die wichtigste Eigenschaft ist die „umgehende“ Erfassung und Speicherung der Inhalte während des Spiels. Die Auswertung und Darstellung der Resultate hingegen hat eine grössere Zeittoleranz.
 - Erklärung, weshalb nur für gewisse Use Cases (Aktionen) Performancegarantien gegeben werden:

Um ein Spiel in einer Basisversion spielen zu können, sind folgende Use Cases erforderlich:
 - * UC22: CRUD Spielleiter
 - * UC1: Spiel erstellen
 - * UC19: CRUD Spieler
 - * UC2: CRUD Gruppe
 - * UC4: Spiel starten
 - * UC5: Runde starten
 - * UC6: Runde spielen
 - * UC11: Runde beenden
- Usability: Erklärung, wieso auf Desktop ausgerichtet und nicht mobile-first:
 - weil die Applikation auf das HSR-Modul ausgerichtet ist: z.B. Übungslektion, dort hat in der Regel jeder Teilnehmer einen Laptop bei sich
 - Schwierigkeit, vergangene Eingaben anzuzeigen und Stand der anderen Teams auf dem selben Screen anzuzeigen wie das Eingabeformular auf Mobile aus Platzgründen
- Security
 - Die Plattform und die Daten sollen ausreichend vor unbefugten Zugriffen geschützt sein, es handelt sich jedoch um keine Hochsicherheits- und Hochverfügbarkeitsapplikation.
 - Wir reduzieren sämtliche persönliche Daten des Users auf ein Minimum. Dafür werden wir die Security Measures der Software nicht so umfangreich gestalten.
- Erklärungen, wieso gewisse Qualitätsattribute nicht garantiert werden:

- Generell: Die spezifizierten Qualitätsattribute zielen darauf ab, „Project Manager“ in einer Basisversion vernünftig spielen zu können, sodass ein Lerneffekt erreicht werden kann.
- Usability: Accessibility: „Project Manager“ ist ein Teamspiel, in welchem sich individuelle Spieler zu einem Team zusammenschliessen. Sollte ein Spieler Schwierigkeiten haben, können die Mitspieler helfen.
- Compatibility: „Project Manager“ ist eine eigenständige Applikation. Entsprechend bestehen keine Verbindungen zu externen Systemen.

5.6. Datenmodell

Abbildung 11 zeigt das Datenmodell im SQL-Schema.

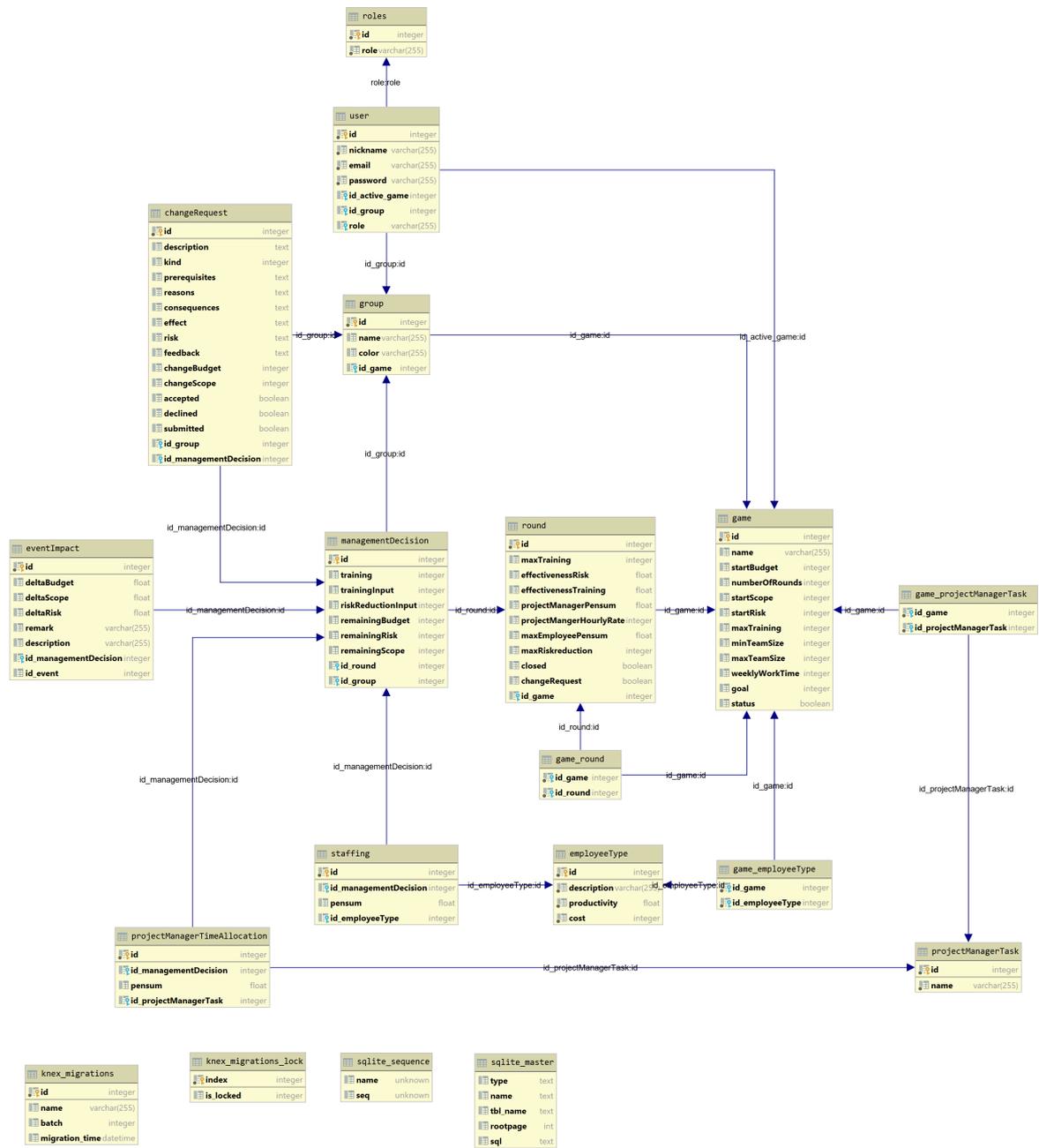


Abbildung 11: Datenmodell. Screenshot aus JetBrains DataGrip-Tool.

5.7. Logische Architektur

Wir haben in der Abbildung 12 bewusst darauf verzichtet, die einzelnen Funktionen in der Controller-Schicht und der Repository-Schicht darzustellen. Dies hätte die Grafik unnötig aufgeblasen. Die relevantesten Funktionen sind in der Service-Schicht enthalten.

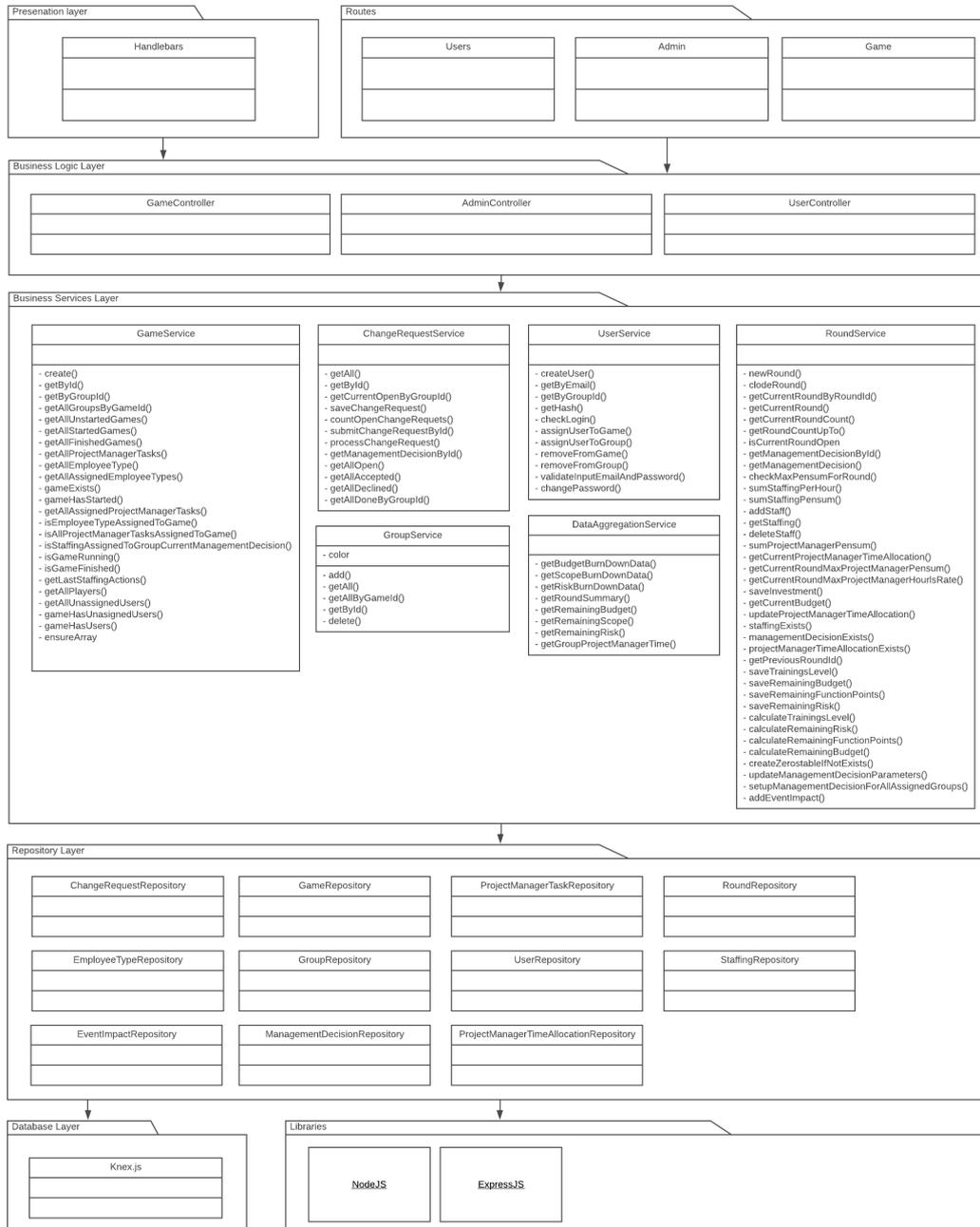


Abbildung 12: Logische Architektur: Schichtendiagramm.

5.8. Deployment-Diagramm

Die Instanzen von „Project-Manager“ laufen jeweils in eigenen Docker Container. Für die HTTP Verbindung zum Benutzer sorgt Traefik, ein Proxy Server, welcher die verschiedenen Domains zu dem jeweiligen Docker Container routet. Traefik kümmert sich ebenfalls um die SSL-Verbindung.

Auf dem GitLab Server wird der Quellcode von „Project-Manager“ verwaltet und die Docker Images gebaut. Diese werden auf den Docker Hub gepusht. Der virtuelle HSR Server kann diese Images anschliessend konsumieren und die Docker Container damit starten. Wir haben dies in der Abbildung 13 als Deployment-Diagramm dargestellt.

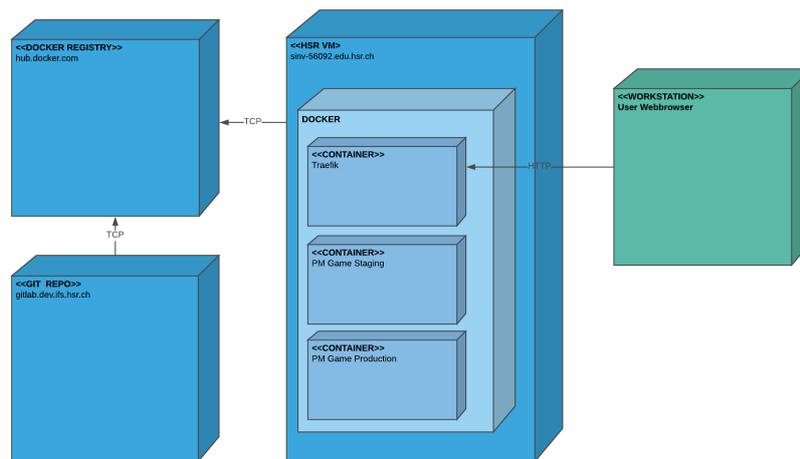


Abbildung 13: Das Deployment-Diagramm zeigt das Zusammenspiel der verschiedenen Services

5.9. Gui-Skizzen

Wir haben für „Project-Manager“ von Hand Gui-Skizzen angefertigt. Diese wurden während der Entwicklung für die Umsetzung des Gui's verwendet. In Abbildung 14 sind einige davon zu sehen.

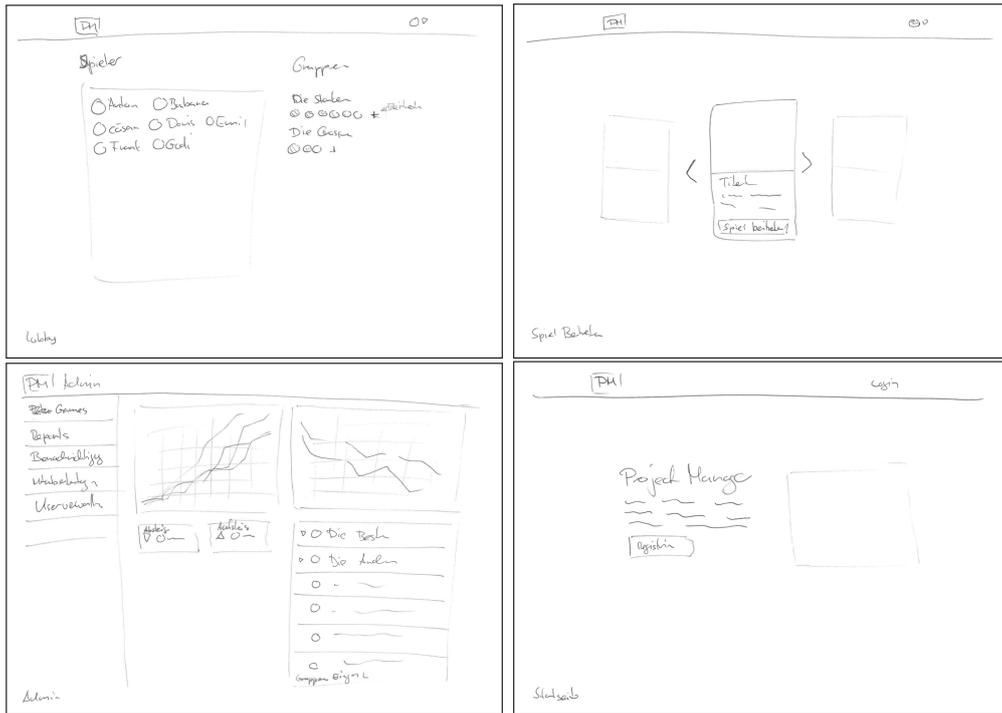


Abbildung 14: Links oben: Gruppenerstellung; Rechts oben: Die Spieleauswahl; Links unten: Das Admin Dashboard; Rechts unten: die Startseite

5.10. Architekturentscheidungen: Diskussion

Item (<i>Anforderung</i>)	Beschreibung
knexjs <i>Portability</i>	Wir setzten knexjs ein, um flexibel von einer SQLite-Datenbank auf eine MySQL- oder Postgres-Datenbank wechseln zu können. So konnten wir zu Beginn ohne grossen Installationsaufwand mit dem Entwickeln auf einer SQLite-Datenbank beginnen. Geplant war, dass wir in der Mitte der Projektlaufzeit dann auf eine MySQL-Datenbank umsteigen. Leider ist dies jedoch nicht gelungen (siehe dazu mehr unter Lösungskonzept & Umsetzung). Entsprechend läuft die Applikation nachwievorder auf einer SQLite-Datenbank.
Node.js <i>Portability</i>	Da die Applikation flexibel und unabhängig eines spezifischen Anbieters betrieben werden können sollte, ist die Auswahl der zu verwendenden Technologien auf PHP und Node.js beschränkt. Diese haben eine genug grosse Verbreitung, so dass für eine entsprechende Applikation vergleichsweise einfach eine Hosting-Plattform dafür gefunden werden kann. Wir haben uns schlussendlich für Node.js entschieden, da wir uns darin besser gefestigt fühlten.
TypeScript <i>Functional Stability, Reliability</i>	Wir setzten auf TypeScript, damit wir eine typensichere Applikation haben.
Wieso keine Model-Klassen? <i>Maintainability</i>	In einem ersten Entwurf war geplant, die Objekte aus der Datenbank jeweils vor der Verwendung in TypeScript-Objekte in Form von Modelklassen umzuwandeln. Jedoch zeigte sich, dass die Applikation (vor allem zu Beginn) eine Datenbankapplikation war: Werte wurden vom Frontend übernommen und direkt in die Datenbank gespeichert. Daher erschien es uns keinen Mehrwert zu bringen, wenn wir die Daten in Modelklassen umwandeln vor der Benutzung. Wenn im einer Erweiterung der Applikation noch mehr Funktionalität hinzukommt, könnte man sich überlegen, Modelklassen einzuführen.
Wieso kein separater Datenbank-Server? <i>Performance</i>	Wir haben während den Live-Tests festgestellt, dass der Betrieb auf der SQLite-Datenbank ausreichend ist, um unsere Performanceziele zu erfüllen. Zudem ist dadurch das Deployment wesentlich einfacher.

<p>Wieso nicht Live-Refreshing auf allen Seiten? <i>Usability</i></p>	<p>Dies war in den nicht-funktionalen Anforderungen nicht enthalten. Wir haben beim Spezifizieren der nicht-funktionalen Anforderungen bewusst darauf verzichtet. Dennoch haben wir an einen oder anderen Ort im Projektverlauf eine live-refreshing-Seite mittels Websockets eingebaut.</p>
<p>Schichtendiagramm <i>Maintainability</i></p>	<p>Durch das Repository-Pattern haben wir eine 3-schichtige Architektur erhalten: Die Anfragen kommen via das Routing zu den Controller. Vom Controller werden sie an den entsprechenden Service weitergeleitet. Und der Service benutzt die korrekten Repositories, um die Daten in der Datenbank zu speichern.</p>
<p>GitLab-CI/CD-Pipeline <i>Maintainability</i></p>	<p>Die erstellte CI/CD-Pipeline in GitLab enthält auf dem develop-Branch die 3 Schritte „lint“, „build“ und „test“. Auf dem staging- und master-Branch kommt jeweils noch der Schritt „deploy“ hinzu. Das Deployment auf dem staging-Branch geht auf die URL staging.pmgame.ch. Das Deployment auf dem master-Branch geht auf die URL pmgame.ch.</p>
<p>NodeJS Crypto-Modul <i>Security</i></p>	<p>Wir haben das in nodeJS eingebaute Crypto-Modul benutzt, um das Passwort in Form eines Salted-Hash in der Datenbank zu speichern.</p>

5.11. Testing

5.11.1. Testing der funktionalen Anforderungen

5.11.1.1. Black-Box-Testing Abbildung 15 zeigt die Resultate des Black-Box-Testing. Im Black-Box-Test haben wir unsere Software genommen und gegen die Use case-Beschreibungen (siehe Abschnitt ??) getestet.

Use Case	Use Case-Name	Priorität	Priorität Product Owner	Resultat	Bemerkungen
UC01	Spiel erstellen	MUSS	1	erfüllt	- Attribut Stakeholder muss nicht eingegeben werden; stattdessen müssen noch folgende zusätzlichen Attribute eingegeben werden: Startrisiko, Wochenarbeitszeit, Mitarbeitertypen - Es kann auch mehrmals der gleiche Spielname verwendet werden
UC02	CRUD Gruppe	MUSS	1	erfüllt	- Gruppe editieren ist nicht implementiert - Es kann auch mehrmals der gleiche Gruppenname im selben Spiel verwendet werden - Spieler löschen nicht implementiert, entsprechend ist Erweiterung 5 hinfällig
UC03	Spiel beitreten	MUSS	2	erfüllt	
UC04	Spiel starten	MUSS	1	erfüllt	
UC05	Runde starten	MUSS	1	teilweise erfüllt	- Der Spielleiter kann kein Risiko auswählen, das eintritt, weil das Feature "Events" noch nicht vollständig implementiert ist und von Hand gespielt werden muss - Der Spielleiter kann noch mehr Attribute eingeben: Maximale Investition in Ausbildung, Wirksamkeit Risikoreduktion, Max. Projektleiterpensum, Stundenansatz für den Projektleiter, Change Request erlaubt (ja/nein)
UC06	Runde spielen	MUSS	1	erfüllt	- Der Spieler erhält auf der Startseite nicht direkt das Rundenformular, sondern die Links zu den Subkategorien des Rundenformulars - Entsprechend werden die Eingaben in den Subkategorien bestätigt - Es gibt keinen spezifischen "Editiermodus": Der Spieler kann direkt editieren und muss nicht zuerst in den "Editiermodus" wechseln
UC07	Change request erstellen	KANN	6	erfüllt	- Der Spieler kann nur Budget und Scope editieren - Der Spieler kann seinen Change Request sogar auch zwischenspeichern
UC08	Change Request erhalten	SOLL	4	erfüllt	
UC09 / UC10	Change Request bestätigen / Change Request ablehnen	KANN	6	erfüllt	- Der Spielleiter kann zudem auch den Change Request zur Überarbeitung zurückgeben an die Gruppe
UC11	Runde beenden	MUSS	1	erfüllt	- Die Erweiterung mit der "Notification" an die Gruppen, welche noch keine Eingaben gemacht haben, ist noch nicht implementiert
UC12	Spielübersicht	KANN	7	nicht erfüllt	- nicht implementiert; Workaround: Der Spieler kann den Spielleiter fragen, denn der Spielleiter hat diese Anzeige
UC13	Dashboard aufrufen	SOLL	3	erfüllt	- wurde leicht abgeändert implementiert - Der Spielleiter kann die Statistikseite einer Gruppe aufrufen und sieht dort die Eingaben aller bisherigen Runden dieser Gruppe - Es gibt aber keine Buttons "Vorgängige Runde anzeigen" oder ähnlich
UC14	Reports einsehen	SOLL	4	erfüllt	- mittels Spielübersicht implementiert: Dort sieht der Spielleiter die Burn-down-Charts für Budget, Scope und Risiko
UC15	Reports exportieren	SOLL	4	nicht erfüllt	
UC16	CRUD Event	SOLL	4	nicht erfüllt	
UC17	Benachrichtigung versenden (E-Mail und Notification)	SOLL	5	nicht erfüllt	
UC18	Benachrichtigung bei ausstehenden Eingaben	KANN	7	nicht erfüllt	
UC19	CRUD Spieler	MUSS	1	teilweise erfüllt	- nur Create und Read implementiert - Update und Delete nicht implementiert
UC20	Spieler eröffnet eine Gruppe	SOLL	3	erfüllt	- keine Warnmeldung, wenn eine Gruppe mit dem gleichen Namen bereits existiert - Nur der Spielleiter kann schlussendlich die Gruppe bestätigen
UC21	Spieler tritt einer Gruppe bei	KANN	7	erfüllt	- Wenn die Gruppe bereits voll ist, ist der "Beitreten"-Button nicht deaktiviert, jedoch erhält der Spieler eine Fehlermeldung
UC22	CRUD Spielleiter	KANN	6	nicht erfüllt	- kann jedoch hard-coded in der Datenbank eingegeben werden
UC23	Spiel editieren oder löschen	SOLL	3	nicht erfüllt	- lediglich die Funktionalität, dass der Spielleiter einen Spieler aus einem noch nicht gestarteten Spiel entfernen, kann ist implementiert
UC24	CRUD Mitarbeitertyp	KANN	8	nicht erfüllt	- kann jedoch hard-coded in der Datenbank eingegeben werden
UC25	CRUD Aufwand	MUSS	2	nicht erfüllt	- kann jedoch hard-coded in der Datenbank eingegeben werden

Getestet am 13. Dezember 2019

Version: commit ced36c0133452c32bc07dc98fcd4bfa78528b7c5

Tester: Marco Keller

Abbildung 15: Black-Box-Testing gegen die Use cases.

5.11.1.2. White-Box-Testing Das White-Box-Testing haben wir vorwiegend mit Integrationstests gemacht. Unter Integration-Tests verstehen wir Funktionsaufrufe aus dem Backend, die bis auf die Datenbank hinunterlaufen. Wir können so zuerst ein Testszenario in die Datenbank laden, dann einen Funktionsaufruf tätigen und anschliessend prüfen, ob (i) die Datenbank gemäss unserer Erwartung geändert wurde und (ii) ob wir den erwarteten Rückgabewert aus der Funktion erhalten haben.

Ein Vorteil von Integrationstests ist, dass die wir so kein Mocking benötigen. Zudem können wir direkt in der Datenbank nachschauen, wo der Fehler lag, wenn etwas nicht geklappt hat. Die Test-Coverage ist in Abbildung 16, Abbildung 17, Abbildung 18 und Abbildung 19 dargestellt.

All files
 37.91% Statements 547/1443 51.32% Branches 194/378 42.27% Functions 323/291 38.09% Lines 545/1431

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	0%	0/26	100%	0/26
src/controller	40.86%	219/536	54.19%	219/535
src/model	0%	0/6	100%	0/6
src/repositories	58.58%	140/239	83.78%	140/239
src/routes	0%	0/63	100%	0/63
src/services	33.04%	188/569	38.97%	186/559
src/utills	0%	0/4	0%	0/3

Abbildung 16: White-Box-Testing: Test-Coverage: Übersicht.

All files src/controller
 40.86% Statements 219/536 54.19% Branches 118/283 35.09% Functions 28/57 40.93% Lines 219/535

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

File	Statements	Branches	Functions	Lines
adminController.ts	17.73%	25/141	3.03%	25/140
gameController.ts	43.4%	138/318	64.38%	138/318
userController.ts	72.73%	56/77	62.5%	56/77

Abbildung 17: White-Box-Testing: Test-Coverage: Controller-Schicht.

All files src/services

33.04% Statements 188/569 38.97% Branches 53/136 34.19% Functions 48/137 33.27% Lines 186/559

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

File	Statements	Branches	Functions	Lines
changeRequestService.ts	14.63%	6/41	0%	6/41
dataAggregationService.ts	9.76%	12/123	0%	12/120
gameService.ts	44.3%	35/79	61.54%	35/79
groupService.ts	75%	15/20	0%	13/18
roundService.ts	27.42%	68/248	12.96%	68/243
userService.ts	89.66%	52/58	93.75%	52/58

Abbildung 18: White-Box-Testing: Test-Coverage: Service-Schicht.

All files src/repositories

58.58% Statements 148/253 83.78% Branches 31/37 56.25% Functions 63/112 58.58% Lines 148/253

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

File	Statements	Branches	Functions	Lines
changeRequestRepository.ts	41.67%	10/24	2/2	10/24
employeeTypeRepository.ts	100%	6/6	2/2	6/6
eventImpactRepository.ts	57.14%	8/14	2/2	8/14
gameRepository.ts	79.41%	27/34	6/6	27/34
groupRepository.ts	70%	7/10	2/2	7/10
managementDecisionRepository.ts	25%	12/48	2/4	12/48
projectManagerTaskRepository.ts	100%	6/6	2/2	6/6
projectManagerTimeAllocationRepository.ts	33.33%	5/15	2/6	5/15
roundRepository.ts	73.91%	34/46	4/4	34/46
staffingRepository.ts	53.33%	8/15	4/4	8/15
userRepository.ts	80.95%	17/21	3/3	17/21

Abbildung 19: White-Box-Testing: Test-Coverage: Repository-Schicht.

5.11.2. Testing der nicht-funktionalen Anforderungen

Dieses Kapitel testet das Erreichen der im Kapitel Nicht-funktionale Anforderungen aufgestellten nicht-funktionalen Anforderungen. Die Testergebnisse sind in Abbildung 20 festgehalten.

Wir haben aus Zeitgründen auf einen automatisierten Load- und Performancetest verzichtet. Zudem haben wir durch die Live-Tests im Unterricht gesehen, dass das Anwendungsszenario 1 (siehe Kapitel Kapitel Nicht-funktionale Anforderungen) unterstützt wird.

Kategorie	Subkategorie	Resultat	Bemerkung
Functional Stability	Functional Completeness	teilweise erfüllt	- User story 37 "Ich als Spielleiter erfasse einen oder mehrere Aufwände" wurde nicht implementiert. Die Aufwände müssen hard-coded in der Datenbank erfasst werden
	Functional Correctness	erfüllt	- getestet mit Unit Tests und Integration Tests, sowie mit Live-Test
	Functional Appropriateness	n/a	
Performance efficiency	Time-Behavior	erfüllt	- mit Live-Test getestet; E-Mail/Notifications verschicken nicht implementiert
	Resource Utilization	n/a	
	Capacity	nicht getestet	
Compatibility	Co-existence	n/a	
	Interoperability	n/a	
Usability	Appropriateness recognizability ("Obvious Applicability")	nicht erfüllt	
	Learnability	teilweise erfüllt	
	Operability ("Intuitiveness")	teilweise erfüllt	- Aufwanderfassung und Eventerfassung nicht implementiert - Übersicht, wie weit die anderen Teams sind nicht implementiert aus Spielersicht - Spielzugeingabe und vorherige Spielzüge nicht auf derselben Seite sichtbar, jedoch in 1 Eingabeaktion erreichbar
	User error protection	erfüllt	
	User interface aesthetics ("Aesthetics", "Look & Feel")	teilweise erfüllt	- Material design: Color-Anforderung nicht vollständig erfüllt: mehr als 1 Primary Color und 1 Accent Color
	Accessibility	n/a	
Reliability	Maturity	n/a	
	Availability	erfüllt	
	Fault Tolerance	n/a	
	Recoverability	nicht erfüllt	- aktuell noch keine Backuplösung implementiert
Security	Confidentiality	erfüllt	
	Integrity	erfüllt	
	Non-repudiation	n/a	
	Accountability	n/a	
	Authenticity	n/a	
Maintainability	Modularity	erfüllt	
	Reusability	n/a	
	Analyzeability	n/a	
	Modifiability	erfüllt	
	Testability	teilweise erfüllt	- nicht alle Use cases mittels Unit-Tests oder Integration-Tests überprüft
Portability	Adaptability	erfüllt	
	Installability	erfüllt	- Readme im Wiki in GitLab
	Replaceability	n/a	

Getestet am 13. Dezember 2019

Version: commit `ced36c0133452c32bc07dc98fcd4bfa78528b7c5`

Tester: Marco Keller

Abbildung 20: Testing der nicht-funktionalen Anforderungen.

5.11.2.1. Live-Tests Wir konnten 3 Mal im PmQm-Unterricht unsere Software testen. Der erste Test diente den Studenten, die Software kennenzulernen. Beim zweiten und dritten Test wurde „scharf“ gespielt.¹⁸ Für uns stellten die Tests eine Gelegenheit dar, die Software unter Live-Bedingungen im Betrieb zu sehen und Verbesserungsmöglichkeiten zu finden.

5.11.2.2. Erkenntnisse Wir haben beim Spezifizieren der nicht-funktionalen Anforderungen in der Elaboration-phase das Niveau bewusst tief gehalten. Zudem haben wir stark darauf geachtet, dass wir die nicht-funktionalen Anforderungen gemäss „SMART“ erfassen. Dennoch haben wir beim Testen der nicht-funktionalen Anforderungen erkannt, dass selbst dieses tiefe Niveau nicht ganz einfach zu erreichen und zu prüfen ist.

Aus Zeitgründen mussten wir auch auf das Testen einiger Anforderungen verzichten.

Wir nehmen für zukünftige Softwareprojekte mit, dass wir besonders vorsichtig sein müssen, welche Qualitätsgarantien wir abgeben. Zudem müssen wir beim Abgeben der Qualitätsgarantien für jede Anforderungen einen Plan haben, wie wir dies testen wollen.

Bezüglich Live-Test halten wir fest, dass dies ein gutes, ergänzendes Testwerkzeug darstellt. Wir haben während den Live-Tests diverse kleinere Bugs und Ungereimtheiten festgestellt: z.B. Buttons zu gross, fehlende Rückmeldung, unpraktikable Passworteingabe. Dies sind Dinge, auf die man beim Programmieren nicht direkt achtet.

¹⁸Bezüglich dem misslungen zweiten Test in der Übungsgruppe 1 siehe Abschnitt 6.6.1.

5.12. Installationsanleitung

5.12.1. Voraussetzungen

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

- min. 2 GB freier Arbeitsspeicher
- min 10 GB freier Speicher
- min Docker Version 19.03

5.12.2. Deployment auf HSR Server

Wir haben auf GitLab eine CI/CD Pipeline eingerichtet, welche automatisches Deployment auf der „Staging“- und „Production“-Umgebung erlaubt. Dazu muss lediglich neuer Inhalt auf den „Staging“- oder „Master“-Branch gemerget werden.

Der Ablauf ist wie folgt:

1. Die Pipeline läuft nach dem Mergen durch den Building-Prozess und erstellt ein neues Image für den Dienst „Docker Hub“. Dieses wird auch sogleich gepusht.
2. Sobald die Pipeline durchgelaufen ist, wartet die Pipeline. Nun kann unter dem Abschnitt „Deployment“ das Deployment auf den Server manuell ausgelöst werden. Die Pipeline ist in Abbildung 21 ersichtlich.



Abbildung 21: Die vier Stages der CI/CD Pipeline sind ersichtlich. Als letzten Schritt kann das Deployment manuell ausgelöst werden.

3. Auf dem HSR Server wird nun das neue Image geholt und die Docker-Instanz neu gestartet.

Hinweis: Beim Deployment bleibt die alte Datenbank erhalten. Ändert sich das Datenbankschema, so muss manuell die Datenbank entfernt werden. Dazu werden in der laufenden Docker-Instanz folgende Befehle ausgeführt:

Production-Umgebung

```
npm run prod_db_down  
npm run prod_db_up  
npm run prod_initate_game
```

Staging-Umgebung

```
npm run dev_db_down
npm run dev_db_up
npm run dev_initate_game
```

Dadurch wird die Datenbank automatisch geleert und mit dem neuen Schema wieder aufgebaut.

5.12.3. Installation auf einem neuen Host

Falls die Software auf einem anderen Host als dem HSR Server betrieben werden soll, dann muss diese manuell installiert werden.

Die Software wird als Docker-Container hochgefahren. Daher ist die Installation schnell und unkompliziert. Ein entsprechende Docker Image steht auf dem Dienst Docker Hub zur Verfügung.

1. Auf dem Host müssen die Docker Hub Logindaten angegeben werden. Dazu wird folgender Befehl ausgeführt: ‘docker login‘
2. Nun wird auf dem Host im gewünschten Verzeichnis die Datei ‘docker-compose.yml‘ mit folgendem Inhalt angelegt:

```
version: '3'
services:
  pm_prod:
    image: hsrpm/pm:prod
    environment:
      ENVIRONMENT: production
    ports:
      - "80:80"
    volumes:
      - ./databases:/app/data
```

3. Nun kann die Applikation mit dem Befehl ‘docker-compose up -d‘ gestartet werden.
4. Die Applikation ist nun über den Port 80 erreichbar.

5.12.4. Lokales Ausführen der Applikation

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

1. Der Quellcode wird vom Git Repository heruntergeladen.
2. Im Verzeichnis, in dem der Quellcode liegt, werden nun folgende Befehle im Terminal ausgeführt:

```
npm run dev_db_up
npm run dev_initiate_game
npm run start
```

3. Die Applikation startet und ist auf dem Localhost nun über den Port 3000 erreichbar.

5.12.5. Datensicherung Datenbank

Um die Datenbank vom HSR Server zu sichern, muss wie folgt vorgegangen werden:

1. Mit einem SFTP-fähigen FTP Client muss auf den Server 152.96.56.92 verbunden werden. Dazu wird der Benutzer 'pm_user' verwendet. Die Zugangsdaten sind im Wiki dokumentiert.
2. Nun kann in das Verzeichnis 'databases' navigiert werden. Dort liegen zwei Datenbankdateien. Eine für die „produktive“ Umgebung und eine für die „Staging“ Umgebung.

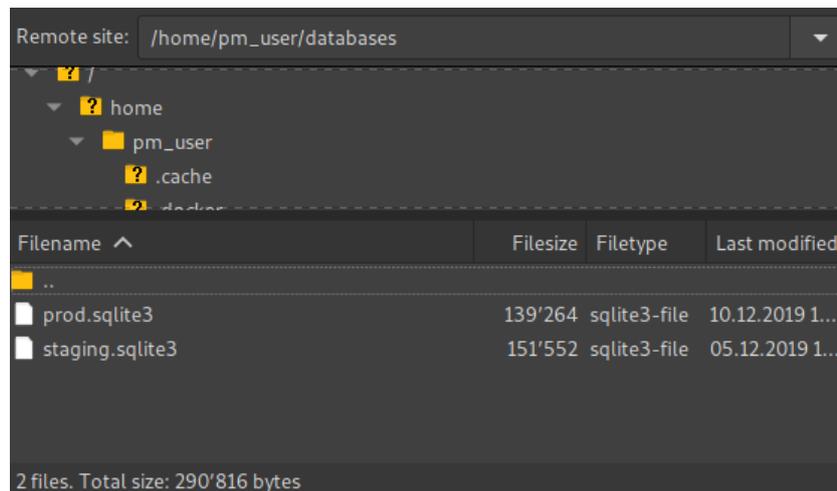


Abbildung 22: Die Dateien der Datenbanken liegen im Verzeichnis Databases.

3. Diese Dateien können nun mit dem FTP Client heruntergeladen werden.
4. Auf diesem Weg können auch Sicherungen wieder eingespielt werden. Dazu reicht es, die Dateien auf dem Server zu ersetzen.

5.13. Benutzerhandbuch (User Manual)

5.13.1. Anlegen eines neuen Spiels (Admin)

1. Über den Menüpunkt „Spiele“ die Seite „Spiele“ aufrufen.

2. Den Button „Neues Spiel erfassen“ klicken, um ein neues Spiel zu erfassen.
3. Nun können die Grunddaten eingetragen werden. Wichtige Grössen sind das Startbudget, Start-Scope und das Startrisiko. Diese beeinflussen das Spiel massgeblich.
4. Ebenfalls müssen auf dieser Seite zwingend die Mitarbeitertypen und Projektleitertaufgaben ausgewählt werden, welche den Spielern zur Verfügung stehen.

Abbildung 23: In dieser Übersicht werden alle wichtigen Daten eingetragen.

5. Mit dem Button „Spiel anlegen“ wird nun das Spiel erstellt. Die Spieler können diesem Spiel nun beitreten und Gruppen erfassen.

5.13.2. 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.

1. Über den Menüpunkt „Spiele“ die Seite „Spiele“ aufrufen.
2. Das Spiel über den Button „Öffnen“ im Abschnitt „Ungestartete Spiele“ öffnen.
3. Prüfen, ob alle Spieler zugeweiht sind. An diesem Punkt kann die Zuteilung durch den Spielleiter noch angepasst werden. In der Abbildung 24 ist die Maske für die Zuteilung ersichtlich.
 - Ein Spieler, welcher sich noch nicht in einer Gruppe befindet, kann einer Gruppe zugeweiht werden, indem der Spieler selektiert wird (Spielername anklicken) und aus dem Dropdownfeld die gewünschte Gruppe ausgewählt wird. Mit dem Button „Spieler Gruppe zuweisen“ werden die ausgewählten Spieler der Gruppe zugeweiht.

- Bestehende Spieler können aus einer Gruppe entfernt werden. Dazu muss beim Spieler, welcher einer Gruppe zugeteilt ist, auf das x geklickt werden. Er erscheint nun wieder im Pool zum Zuteilen einer Gruppe.
- Ein Spieler, welcher sich im Pool mit den nicht zugeteilten Spielern befindet, kann gänzlich aus dem Spiel entfernt werden, indem auf das x neben dem Spielernamen geklickt wird. Dieses erscheint nachdem sich die Maus für eine Sekunde über dem Feld des betreffenden Spielers befunden hat.

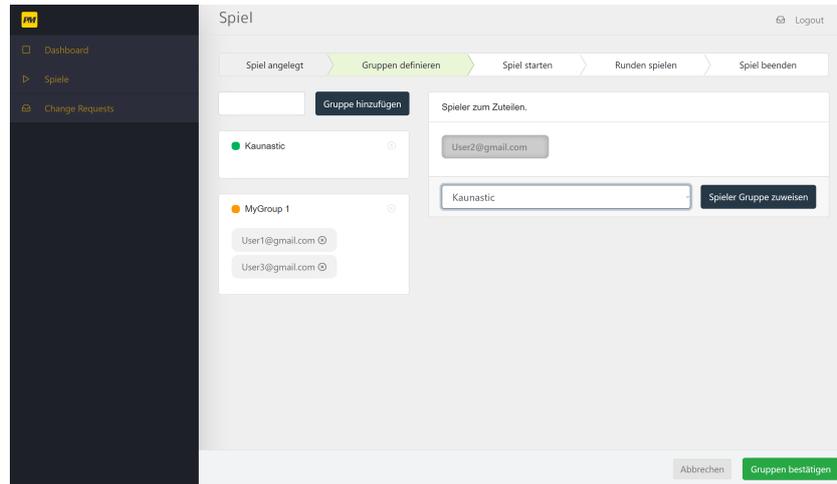


Abbildung 24: Die Benutzer werden in dieser Übersicht einer Gruppe zugeteilt oder auch entfernt.

4. Mit Gruppen bestätigen wird nun die Gruppenzuteilung bestätigt.
5. Nun werden die Parameter für die erste Runde gesetzt.
6. Mit dem Klick auf „Bestätigen“ wird die erste Runde gestartet und die Spieler können Ihre Eingabe machen.

5.13.3. Spielereingabe (Spieler)

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

Staffing Die Spieler können aus den zur Verfügung stehenden Mitarbeitertypen ihr Team zusammenstellen. Es sind max. so viele Stellenprozent möglich wie der Spielleiter für diese Runde definiert hat.

Projektleiter Planung Die Spieler definieren, wie viel Prozent der Zeit des Projektleiters in die verschiedenen Projektleitertätigkeiten investiert wird. Der Spielleiter kann diese Eingaben dann beispielsweise bei einem Change Request einsehen.

Investitionen Die Spieler investieren in Risikoreduktion oder Ausbildung. Dies kann beispielsweise die Effektivität der Mitarbeiter steigern.

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.

5.13.4. 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. Über den Menüpunkt „Change Requests“ die Seite „Change Requests“ aufrufen.
2. Den gewünschten Change Request öffnen.
3. Es wird nun der Change Request dem Spielleiter angezeigt. Auf der rechten Seite kann der Spielleiter einsehen, wie viel die Spieler in welche Projektleitertätigkeiten 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 greift für die Spieler per sofort.

5.13.5. 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 Spiel über den Button „Öffnen“ im Abschnitt „Gestartete Spiele“ 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.

5.14. Known Bugs

In diesem Abschnitt erfassen wir alle Bugs. „Nice-to-have-Features“ zählen wir nicht als Bugs. Diese sind in einer separaten Liste im Kapitel Erweiterungsmöglichkeiten aufgeführt.

- Abgelehnte Change Requests von einer Gruppe werden als Link auch allen anderen Gruppen angezeigt. Der Zugriff darauf ist jedoch für die anderen Gruppen nicht möglich.
- floating-number-Issue beim Summieren der Stellenprozente für den Projektmanager. Wenn zum Beispiel das maximal mögliche Pensum 50% ist und man mit dem Slider über die diversen Tasks das maximale Pensum vergeben möchte, kann es vorkommen, dass beim Summieren 50.000000324% herauskommt, was dann über 50% sind und somit die Aktion nicht zugelassen wird. Workaround: einfach nur 49% dann auswählen. Für einen sauberen Bugfix war bisher noch keine Zeit.
- Admin: Burn-down charts: beginnen aktuell mit: Start, Runde 0, Runde 1, ...; sollte aber Start, Runde 1(!), Runde 2, ... sein
- Foreign key in der user-Table auf roles-Table ist verloren gegangen bei einem Merge.
- Gruppenfarben: aktuell sind nur 15 Farben eingetragen. Wenn mehr als 15 Teams spielen, haben die Teams über 15 keine Farbe mehr zugeordnet
- Die Function Points pro Mitarbeitertyp bei der Auswahl des Staffings werden noch nicht mit dem Trainingslevel multipliziert
- Auf der Startseite (wenn man noch nicht eingeloggt ist) verdeckt das Bild die Controls für das Login / Signup auf kleinen Spielschirmgrößen
- Admin: Ansicht „offene Change Requests“: Der Footer überdeckt die Aktionsbuttons (je nach Auflösung, Browser). Aktueller work-around: Im Browser den Zoom verändern.
- Bei der Erfassung eines neuen Spiels kann zwischen „Am schnellsten Fertig“ und „Am meisten Budget übrig“ gewählt werden. Jedoch ist diese Funktionalität noch nicht implementiert im Backend.
- Admin möchte einen Spieler einer Gruppe zuweisen, hat aber keinen Spieler ausgewählt. Dann wird eine Datenbank-interne Exception geworfen.
- Change Request abschicken: Falls der Spieler den Change Request noch nicht zwischengespeichert hat, muss der Button zum Abschicken 2mal gedrückt werden.

5.15. 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.¹⁹

¹⁹Diesen Tipp haben wir in der Vorlesung „SE2“ erhalten

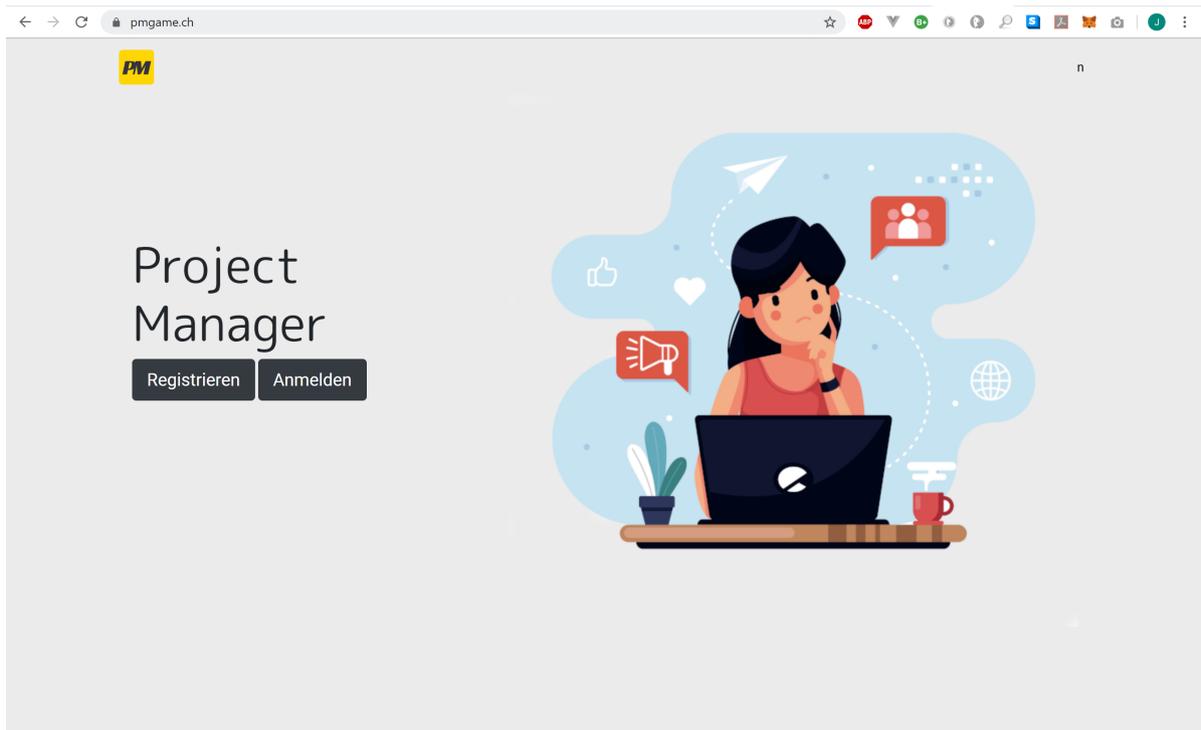


Abbildung 25: Landing page.

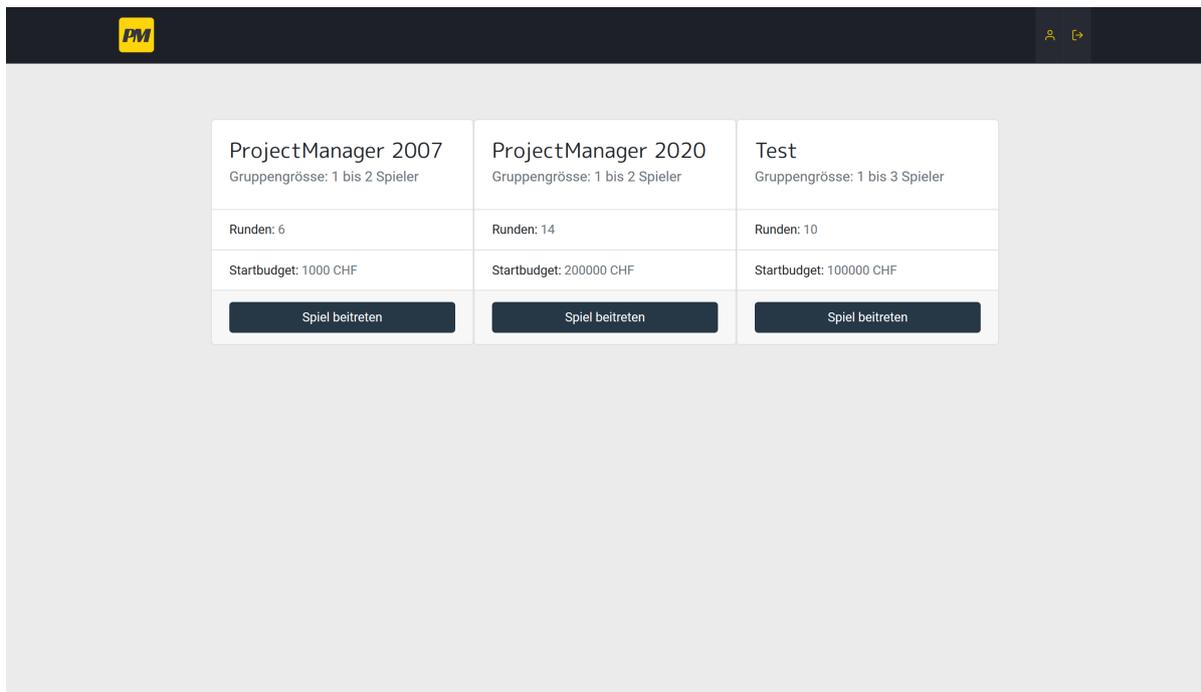


Abbildung 26: Spielersicht: Einem Spiel beitreten.

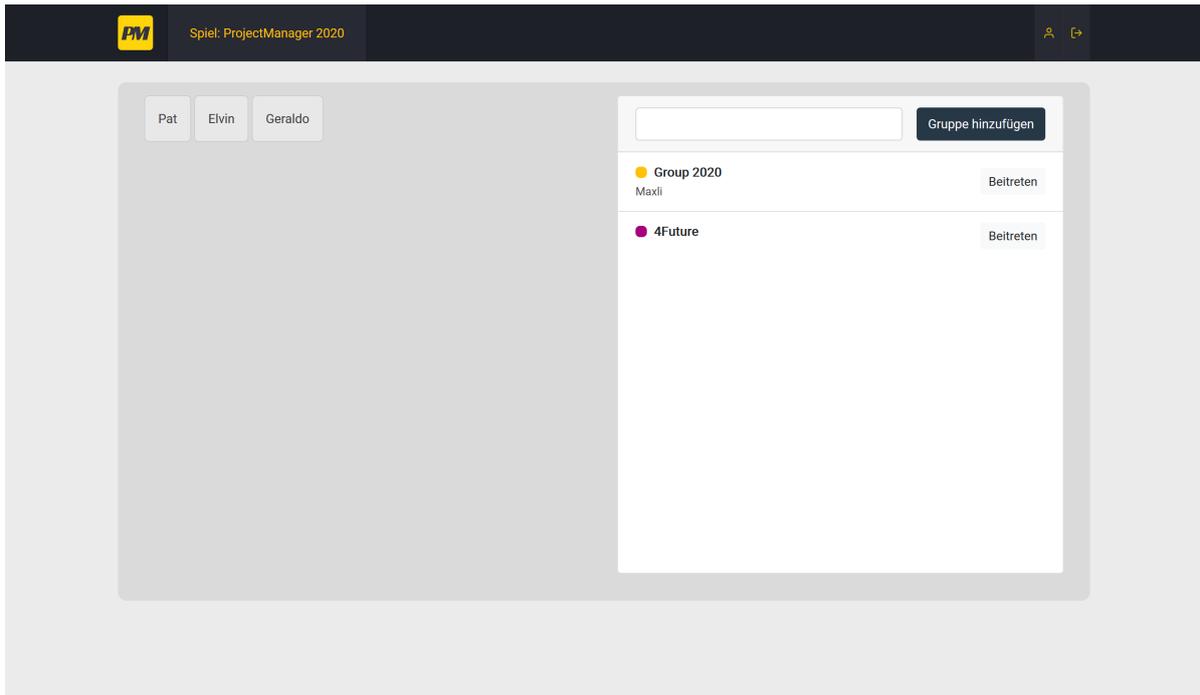


Abbildung 27: Spielersicht: Einer Gruppe beitreten.

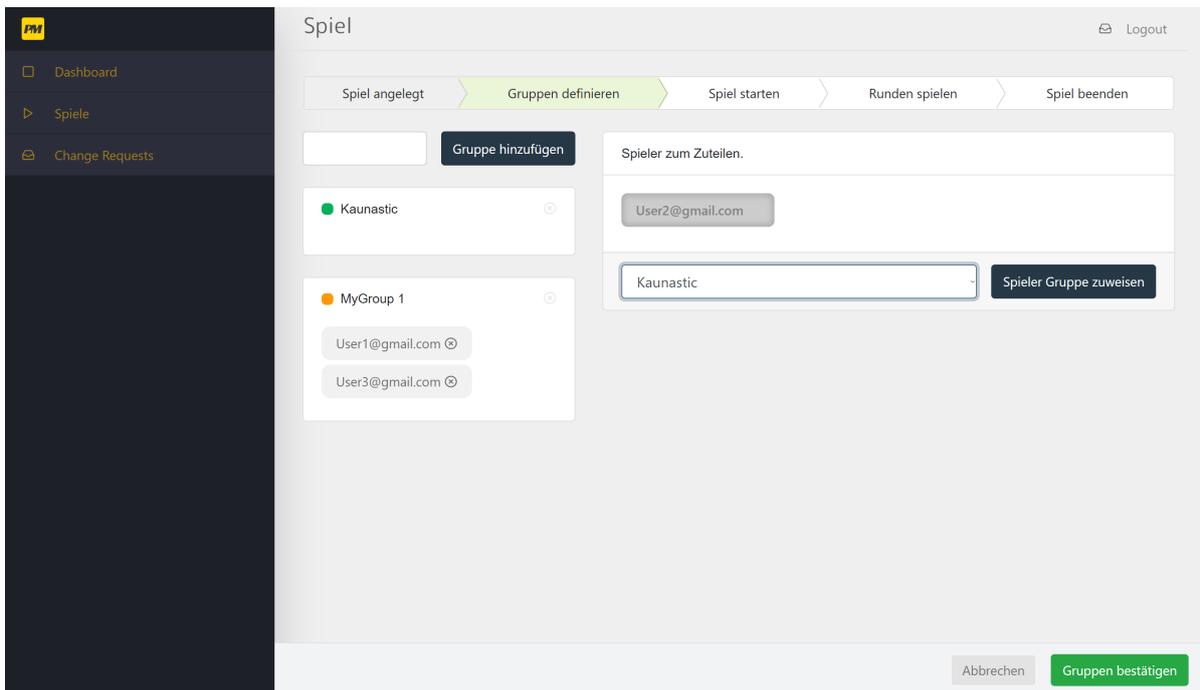


Abbildung 28: Spielleitersicht: Gruppenadministration.

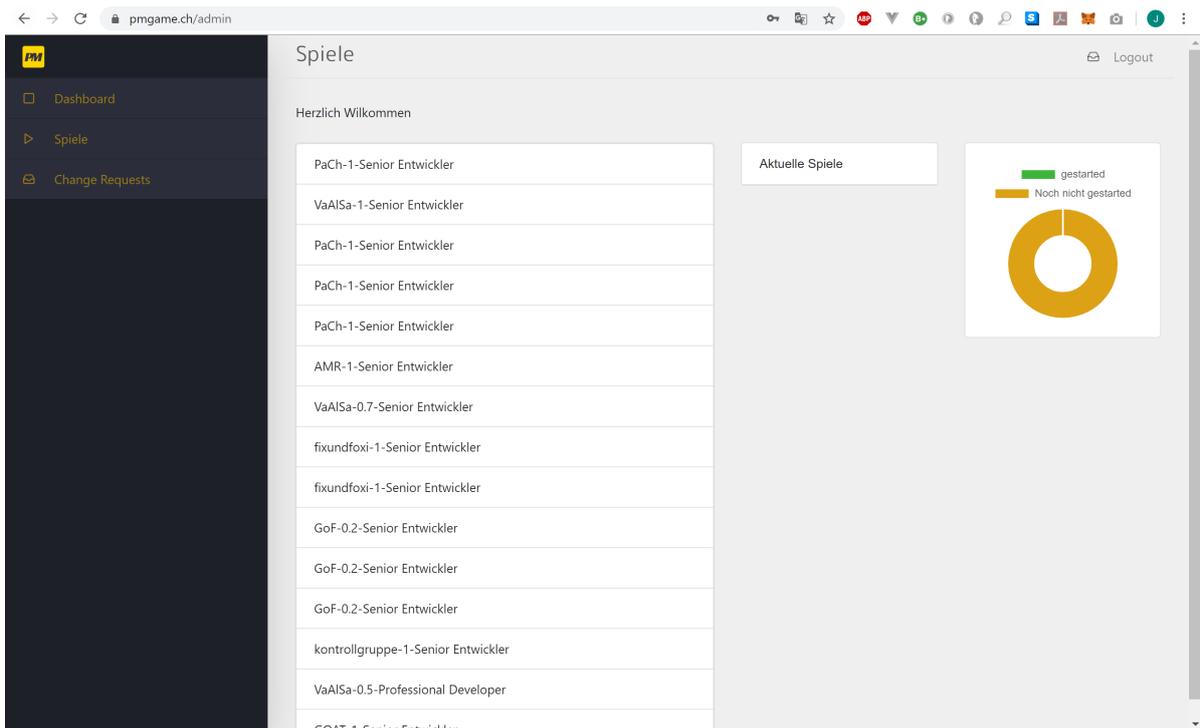


Abbildung 29: Spielleitersicht: Landing page.

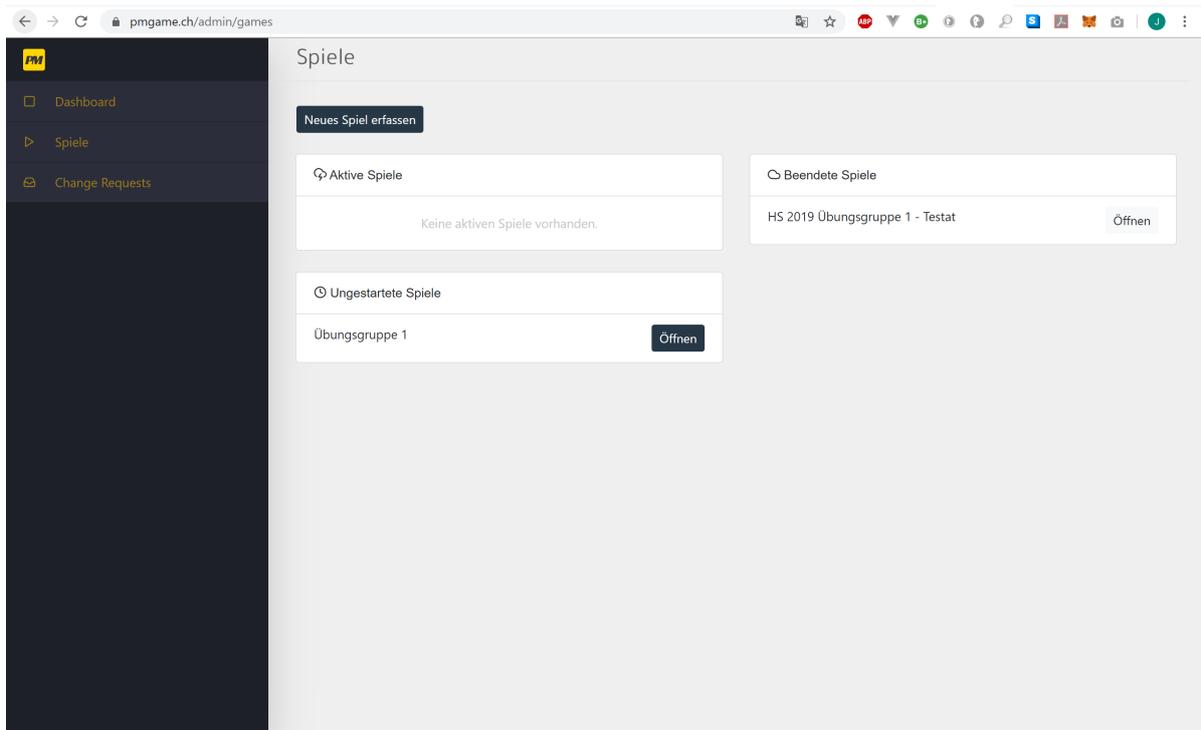


Abbildung 30: Spielleitersicht: Übersicht über die laufenden und beendeten Spiele.

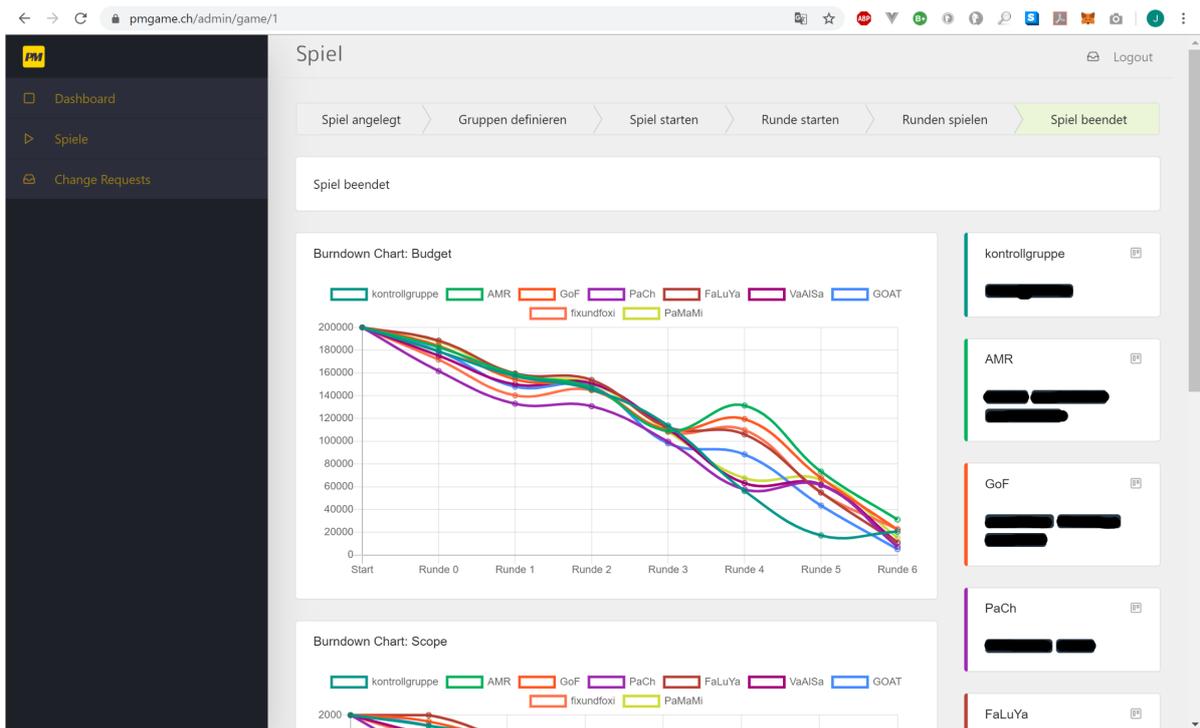


Abbildung 31: Spielleitersicht: Übersicht eines spezifisches Spiels.

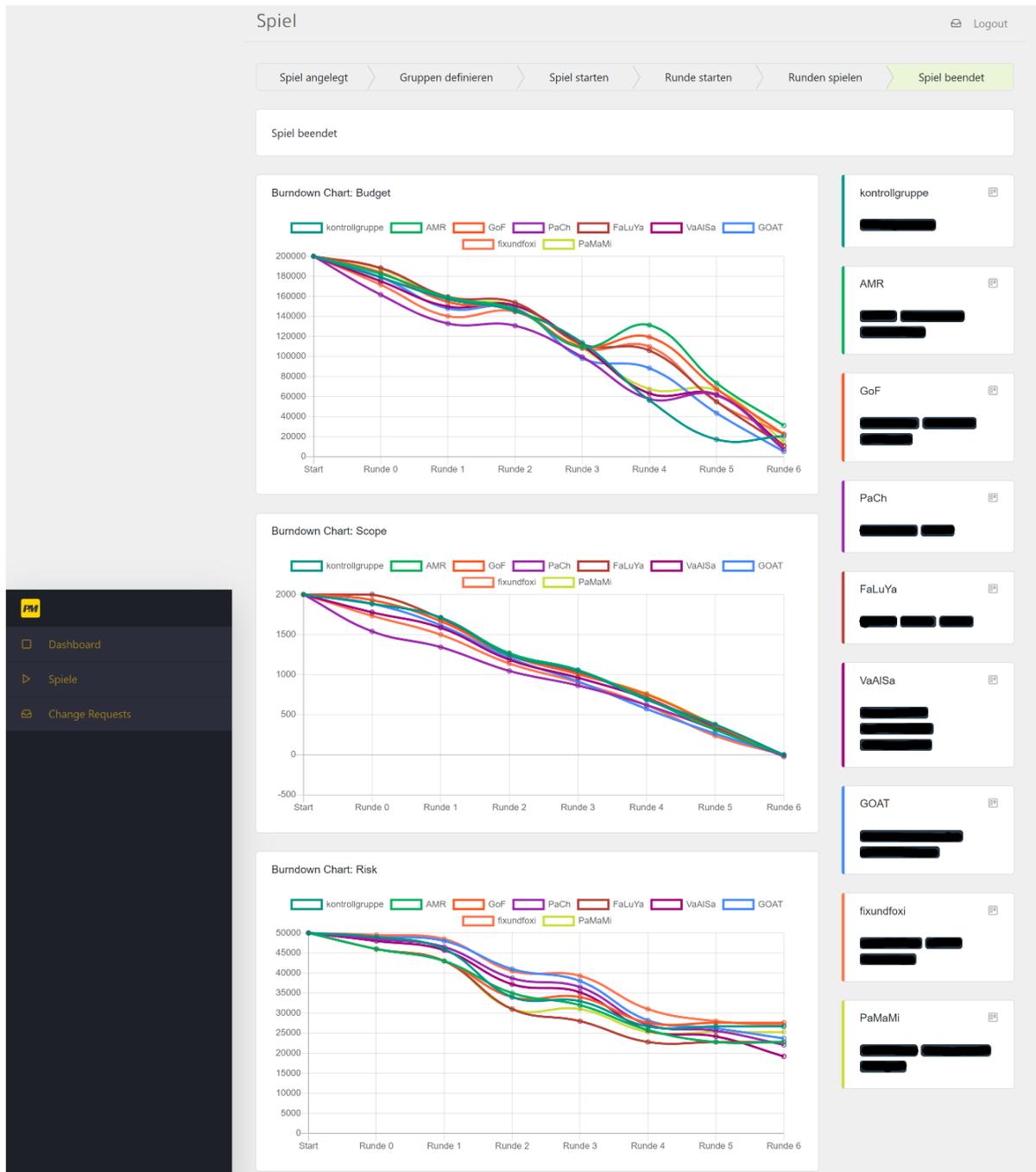


Abbildung 32: Spielleitersicht: Übersicht eines spezifisches Spiels (fullpage screenshot).

PM

Dashboard
Spiele
Change Requests

Neues Spiel erfassen

Grunddaten

Name
HS 2019 Übungsgruppe 1

Startbudget: 200000 Start-Scope (FP): 2000 Startrisiko: 90000

Max. Ausbildung: 10000

Min. Gruppengröße: 2 Max. Gruppengröße: 5

Anzahl Runden: 10

Wochenarbeitszeit in Stunden: 42

Ziel des Spiels

Am schnellsten Fertig
 Am meisten Budget übrig

Projektleiter Aufgaben zuweisen

Verfügbare Mitarbeiter zuweisen

Abbrechen **Spiel anlegen**

Abbildung 33: Spielleitersicht: Erstellen eines neuen Spiels.

PM

Dashboard
Spiele
Change Requests

Runde starten

Logout

Spiel angelegt > Gruppen definieren > **Spiel starten** > Runden spielen > Spiel beenden

Grunddaten

Max. Investition Ausbildung: CHF 5000

Wirksamkeit Risikoreduktion: 50% Wirksamkeit Ausbildung: 100%

Max. Projektleiterpensum: 50% Max. Mitarbeiterpensum: 400 %

Stundenlohn Projektmanager: 170 Max. Risikoreduktion: 6000

Diese Runde erlaubt einen Change Request:

Abbrechen **Bestätigen**

Abbildung 34: Spielleitersicht: Neue Runde starten.

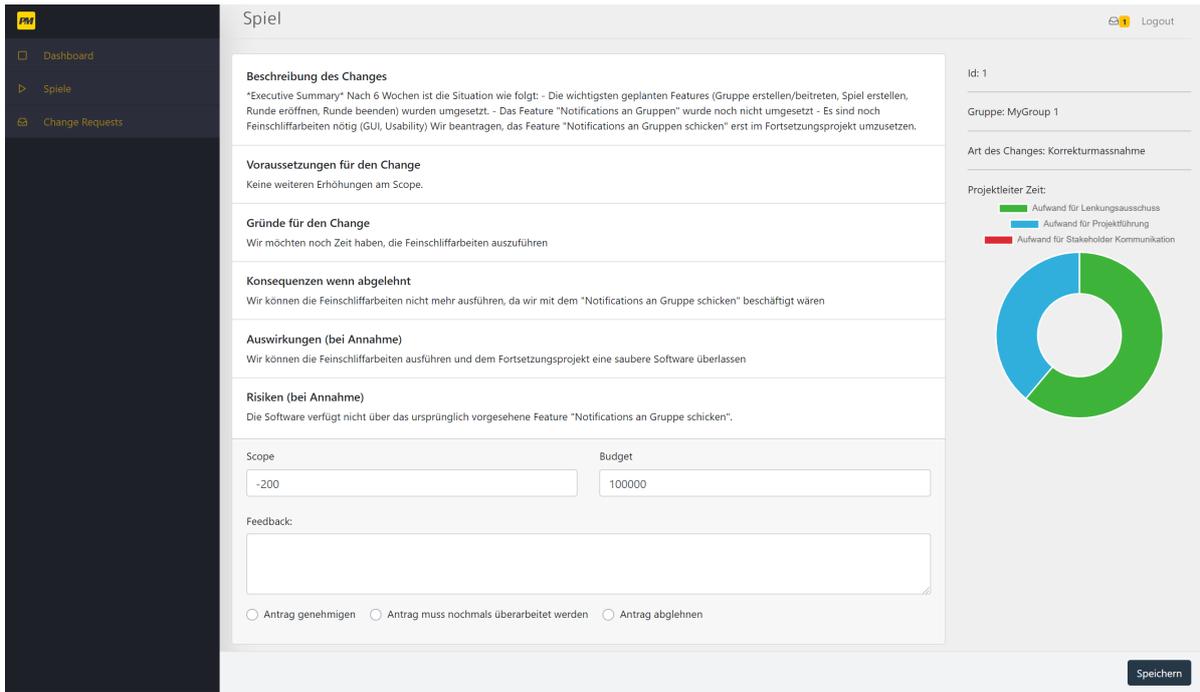


Abbildung 35: Spielleitersicht: Erhalten eines Change Requests.

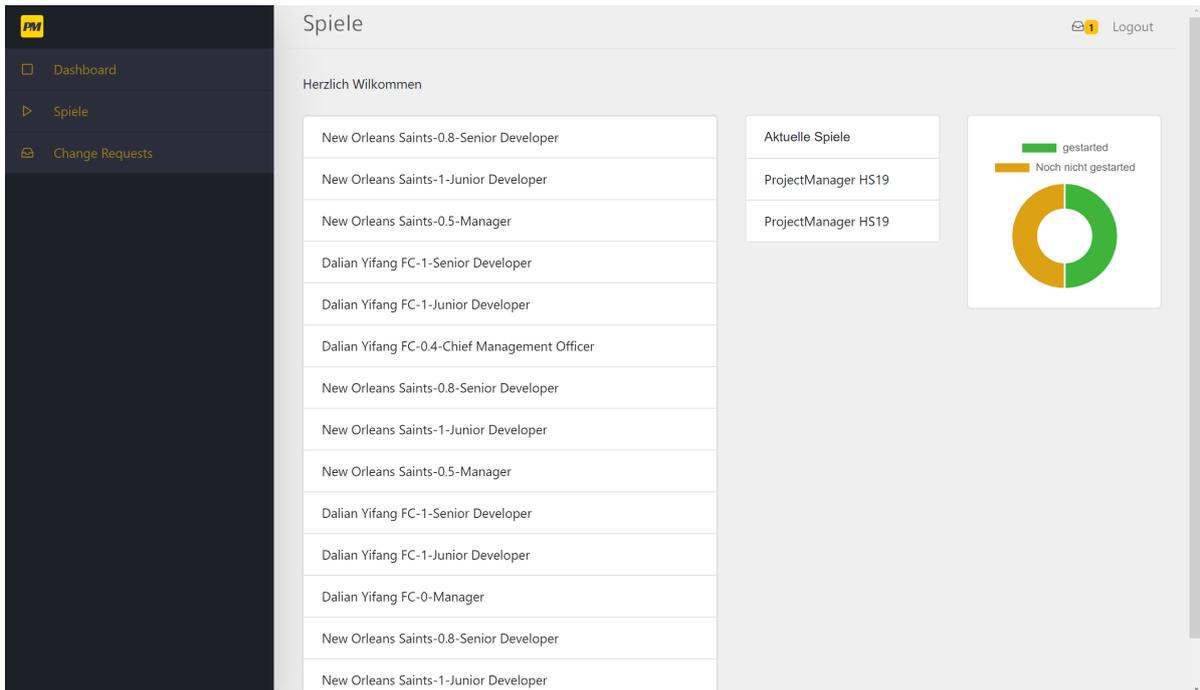


Abbildung 36: Spielleitersicht: Notification oben rechts bei Erhalt eines neuen Change Requests.

pmgame.ch/admin/changeRequest/18

Spiel Logout

Beschreibung des Changes
 Executive Summary Nach 6 Wochen ist die Situation wie folgt: - Die wichtigsten geplanten Features (Gruppe erstellen/beitreten, Spiel erstellen, Runde eröffnen, Runde beenden) wurden umgesetzt. - Das Feature "Notifications an Gruppen" wurde noch nicht umgesetzt - Es sind noch Feinschliffarbeiten nötig (GUI, Usability) Wir beantragen, das Feature "Notifications an Gruppen schicken" erst im Fortsetzungsprojekt umzusetzen.
 Scopeumfang:

Voraussetzungen für den Change
 Keine weiteren Erhöhungen am Scope.

Gründe für den Change
 Wir möchten noch Zeit haben, die Feinschliffarbeiten auszuführen

Konsequenzen wenn abgelehnt
 Wir können die Feinschliffarbeiten nicht mehr ausführen, da wir mit dem "Notifications an Gruppe schicken" beschäftigt wären

Auswirkungen (bei Annahme)
 Wir können die Feinschliffarbeiten ausführen und dem Fortsetzungsprojekt eine saubere Software überlassen

Risiken (bei Annahme)
 Die Software verfügt nicht über das ursprünglich vorgesehene Feature "Notifications an Gruppe schicken".

Dieser Change Request wurde angenommen.

Id: 18
 Gruppe: kontrollgruppe
 Art des Changes: Korrekturmaßnahme

Projektleiter Zeit:
 Aufwand für Lenkungsausschuss
 Aufwand für Projektführung
 Aufwand für Stakeholder Kommunikati
 Aufwand für Qualitätsmanagement
 Aufwand für Motivation, Coaching, Team-

Abbildung 37: Spielleitersicht: Beispiel eines akzeptierten Change Requests.

← → ↻ pmgame.ch/game/statistics

kontrollgruppe Budget:

Runde 7 abgeschlossen

Mitarbeitertyp	Pensum	Arbeitsfähigkeit	Stundenlohn	Kosten	Erliefte Arbeit
Senior Entwickler	100%	1.22 Fp/h	150 CHF	6000 CHF	48.8 Fp
Senior Entwickler	100%	1.22 Fp/h	150 CHF	6000 CHF	48.8 Fp
Senior Entwickler	100%	1.22 Fp/h	150 CHF	6000 CHF	48.8 Fp
Professional Developer	100%	0.98 Fp/h	115 CHF	4600 CHF	39.0 Fp
Senior Entwickler	100%	1.22 Fp/h	150 CHF	6000 CHF	48.8 Fp
Senior Entwickler	100%	1.22 Fp/h	150 CHF	6000 CHF	48.8 Fp
Senior Entwickler	100%	1.22 Fp/h	150 CHF	6000 CHF	48.8 Fp
Senior Entwickler	100%	1.22 Fp/h	150 CHF	6000 CHF	48.8 Fp

Insgesamt wurden in dieser Runde 380.7 Function Points umgesetzt. Dies hat Kosten im Umfang von 46600 CHF verursacht.
 Durch die Mitarbeiterausbildung konnte in der vergangenen Runde ein Trainingslevel von 1.22 erreicht werden.

Aufwand	Pensum
Aufwand für Lenkungsausschuss	0%
Aufwand für Projektführung	0%
Aufwand für Stakeholder Kommunikation	0%
Aufwand für Qualitätsmanagement	0%
Aufwand für Motivation, Coaching, Team-Spirit	0%

Der Projektleiter hat für diese Runde 0 CHF Lohnkosten verursacht.
 Ebenfalls wurden in dieser Runde 0 CHF für die Risikominimierung und weitere 0 CHF für die Ausbildung der Mitarbeiter investiert.
 Insgesamt hat diese Runde 46600 CHF Kosten verursacht.

Der Change Request mit der ID #18 wurde akzeptiert
 ok
 Auswirkungen: 50000 CHF, 0 Fp

Runde 6 abgeschlossen

Mitarbeitertyp	Pensum	Arbeitsfähigkeit	Stundenlohn	Kosten	Erliefte Arbeit
Professional Developer	100%	0.96 Fp/h	115 CHF	4600 CHF	38.5 Fp
Professional Developer	100%	0.96 Fp/h	115 CHF	4600 CHF	38.5 Fp
Professional Developer	100%	0.96 Fp/h	115 CHF	4600 CHF	38.5 Fp

Abbildung 38: Gruppensicht: Statistikseite mit einem Change Request.

← → ↻ pmgame.ch/game/statistics

Durch die Mitarbeiterausbildung konnte in der vergangenen Runde ein Trainingslevel von 1.12 erreicht werden.

Aufwand	Pensum
Aufwand für Lenkungsausschuss	1%
Aufwand für Projektführung	1%
Aufwand für Stakeholder Kommunikation	1%
Aufwand für Qualitätsmanagement	1%
Aufwand für Motivation, Coaching, Team-Spirit	1%

Der Projektleiter hat für diese Runde 340 CHF Lohnkosten verursacht.
 Ebenfalls wurden in dieser Runde 0 CHF für die Risikominimierung und weitere 3000 CHF für die Ausbildung der Mitarbeiter investiert.
 Insgesamt hat diese Runde 51340 CHF Kosten verursacht.

Es wurden Anforderungen von der falschen Person erhoben
 Auswirkungen: -4600 CHF, 0 Fp und -4600 Risiko

Streit im Team
 Auswirkungen: -1500 CHF, 0 Fp und -1700 Risiko

Runde 4 abgeschlossen

Mitarbertyp	Pensum	Arbeitsfähigkeit	Stundenlohn	Kosten	Erfledigte Arbeit
Professional Developer	100%	0.84 Fp/h	115 CHF	4600 CHF	33.6 Fp
Professional Developer	100%	0.84 Fp/h	115 CHF	4600 CHF	33.6 Fp
Professional Developer	100%	0.84 Fp/h	115 CHF	4600 CHF	33.6 Fp
Professional Developer	100%	0.84 Fp/h	115 CHF	4600 CHF	33.6 Fp
Student	100%	0.53 Fp/h	65 CHF	2600 CHF	21.0 Fp
Student	100%	0.53 Fp/h	65 CHF	2600 CHF	21.0 Fp
Student	100%	0.53 Fp/h	65 CHF	2600 CHF	21.0 Fp

Insgesamt wurden in dieser Runde 197.4 Function Points umgesetzt. Dies hat Kosten im Umfang von 26200 CHF verursacht.
 Durch die Mitarbeiterausbildung konnte in der vergangenen Runde ein Trainingslevel von 1.05 erreicht werden.

Aufwand	Pensum
Aufwand für Lenkungsausschuss	7%
Aufwand für Projektführung	2%
Aufwand für Stakeholder Kommunikation	6%
Aufwand für Qualitätsmanagement	4%
Aufwand für Motivation, Coaching, Team-Spirit	3%

Abbildung 39: Gruppensicht: Statistikseite mit Events.

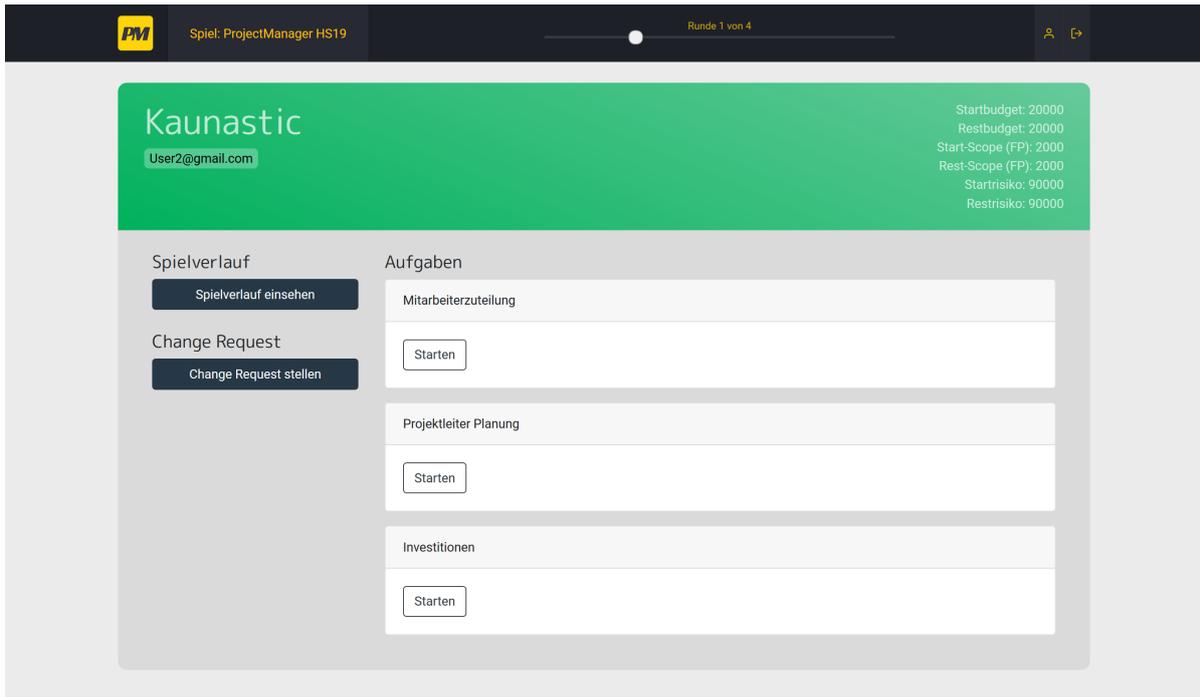


Abbildung 40: Gruppensicht: Startseite in einem laufenden Spiel. Von dieser Seite können die Management-Entscheidungen erreicht werden.

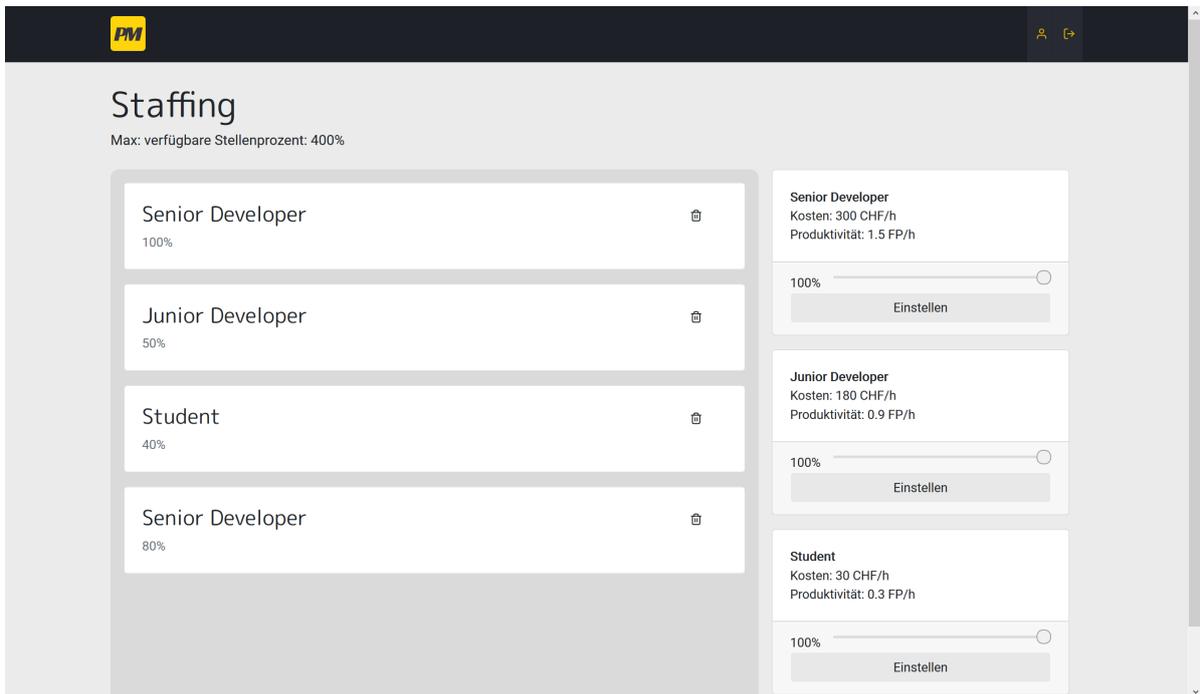


Abbildung 41: Gruppensicht: Eingabe des Staffings für die aktuelle Runde.

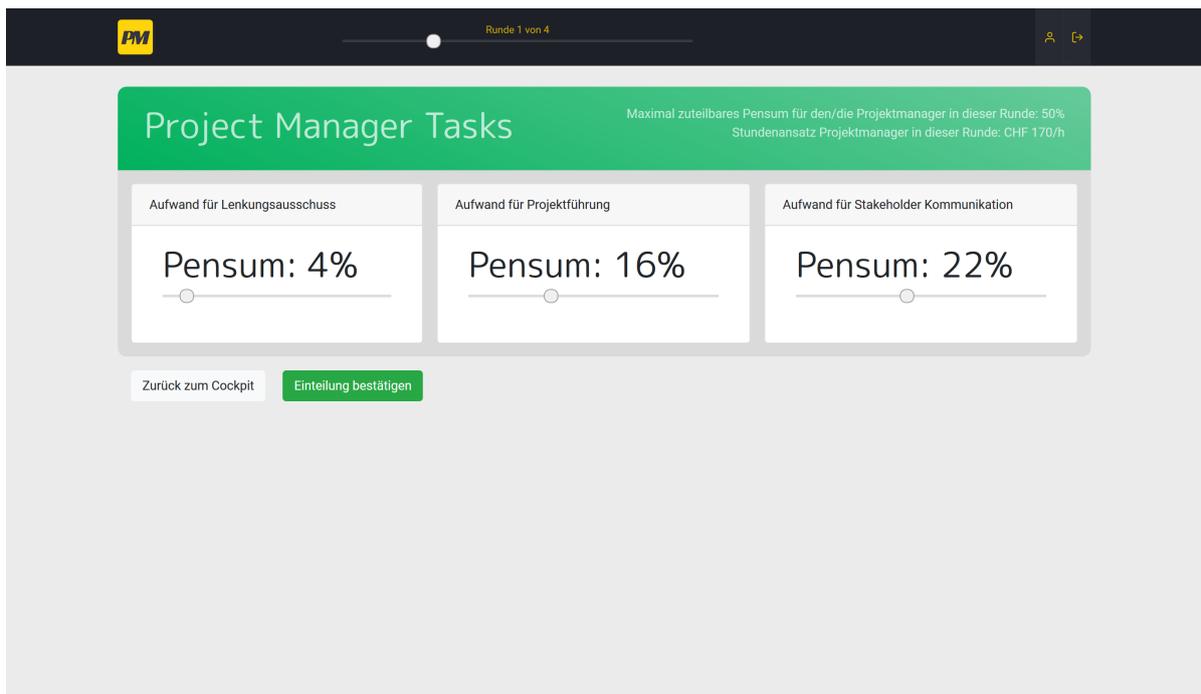


Abbildung 42: Gruppensicht: Eingabe Projektleiterzuteilung auf die verschiedenen Tasks.

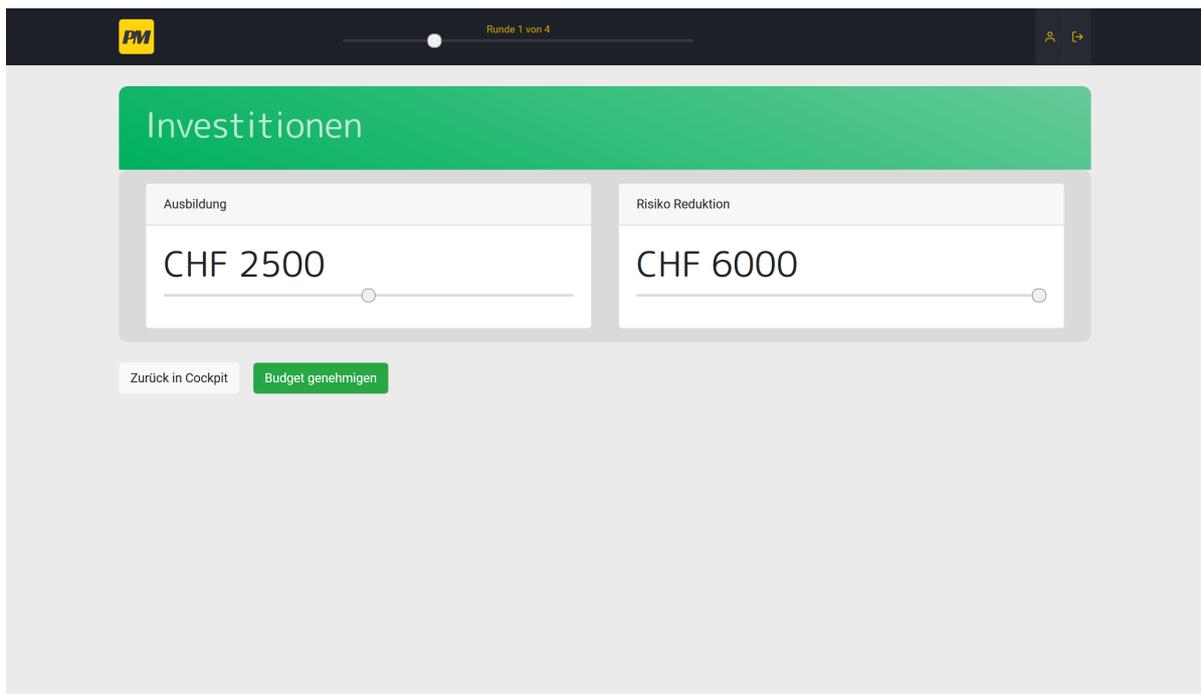


Abbildung 43: Gruppensicht: Eingabe der Investitionen für die Ausbildung und die Risikoreduktion.

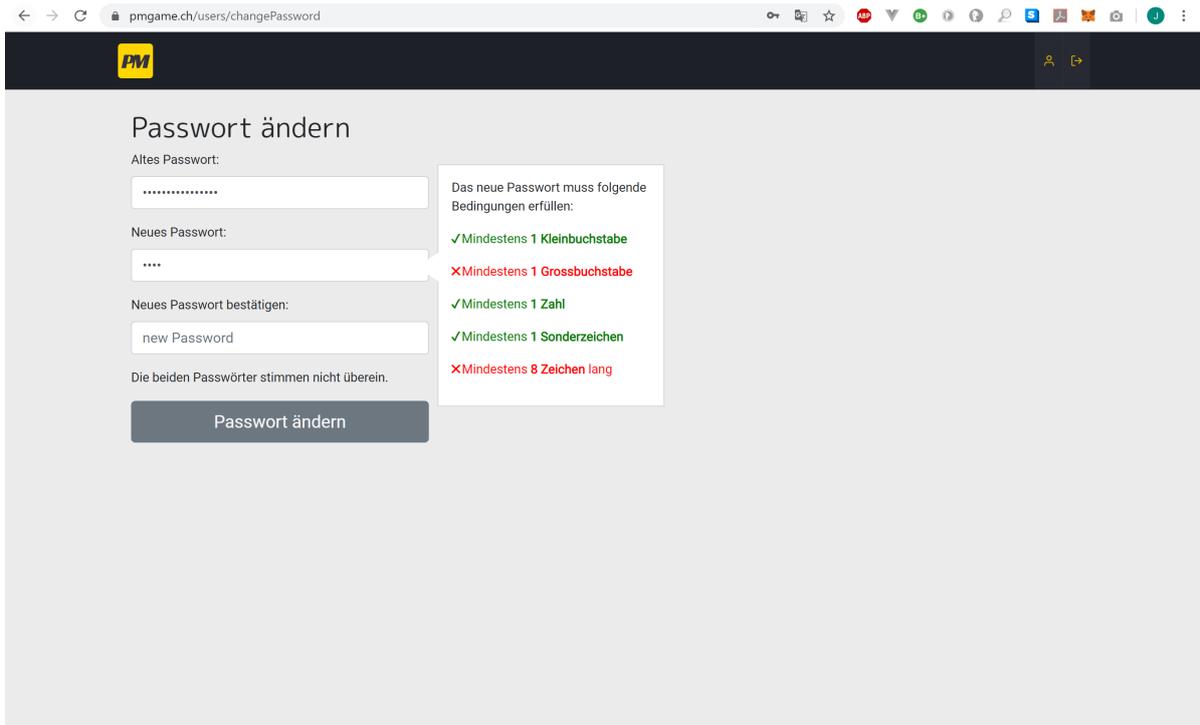


Abbildung 44: Spieler: Passwortwechsel mit interaktivem Feedback.

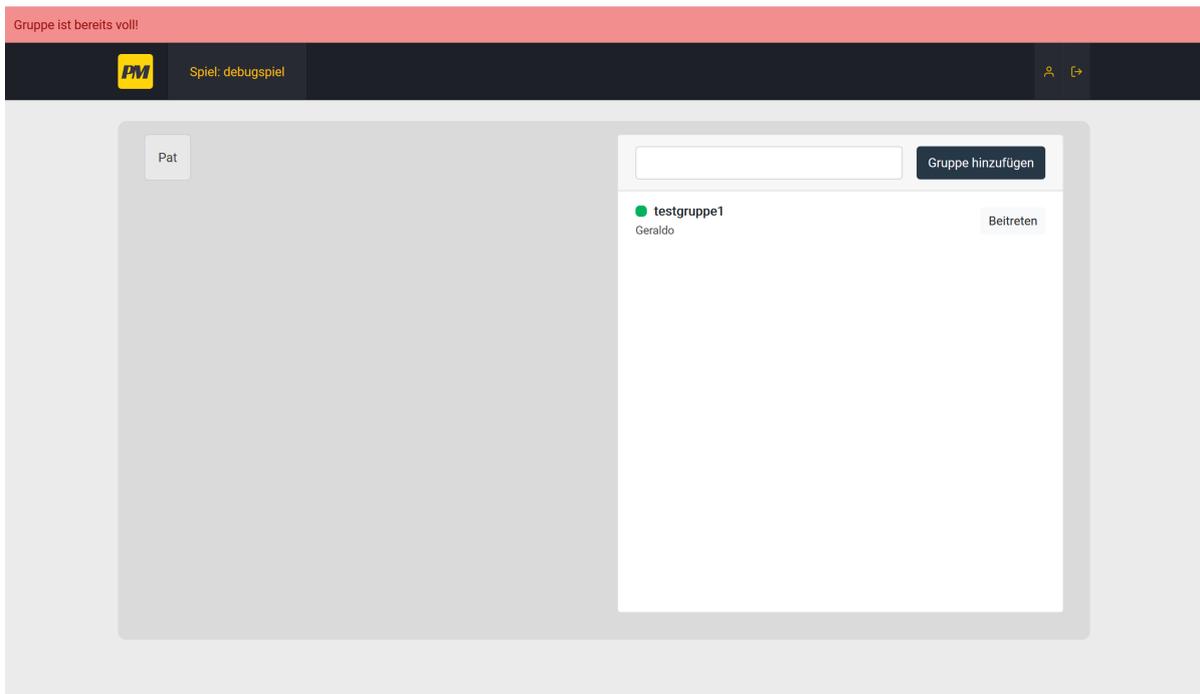


Abbildung 45: Beispiel für die Ausgabe einer Fehlermeldung (oben links).

5.16. Software-Metriken und Code Analyse

Insgesamt haben wir 5'830 Source Lines of Code geschrieben. Darin ist der Code für das Testing nicht enthalten. Für den Testing-Code haben wir total 5209 Source Lines of Code geschrieben.

Frontend	2'808		
Backend	2'842	Backend	4'521
Datenbank	180	Datenbankszenarien	688
Total	5'830	Total	5'209

Die Originalstatistiken sind in den Abbildungen 46, 47, 48 und 49 enthalten. Die Statistiken wurden mit dem npm-Modul „sloc“ erstellt.²⁰ Nicht enthalten in den Statistiken sind die Dateien für das Aufsetzen der Environments und Tools wie GitLab-CI, Dockerfiles, Server-Setup mit traefik, etc.

```
C:\Users\eli\ownCloud\Study\HS19_SA\repos\project-manager>sloc --format cli-table --keys source,comment,todo ./dist/views
```

Extension	Source	Comment	To Do
- Total -	1975	0	0
hbs	1975	0	0

```
C:\Users\eli\ownCloud\Study\HS19_SA\repos\project-manager>sloc --format cli-table --keys source,comment,todo --exclude bootstrap* ./dist/public/stylesheets
```

Extension	Source	Comment	To Do
- Total -	833	23	0
css	833	23	0

Abbildung 46: Frontend: Source Lines of Code.

²⁰Siehe Webseite <https://www.npmjs.com/package/sloc>.

```
C:\Users\eli\ownCloud\Study\HS19\_SA\repos\project-manager>sloc --format cli-table --keys source,comment,todo ./src
```

Extension	Source	Comment	To Do
- Total -	7363	349	7
ts	7363	349	7
scss	0	0	0

```
C:\Users\eli\ownCloud\Study\HS19\_SA\repos\project-manager>sloc --format cli-table --keys source,comment,todo ./src/tests
```

Extension	Source	Comment	To Do
- Total -	4521	292	6
ts	4521	292	6

Abbildung 47: Backend: Source Lines of Code total und Tests.

```
C:\Users\eli\ownCloud\Study\HS19_SA\repos\project-manager>sloc --format cli-table --keys source,comment,todo ./src/app.ts
```

Extension	Source	Comment	To Do
- Total -	28	1	0
ts	28	1	0

```
C:\Users\eli\ownCloud\Study\HS19_SA\repos\project-manager>sloc --format cli-table --keys source,comment,todo ./src/routes
```

Extension	Source	Comment	To Do
- Total -	82	3	0
ts	82	3	0

```
C:\Users\eli\ownCloud\Study\HS19_SA\repos\project-manager>sloc --format cli-table --keys source,comment,todo ./src/controller
```

Extension	Source	Comment	To Do
- Total -	953	13	0
ts	953	13	0

```
C:\Users\eli\ownCloud\Study\HS19_SA\repos\project-manager>sloc --format cli-table --keys source,comment,todo ./src/services
```

Extension	Source	Comment	To Do
- Total -	1031	36	0
ts	1031	36	0

```
C:\Users\eli\ownCloud\Study\HS19_SA\repos\project-manager>sloc --format cli-table --keys source,comment,todo ./src/repositories
```

Extension	Source	Comment	To Do
- Total -	730	3	1
ts	730	3	1

```
C:\Users\eli\ownCloud\Study\HS19_SA\repos\project-manager>sloc --format cli-table --keys source,comment,todo ./src/model
```

Extension	Source	Comment	To Do
- Total -	12	0	0
ts	12	0	0

```
C:\Users\eli\ownCloud\Study\HS19_SA\repos\project-manager>sloc --format cli-table --keys source,comment,todo ./src/utlis
```

Extension	Source	Comment	To Do
- Total -	6	1	0
ts	6	1	0

Abbildung 48: Backend: Source Lines of Code - Details.

```
C:\Users\eli\ownCloud\Study\HS19_SA\repos\project-manager>sloc --format cli-table --keys source,comment,todo ./knex
```

Extension	Source	Comment	To Do
- Total -	868	167	12
js	868	167	12

```
C:\Users\eli\ownCloud\Study\HS19_SA\repos\project-manager>sloc --format cli-table --keys source,comment,todo --exclude test ./knex
```

Extension	Source	Comment	To Do
- Total -	180	5	0
js	180	5	0

```
C:\Users\eli\ownCloud\Study\HS19_SA\repos\project-manager>sloc --format cli-table --keys source,comment,todo ./knex/seeds/test
```

Extension	Source	Comment	To Do
- Total -	688	162	12
js	688	162	12

Abbildung 49: Datenbank: Source Lines of Code.

Die Projektlaufzeit betrug knapp 3 Monate. Umgerechnet pro Person und Monat ergibt dies 970 produktive Source Lines of Code. Gemäss den Aussagen im HSR-Modul SE2 im FS2019²¹ deutet dies darauf hin, dass vermutlich an einen oder anderen Ort Copy-Paste verwendet wurde und zu wenig Refactorings gemacht wurde, oder der Code nicht sauber getestet wurde. Eine alternative Interpretation wäre, dass es hier um ein kleines Spitzenteam handelte.

Die Abbildung 50 zeigt als statische Code-Analyse einen Dependency-Graph des Backends. Die Abbildung wurde mit „Madge“ erstellt.²²

²¹Siehe Folie im Anhang in Abschnitt D.

²²Siehe Webseite <https://github.com/pahen/madge>; command-line: madge --image code-dependency-graph.svg ./dist/app.js

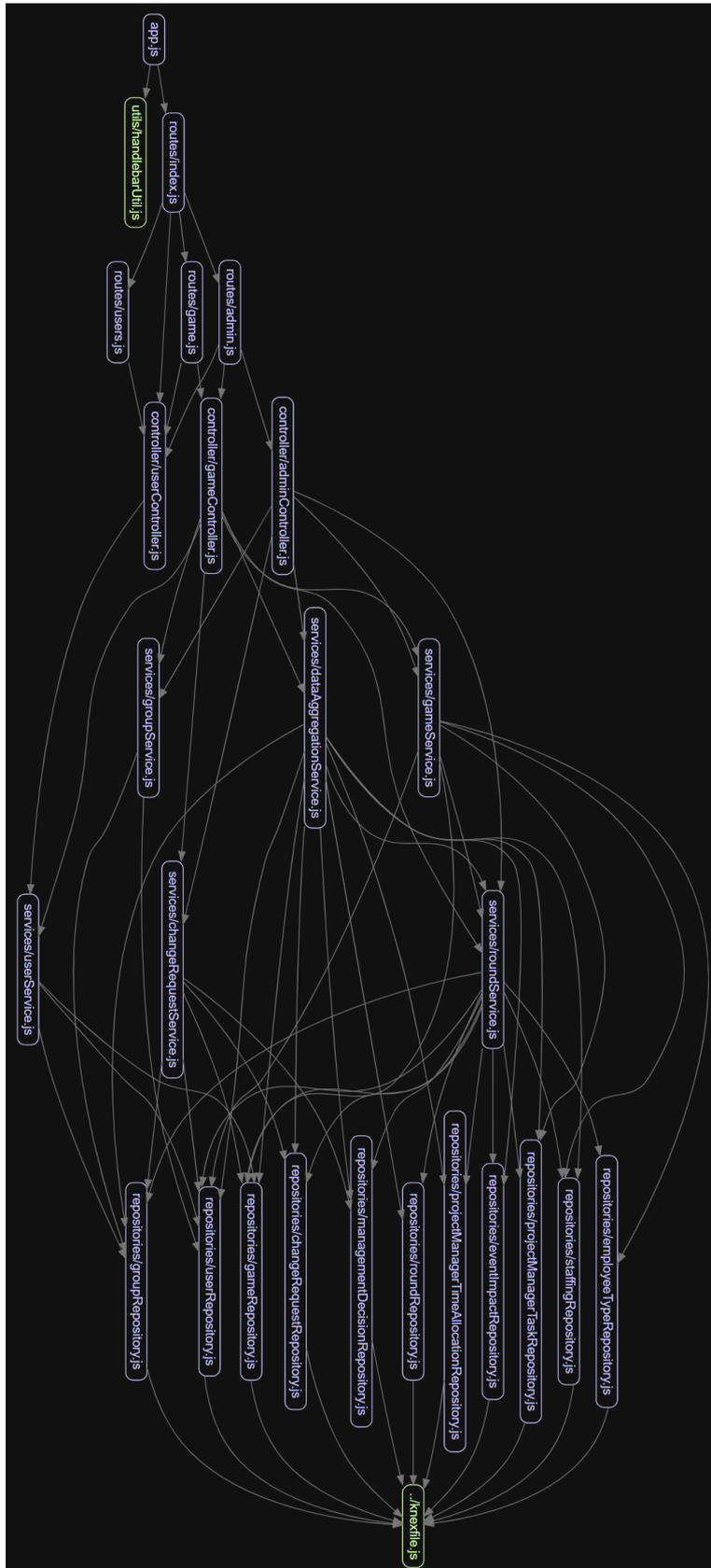


Abbildung 50: Dependency-Graph der Code-Struktur im Backend.

5.17. Sonstiges

5.17.1. Routing-Tabelle

Die Abbildung 51 zeigt das Routing in der Middleware für die wichtigsten Routes. Insbesondere ist daraus ersichtlich, welche Seiten in welchem Spielzustand erreichbar sind.

Diese Tabelle ist ca. Mitte Construction-Phase entstanden, da wir Schwierigkeiten hatten, die Managementeingaben der Teams abzuklemmen, wenn keine Runde geöffnet war. Nachdem wir das Routing systematisch erfasst hatten, konnten diese Probleme behoben werden.

Routing Table /game	Combine	Tests written?	active game check		active group check		game is not started yet		game is running		game is finished
			separate middleware 1	separate middleware 2	separate middleware 2	separate middleware 3	round not closed	round closed			
/game/staffing	Management Decisions -> same routing for all	yes	if assigned to game -> allow next() else -> redirect to /game/choose	if assigned to group -> allow next() else -> redirect to /game/lobby	if assigned to group -> allow next() else -> redirect to /game/lobby	deny -> redirect to /game/statistics	deny -> redirect to /game/statistics	deny -> redirect to /game/statistics	deny -> redirect to /game/statistics	deny -> redirect to /game/statistics	deny -> redirect to /game/statistics
/game/projectManager/TimeAllocation		yes									
/game/investments		yes									
/game/choose (displays view to choose a game)		yes	if not assigned to game -> allow next() else -> redirect to /game/lobby	if not assigned to group -> allow next() else ERROR	if not assigned to group -> allow next() else ERROR	GET: n/a POST: allow next()	GET: n/a POST: deny -> redirect to GET /game/choose	deny -> redirect to /game/statistics			
/game/lobby (displays view to join a group)		yes	if assigned to game -> allow next() else -> redirect to /game/choose	if not assigned to group -> allow next() else -> redirect to /game/start	if not assigned to group -> allow next() else -> redirect to /game/start	allow next()	deny -> redirect to /game/statistics	deny -> redirect to /game/statistics	deny -> redirect to /game/statistics	deny -> redirect to /game/statistics	deny -> redirect to /game/statistics
/game/		no				redirect to /game/start					
/game/start		yes	if assigned to game -> allow next() else -> redirect to /game/choose	if assigned to group -> allow next() else -> redirect to /game/lobby	if assigned to group -> allow next() else -> redirect to /game/lobby	allow next()	deny -> redirect to /game/statistics	deny -> redirect to /game/statistics	deny -> redirect to /game/statistics	deny -> redirect to /game/statistics	deny -> redirect to /game/statistics
/game/statistics		yes	if assigned to game -> allow next() else -> redirect to /game/choose	if assigned to group -> allow next() else -> redirect to /game/lobby	if assigned to group -> allow next() else -> redirect to /game/lobby	allow next()	allow next()	allow next()	allow next()	allow next()	allow next()
/game/group/add/groupid		yes	if assigned to game -> allow next() else -> redirect to /game/choose	if assigned to game -> allow next() else -> redirect to /game/choose	if groupid is assigned to the same game as in user.id_active_game -> allow next() else ERROR	allow next()	deny -> redirect to /game/start	deny -> redirect to /game/start	deny -> redirect to /game/start	deny -> redirect to /game/start	deny -> redirect to /game/start
/game/addGroup		yes	if assigned to game -> allow next() else -> redirect to /game/choose	if assigned to game -> allow next() else -> redirect to /game/choose	allow next()	allow next()	deny -> redirect to /game/start	deny -> redirect to /game/start	deny -> redirect to /game/start	deny -> redirect to /game/start	deny -> redirect to /game/start

6. Projektdokumentation

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

6.1. Projektausschreibung

Der Vollständigkeit halber dokumentieren wir hier, wie wir zu diesem Projektthema gekommen sind.

Im Arbeitsverwaltungstool der HSR war die Arbeit wie in Abbildung 52 gezeigt ausgeschrieben.²³

Projekt-Game programmieren	Dozent: Claudio Fuchs Student(en): 2
<p>Diese Arbeit erstellt eine Simulation (Spiel) für das Modul PmQm. In dieser Simulation treten verschiedene Studententeams gegeneinander an und müssen rundenbasiert Entscheidungen treffen. Nach einer definierten Rundenanzahl wird der Sieger erkoren.</p>  <p>Das Team benötigt Erfahrung in der Software Programmierung, um diese Applikation webbasiert umzusetzen.</p>	<ul style="list-style-type: none">■ Ich als Spielleiter lanciere ein Spiel mit 1-n Runden■ Ich als Spielleiter terminiere die Runden jeweils■ Ich als Spielleiter kann pro Spiel Vorgaben an Budget, Zeit, Function Points, Projektziele, Risiken, Ausbildung, Anzahl Runden und Stakeholder geben■ Ich als Spielleiter kann die Teams einrichten (1-n Spieler pro Team)■ Ich als Spielleiter habe ein Dashboard um die Eingabe aller Teams zu überwachen■ Ich als Spielleiter habe die Möglichkeit, die Teams via Email zu informieren■ Ich als Spielleiter habe die Möglichkeit, Reports hinsichtlich Burn Down Chart oder Kostenentwicklung zu exportieren■ Ich als Spielleiter lege fest, was die Teams genau pro Runde einzugeben haben■ Ich als Spielleiter lege fest, wie fest die Ausbildung pro Runde wirksam ist■ Ich als Spielleiter lege fest, wann welche Risiken eintreten■ Ich als Spielleiter lege fest, wann der Lenkungsausschuss stattfindet■ Ich als Spielleiter lege fest, ob die Changes vom Lenkungsausschuss genehmigt werden■ Ich als Spieler gebe meinem Team einen Fantasie Namen■ Ich als Spieler gebe jede Runde die vom Spielleiter verlangten Parameter ein■ Ich als Spieler definiere, wie viel pro Runde in Risiko-Reduktion investiert wird■ Ich als Spieler definiere, wie viele Personentage pro Runde gearbeitet werden■ Ich als Spieler definiere, wie viel pro Runde in Ausbildung investiert wird■ Ich als Spieler definiere, für was der Projektleiter seine Zeit verwendet■ Ich als Spieler definiere, wann ein Change Request gestellt wird■ Ich als Spieler habe einen groben Überblick, wie weit die anderen Teams sind (Details nur für Spielleiter)■ Ich als Spieler werde per E-Mail informiert, wenn für meine Gruppe eine Eingabe ausstehend ist

1

Claudio Fuchs, PmQm, Rapperswil, Herbst 2018

Abbildung 52: Ausschreibungstext aus dem HSR-Arbeitsverwaltungstool.

²³Quelle: Claudio Fuchs, PmQm, Rapperswil, Herbst 2018

6.2. Aufgabenstellung



Studienarbeit: Projekt-Game

Einführung	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.
Aufgabe	<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"> 1. Gibt es in der Literatur oder bei anderen Instituten Hinweise darauf, um wieviel effizienter spielerisches Lernen ist? 2. Können die Anforderungen des Dozenten in einer Software umgesetzt werden? Begründung, wenn Anforderungen nicht wie gefordert umgesetzt werden konnten 3. Könnte die HSR diese Software in Form einer -Trainingsdienstleistung im freien Markt anbieten, um Software-Unternehmen im Bereich Projektmanagement fitter zu machen?
Hinweise	Es besteht eine manuelle Spielabwicklung in Excel als Vorversion, welche den Studierenden vorab abgegeben wurde. In den Übungsstunden vom 21. November, 28. November und 5. Dezember besteht die Möglichkeit, eine erste Version des Games mit den Studierenden auszuprobieren. Damit diese Software jedes Semester wiedereingesetzt werden kann, organisieren die Studierenden einen Server, auf welchem diese Software mindestens drei Jahre lang betrieben werden kann (der Dozent sponsert den Domain-Namen).
Erwartete Resultate	<ul style="list-style-type: none"> • Erwartet wird die Beantwortung der Fragestellung in Form eines Berichts und im Speziellen die Fragestellung 2 als realisierte Software. Die Software wird einerseits als implementierte Lösung abgegeben, andererseits aber auch in Form von Source-Code und Dokumentation. • 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. • Die investierten Arbeitsstunden müssen personen- und datumsbezogen dokumentiert werden. Der Rapport wird im Anhang des Berichts beigelegt. • Es muss auf Basis des Stundenrapports oder der Kapitel im Bericht ersichtlich sein, wer an was gearbeitet hat.

6.3. Zeitplan

Der Zeitplan enthält die Grobplanung und externen Deadlines. Die Feinplanung mit höherem Detaillierungsgrad wurde jeweils in GitLab mit den Sprints gemacht.

Datum	KW	SW	Phase	Sprintnummer	Feiertage	Abgabe / Deadlines	Meeting
So 01.09.2019							
Mo 02.09.2019							
Di 03.09.2019							
Mi 04.09.2019	36	n/a					11:00 - 11:30 (Skype): 1. informelles Meeting
Do 05.09.2019							
Fr 06.09.2019							
Sa 07.09.2019							
So 08.09.2019							
Mo 09.09.2019			n/a	n/a			
Di 10.09.2019							
Mi 11.09.2019	37	n/a					
Do 12.09.2019							
Fr 13.09.2019						Einrichten der Arbeitsplätze in den Labors	18:00 (Skype): Internes Meeting
Sa 14.09.2019							
So 15.09.2019							
Mo 16.09.2019						Ausgabe der Aufgabenstellung durch den Betreuer	
Di 17.09.2019							
Mi 18.09.2019	38	1	Initiation	Sprint 1		Projektplan-Dokument rudimentär aufgesetzt	
Do 19.09.2019						Erste Projektschwerpunkte bestimmen, siehe Protokoll vom 3.9.2019, Administrative Arbeitspakete erfasst, Schätzung erfasst, Alle Use Cases liegen im Brief Format vor, GitLab eingerichtet (ohne CKCD), Meilensteine definieren	12:10 - 13:00 (HSR, Raum 1.281): 1. Face-to-Face-Meeting
Fr 20.09.2019							
Sa 21.09.2019							
So 22.09.2019							
Mo 23.09.2019						Domainmodel erster Entwurf, GUI Skizzen erster Entwurf, NIA erster Entwurf, Architektur erster Entwurf	
Di 24.09.2019							
Mi 25.09.2019	39	2	Elaboration	Sprint 1		Use Cases überarbeitet, Domainmodel überarbeitet, GUI Skizzen überarbeitet, NIA überarbeitet, Architektur überarbeitet	Sprint Planning Skype Call (Nachmittag)
Do 26.09.2019							
Fr 27.09.2019							
Sa 28.09.2019							
So 29.09.2019							
Mo 30.09.2019							Sprint Review, Sprint Retrospective, Daily
Di 01.10.2019							
Mi 02.10.2019	40	3	Elaboration	Sprint 2			Daily
Do 03.10.2019							Sprint Planning
Fr 04.10.2019							
Sa 05.10.2019							
So 06.10.2019							
Mo 07.10.2019							Daily
Di 08.10.2019							
Mi 09.10.2019	41	4	Elaboration	Sprint 2			Daily
Do 10.10.2019							Sprint Planning (Nur am Morgen)
Fr 11.10.2019						Server ist aufgesetzt und betriebsbereit.	
Sa 12.10.2019							
So 13.10.2019							
Mo 14.10.2019							Sprint Review, Sprint Retrospective, Daily
Di 15.10.2019							
Mi 16.10.2019	42	5	Elaboration	Sprint 3			Daily
Do 17.10.2019							
Fr 18.10.2019							
Sa 19.10.2019							
So 20.10.2019							
Mo 21.10.2019							
Di 22.10.2019							
Do 24.10.2019	43	6	Construction	Sprint 3			Daily
Fr 25.10.2019							Sprint Planning
Sa 26.10.2019							
So 27.10.2019							Sprint Review, Sprint Retrospective, Daily
Mo 28.10.2019							
Di 29.10.2019							
Mi 30.10.2019	44	7	Construction	Sprint 4			Daily
Do 31.10.2019						Alleinheiten	
Fr 01.11.2019							
Sa 02.11.2019							
So 03.11.2019							
Mo 04.11.2019							Daily
Di 05.11.2019							
Mi 06.11.2019	45	8	Construction	Sprint 4			Sprint Planning Skype Call (Nachmittag: 14:00 Uhr), Daily
Do 07.11.2019							
Fr 08.11.2019							
Sa 09.11.2019							
So 10.11.2019							
Mo 11.11.2019							Sprint Review, Sprint Retrospective, Daily
Di 12.11.2019							
Mi 13.11.2019	46	9	Construction	Sprint 5			Daily
Do 14.11.2019							
Fr 15.11.2019							
Sa 16.11.2019							
So 17.11.2019							
Mo 18.11.2019							Daily
Di 19.11.2019							
Mi 20.11.2019	47	10	Construction	Sprint 5			Daily
Do 21.11.2019						Live Test Unterricht	Sprint Planning: 11:40 Uhr
Fr 22.11.2019							
Sa 23.11.2019							
So 24.11.2019							
Mo 25.11.2019							Sprint Review, Sprint Retrospective, Daily
Di 26.11.2019							
Mi 27.11.2019	48	11	Construction	Sprint 6			Daily
Do 28.11.2019						Live Test Unterricht	
Fr 29.11.2019							
Sa 30.11.2019							
So 01.12.2019							
Mo 02.12.2019							Daily
Di 03.12.2019							
Mi 04.12.2019	49	12	Construction	Sprint 6			Daily
Do 05.12.2019						Live Test Unterricht	Sprint Planning
Fr 06.12.2019							
Sa 07.12.2019							
So 08.12.2019							Sprint Review, Sprint Retrospective, Daily
Mo 09.12.2019							
Di 10.12.2019							Daily
Mi 11.12.2019	50	13	Transition	Sprint 7			
Do 12.12.2019							
Fr 13.12.2019							
Sa 14.12.2019							
So 15.12.2019							
Mo 16.12.2019						Erfassung des Abstracts im Online-Tool https://abstract.hsr.ch/ (Freigabe des Abstracts zum Review an Betreuer)	Daily
Di 17.12.2019							
So 18.12.2019	51	14	Transition	Sprint 7		Betreuer gibt das Dokument mit dem korrekten und vollständigen Abstract zur Weiterverarbeitung an das Studiengangsekretariat frei.	Daily
Do 19.12.2019							
Fr 20.12.2019						Hochladen aller Dokumente bis 17:00 Uhr	
Sa 21.12.2019							
So 22.12.2019							

6.4. Projektplan

Durch Diskussionen mit Peers über das Vorgehensmodell Scrum+ sei folgende Ergänzung erwähnt: Wir verstehen unter Scrum+, dass auch nach End of Elaboration noch Änderungen an den funktionalen Anforderungen, Domain Model und Architektur möglich sind, diese jedoch nicht grundlegender Natur sein sollten.

Unser vollständiger Projektplan befindet sich im Anhang B „Projektplan“.

6.4.1. Kritischer Rückblick: Vergleich Ist vs. Soll

Nachfolgend werfen wir einen kritischen Blick auf den Projektverlauf mit Bezug auf den Projektplan. 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 noch nicht beantwortet werden
 - ✓ Ziele 2-5 wurden erarbeitet
- Projektorganisation
 - ✓ Wir haben uns bis auf 2-3 Ausnahmen jeden Montag und Mittwoch sowie Dienstags über den Mittag getroffen.
- Managementabläufe
 - ✓ Wir haben das Vorgehensmodell Scrum+ grösstenteils eingehalten. Lediglich kurz vor dem 1. und 2. Live-Test hatten wir Schwierigkeiten, Zeit zu finden, die Scrum-Meetings sauber durchzuführen.
 - ✓ Wir haben konsequent zu jedem Sprint Planning, Sprint Review und Sprint Retrospective ein Protokoll geschrieben
 - ✓ Dass wir uns Leitplanken gesetzt haben durch das Einbeziehen von Unified Process, hat uns geholfen.
 - Zeitabrechnung: siehe Abschnitt Projektcontrolling
 - ✓ Die Review-Termine mit dem Betreuer haben wir alle eingehalten. Vor den Live-Tests hätten wir uns proaktiver beim Betreuer melden können, und so besser koordinieren können, welche Features am Live-Test präsent sein werden.
 - × Da wir bis Mitte der Construction-Phase Probleme hatten, die Software auf einen öffentlich zugänglichen Server zu deployen (aufgrund des von der HSR gesperrten Docker-Ports), kamen wir nie in den Fluss, laufend die neusten Features dem Betreuer zugänglich zu machen. Eigentlich war geplant, dass

der Betreuer jederzeit eine stabile Version auf der Live-Umgebung und eine explorative Version auf der Staging-Umgebung zugänglich hat.

- Risikomanagement
 - ✓ Wir haben bis End of Elaboration eine Liste mit den technischen Projektrisiken erfasst und als Chart dargestellt.
 - × Es war geplant, die Risikoliste in jedem Sprint 1mal zu aktualisieren. Dies haben wir nur alle 3 Sprints gemacht.
- Infrastruktur
 - ✓ Dank der Erfahrung aus dem Engineering-Projekt haben wir die CI/CD-Pipeline bereits zu Beginn erstellt. Die CI/CD-Pipeline hat uns stark geholfen bei der Entwicklung einer sauberen Software.
- Qualitätsmassnahmen
 - × Die Definition of Done haben wir nicht immer 100% befolgt. Insbesondere vor den ersten beiden Live-Tests wurden wir ein wenig nachlässig beim Testing und der Dokumentation nachführen
- Qualitätsmassnahmen: Generelle Qualitätsmassnahmen
 - × Code-Reviews im Team haben wir eher selten durchgeführt.
 - ✓ Die Sprint Retrospektive haben wir konsequent durchgeführt und auch mit der nötigen Ernsthaftigkeit. Wir haben uns im Verlauf sogar ein Schema erarbeitet (3 Fragen), sowie 1 Takeaway aus jeder Sprint Retrospektive in den nächsten Sprint genommen.
 - ✓ Das Sprint Goal haben wir in jedem Sprint festgehalten. Wir haben im Verlauf auch begonnen, das Sprint Goal als Checkliste zu formulieren.
 - ✓ Alle Projektdokumente waren für jedes Projektmitglied jederzeit erreichbar.
 - × Arbeitspakete mit dem Label „#Development“, „#Review“ oder „#Testing“ haben wir nicht konsequent mit der Definition of Done als Checkliste in GitLab erfasst.
- Qualitätsmassnahmen: Code-Qualität
 - × Unit-Tests und vor allem Integrationstests haben wir durchgeführt, jedoch die vorgesehene Test-Coverage von 65% nicht erreicht. Insbesondere vor den ersten beiden Live-Tests wurden wir beim Testing nachlässig.
 - × Kommentare im Code: An einigen Stellen haben wir unnötige Kommentare im Code (Stichwort: unaufgeräumte Baustelle).
 - ✓ Den Github-Flow haben wir grösstenteils befolgt. Nur haben wir zu stark auf

dem develop-Branch gearbeitet und zu selten auf den staging- und master-Branch gemergt.

- × Wir haben keine Merge-Request-Templates verwendet. Bei der Aufsetzung des GitLab-Repos in der Elaboration-Phase haben wir dies nicht explizit als Arbeitspaket drin gehabt und später in der Construction-Phase haben wir es ignoriert. Wir hätten entweder ein separates Arbeitspaket „Merge-Request-Template in GitLab einbinden“ machen oder in Checklistenform im Arbeitspaket „GitLab-Repo aufsetzen“ auflisten sollen.
- ✓ 4-Augen-Prinzip für Merges haben wir umgesetzt. Jedoch schriftliche Kommentare zum Code haben wir nur selten hinterlassen, da das Projektteam mit 2 Mitgliedern ja sehr klein war.
- ✓ Mit Continuous Integration hatten wir zu Beginn des Projekts mühe: Die Merge Requests fanden zu selten statt, die Commits waren zu gross. Wir haben dies dann in der Sprint Retrospektive besprochen und konnten uns so im Projektverlauf stetig verbessern.
- ✓ Codestyle verwendet
- Qualitätsmassnahmen: Testing
 - ✓ Unit-Tests und Integration-Tests wurden gemacht
 - ✓ Live-Test im Unterricht durchgeführt
 - × Den Performance-Test haben wir nur in Form des Live-Tests im Unterricht durchgeführt
 - × Für einen koordinierten Systemtest mit allen Projektmitgliedern hat die Zeit gefehlt

6.5. Projektcontrolling

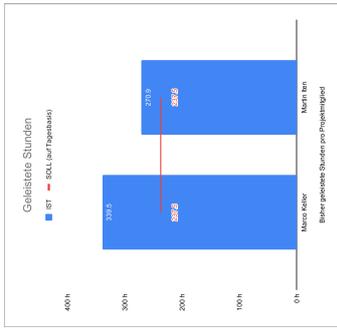
6.5.1. Erstellung des Controlling-Cockpits

Das Cockpit für das Projektcontrolling in Abbildung 53 haben wir in der Inception-Phase aufgestellt. Die Zeiterfassung auf die Arbeitspakete haben wir in unserem GitLab-Repo gemacht. Danach haben wir uns ein Python-Script geschrieben, welches die Zeitdaten über die GitLab-API in eine lokale SQLite-Datenbank speichert. Von dort haben wir mit einem weiteren Python-Script die Daten ausgewertet und automatisch in ein Google-Spreadsheet übertragen. Das Google-Spreadsheet hat dann daraus automatisch die Diagramme aktualisiert.²⁴

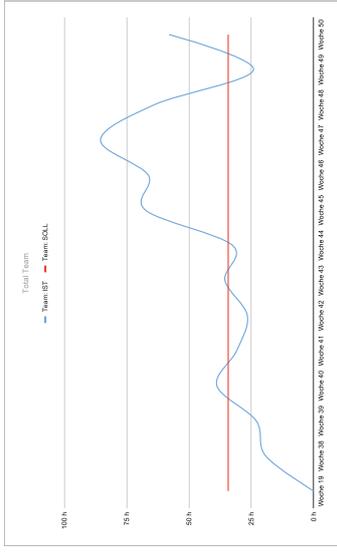
²⁴Die Google-API erlaubt 100 Requests pro 100 Sekunden ohne spezielle Registrierung. Entsprechend mussten wir die Geschwindigkeit der Zugriffe drosseln. Die Aktualisierung des Cockpits benötigt nun in Woche 14 rund 10 Minuten.

Plan-Budget ca. 600 Stunden
 Ist-Gesamt: 19.12.2019 05:00

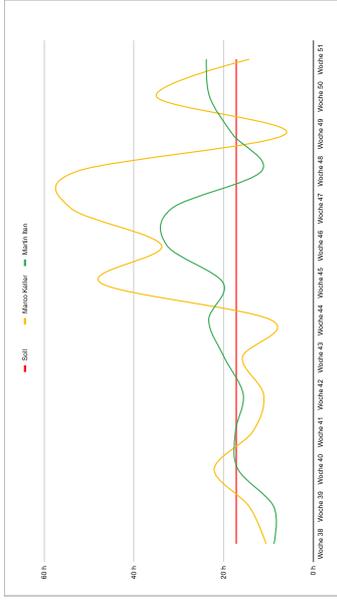
Zeit Total



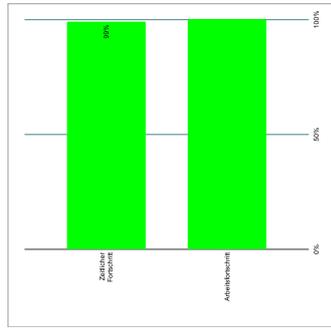
Zeit pro Woche (Team)



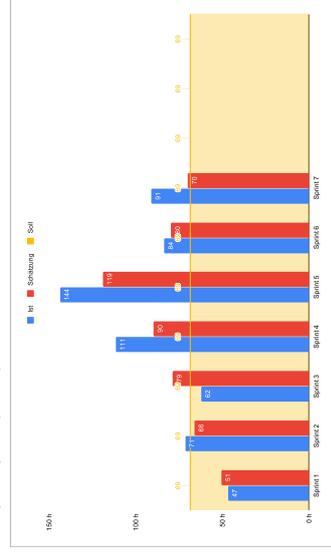
Zeit pro Woche pro Projektmitglied



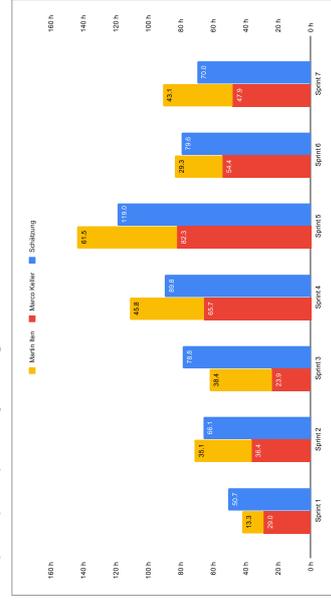
Fortschritt



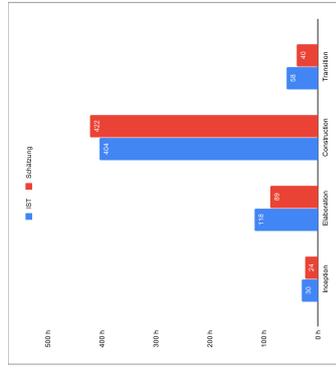
Zeit pro Sprint (Team)



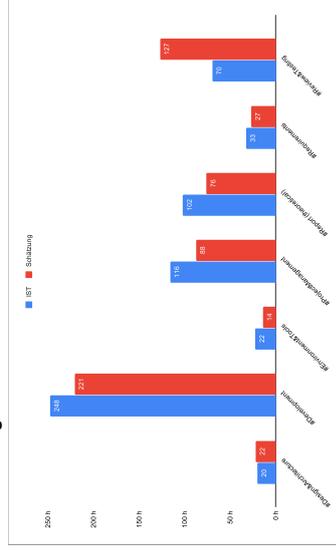
Zeit pro Sprint pro Projektmitglied



Zeit nach Phasen



Zeit nach Tätigkeiten



Risikomatrix

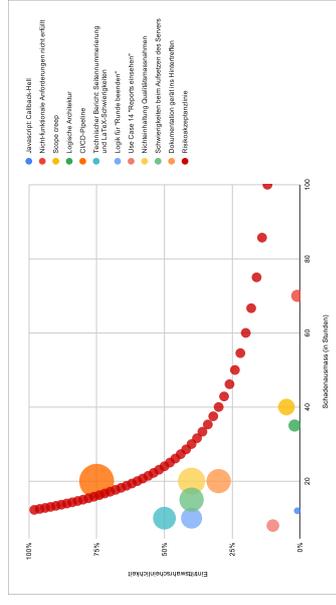


Abbildung 53: Projektcontrolling der Studienarbeit.

Die Fortschrittsanzeige ist experimenteller Natur. Denn um eine aussagekräftige Prozentzahl für den Fortschritt zu erhalten, hätten wir mindestens 1 Mal pro Sprint eine saubere Restaufwandschätzung machen müssen. Aus Zeitgründen haben wir jedoch darauf verzichtet. Grundsätzlich könnte man das auch automatisieren, sofern man alle Arbeitspakete konsequent in GitLab (Product Backlog) erfasst und mit einem „Tag“ versieht, ob es in Scope ist für die aktuelle Studienarbeit oder für das Nachfolgeprojekt.

6.5.2. Projektbudget

Aus Eigeninteresse rechnen wir hier die Stunden in effektive Kosten um, indem wir einen hypothetischen externen Stundenansatz von CHF 110 / h zu Grunde legen. Insgesamt waren 480 Stunden geplant.²⁵ Dies entspricht einem Projektbudget von CHF 52'800. Total stehen am Schluss rund 610 Stunden zu Buche. Dies entspricht Kosten von CHF 67'100. Wir haben also das Budget um CHF 14'300 überschritten. Trotzdem haben wir noch nicht alle User stories umgesetzt (siehe Abbildung 6 für die User stories).

Wieso konnten wir das Projektbudget nicht einhalten?

Wir haben viele der Arbeitspakete zeitlich unterschätzt. Im Nachhinein betrachtet war es nicht realistisch, dass wir alle spezifizierten User Stories innerhalb der Studienarbeit implementieren können. Interessant ist, dass wir fast in jedem Sprint mehr gearbeitet haben als wir geschätzt haben. Und dennoch sind wir fast in keinem Sprint mit allen Arbeitspaketen fertig geworden. Insbesondere haben wir die Testing-Arbeitspakete unterschätzt. Wir haben gelernt, dass für das Testing mindestens ähnlich viel Zeit benötigt wird wie für das Implementieren der Funktionalität. Zudem haben wir zu wenig Zeit für die Live-Tests einkalkuliert. Dort haben wir den administrativen Aufwand unterschätzt.

6.6. Risikoliste

Die Risikoliste in Abbildung 54 haben wir in der Elaboration-Phase aufgestellt. Die Risikoliste beinhaltet ausschliesslich technische Risiken. Risiken wie zum Beispiel „Ein Projektmitglied wird krank“ sind somit nicht abgebildet.

Die Risiken haben wir visuell im Controlling-Cockpit aufbereitet, inklusive der Risikoakzeptanzlinie (siehe in Abbildung 53 unten rechts).

6.6.1. Eintretene Risiken

Vor dem zweiten Live-Test ist ein Risiko eingetreten. Bei der Übungsgruppe 1 konnten wir die Software nicht in der aktuellsten Version zum Laufen bringen. Grund dafür eine falsch gesetzte Umgebungsvariable in Bezug auf die Datenbank. Der Server hat dann beim Aufstarten die Datenbank nicht gefunden. Dadurch ist beim Push auf den

²⁵Im Projektplan in Kapitel 5.2 „Kostenvoranschlag“ beschrieben, siehe Abschnitt 6.4.

Risikomanagement: Technische Risiken

Projekt:	Project Manager
Erstellt am:	01.10.2019
Zuletzt aktualisiert am:	09.12.2019
Autor:	Marco Keller, Martin Iken
Gewichteter Schaden (in Stunden):	48,32

Risikoakzeptanzlinie
 Es werden nur Risiken überführt, die einen **gewichteten Schaden von maximal 1,6 Arbeitstagen** (12 Stunden) haben. **die Summe der gewichteten Schäden die Länge eines Sprints (7m)** nicht übersteigt. Beachte: Aus Darstellunggründen enthält die Spalte "gewichteter Schaden" für die Risikoakzeptanzlinie nicht den gewichteten Schaden. Die Biasvorgabe (12h) wurde, sonst die Risikomatrix unleserlich machen, da die anderen Punkte überdeckt würden. Das Weiteren umfasst die Summe der gewichteten Schäden selbstverständlich nicht die Punkte der Risikoakzeptanzlinie.

Nr	Titel	Beschreibung	max. Schaden [h]	Eintrittswahrscheinlichkeit	Gewichteter Schaden	Risikostrategie abwägen, vermeiden, selbst tragen	Vorbereitung (Maßnahmen)	Verhalten beim Eintreten
R1	JavaScript, Callback-Hell	Komplexität des Codes, Front-End, Code, wird weniger verständlich, was zu längeren Arbeitszeiten bei Programmierupdates führt.	12	1%	0,12	vermeiden	Minimierung der Komplexität hinsichtlich der Benutzung von "async-await". Verwendung von abstrakten Daten anstelle von Callbacks.	
R2	Nicht-funktionale Anforderungen nicht erfüllt	Es stellt sich heraus, dass die getriebene und implementierte Infrastruktur die nicht-funktionalen Anforderungen nicht erfüllt.	70	1%	0,7	reduzieren	1.) Nicht-funktionale Anforderungen zweckmäßig definieren. Von der Applikation sind beispielsweise keine Geschäftsprozesse betroffen, es werden keine persönlichen Daten erfasst, etc. Daswegen können die nicht-funktionalen Anforderungen minimal gehalten werden. 2.) Nicht-funktionale Anforderungen "SMART" spezifizieren (mindestens "S" (spezifisch) und "M" (messbar))	
R3	Scope creep	Wichtige Anforderungen werden in der Elaborationsphase übersehen	40	5%	2	reduzieren / abwägen	1.) Anforderungenung gemäß HSB (hoch) SEI und SE2 sowie ProCom. 2.) Negativlist ("Addendum-Text") für Kernfunktionalität durchführen nachweiliger Erhebung der Anforderungen (die Anforderungen regieren und systematisch mit dem Product Owner durchgehen a la: "Ist ist du tatsächlich NICHT so gemeint?")	In nach Aufwand die Zusatzfunktionalität aus dem Scope (reduzieren), oder in späteren Releases verschieben (Risikostategie Abwägen).
R4	Logische Architektur	Die in der Elaborationsphase definierte logische Architektur (Layers-Schichtenprogramm) erweist sich in der Constroctiophase als suboptimal.	35	2%	0,7			
R5	CI/CD-Pipeline	Das Projektteam hat wenig Erfahrung mit dem Aufsetzen von Continuous Integration (CI) / Continuous Deployment (CD) Servern, im Ergänzungsprojekt hatten andere Programmierer diese Tasks übernommen.	20	75%	15	reduzieren	Genügend Zeit bereitstellen im betreffenden Arbeitpaket, um sich in das Thema einzesen zu können.	Experten einholen bei Peers, die Erfahrung im Bereich CI/CD haben.
R6	Technischer Bericht: Seriennummerierung und LUTax-Schwierigkeiten	Die HSB gibt in den Richtlinien zur Dokumentation der Struktur an, dass alle Seiten eine "unique identifier" haben müssen. Da wir sowohl Dokumente in Google Drive erstellen als auch in GitLab-Repositorys und Grafikools, könnte dies Probleme geben, da wir alles in einem einzigen Bericht abgeben müssen.	10	50%	5	vermeiden	Beide in der Elaborationsphase des Strategie zu recht legen, wie alle einzelnen Dokumente einfach zu einem einzigen Bericht mit richtliniengestauter Seitennummerierung zusammengeführt werden können.	Experten einholen bei Peers, die viel Erfahrung mit LUTax haben
R7	Logik für "Runde benden"	Untersuchung der Komplexität respektive nur wagen Erfassung der Logik zum Ermitteln des Zwischenstands nach einer gespielten Runde.	10	40%	4	reduzieren	Neben dem Ermitteln des Domain Modells in UML "welches mit dem an der HSB gehalten. Vorgehensweise häufig zu datenzentriert ist" (Zitat Olaf Zimmermann, Approach-Vorlesung vom 22.02.2019) sollen auch weitere Herangehensweisen herangezogen werden (beispielsweise das C4 Model).	
R8	Use Case 14 "Reports einsehen"	Untersuchung der Komplexität des Use Case "Reports einsehen". Das Projektteam hat noch keine Erfahrung mit dem Erstellen von Charts in Vue.js (Frontend-Framework).	8	10%	0,8	selbst tragen	-	
R9	Nichternhaltung Qualitätsmaßnahmen	Qualitätsmaßnahmen und -richtlinien werden aus Zeitdruck nicht eingehalten, was schlussendlich zu Fehlern und Inflation der Zusatzarbeitszeit führt.	20	40%	8	reduzieren	1.) Wenige, aber dafür griffige Qualitätsmaßnahmen schriftlich festhalten. 2.) Die Einhaltung dieser Maßnahmen durch die Teammitglieder vor einem Meeting sollte ein Kontrollpunkt durch ein zweites Projektmitglied beinhalten. 3.) Wiederholte auf das Erhalten der Qualitätsmaßnahmen hinweisen.	
R10	Schwierigkeiten beim Aufsetzen des Servers	Es treten Probleme beim Aufbau der technischen Plattform auf (Server mit Backend, Frontend und Datenbank)	15	40%	6	reduzieren / selbst tragen	1.) Server Zeit bereitstellen, um sich in die Thematik einzulesen 2.) Möglichst einfache Struktur verwenden	Experten einholen
R11	Dokumentation geht ins Hintertreffen	Die Dokumentation fällt zu Gunsten der Programmierarbeit zurück. Die nötige Qualität für die Studienarbeit kann nicht erreicht werden.	20	30,00%	6	reduzieren	1.) Stetige Überwachung des Fortschritts der Arbeiten 2.) "Dokumentation auf dem aktuellen Stand" in Definition of Done aufnehmen. 3.) Dokumentation auf dem aktuellen Stand in Definition of Done aufnehmen. 4.) Definition of Done in GitLab in jedes Arbeitpaket schreiben als Checklist	
Kosten	Summe					48,32		

Abbildung 54: Projektcontrolling der Studienarbeit.

master-Branch die CI/CD-Pipeline abgebrochen, was uns davor bewahrt hat, eine nicht funktionierende Software auf die Live-Umgebung zu pushen. Dadurch war immerhin die vorherige Version noch online.

Man kann nun argumentieren, dass dieses Risiko nicht eingetreten wäre, wenn wir die im Projektplan²⁶ definierten Qualitätsmassnahmen vollständig eingehalten hätten. Siehe dazu das Risiko R9 „Nichteinhaltung Qualitätsmassnahmen“ in der Risikoliste. In den Qualitätsmassnahmen haben wir definiert, dass wir den Github-Flow anwenden und Continuous Integration und Continuous Deployment machen. Insbesondere vor den ersten beiden Live-Tests haben wir kein (oder zumindest zu wenig) CI/CD betrieben. Wir haben fast ausschliesslich rund um den develop-Branch gearbeitet und nur selten auf den staging- und master-Branch gepusht. Wären wir vor den Live-Tests immer auch auf den master-Branch gegangen, hätten wir die falsch gesetzte Umgebungsvariable früher entdeckt.

Man könnte nun noch einen Schritt weitergehen und argumentieren, dass aufgrund von Hindernissen beim Aufsetzen des Servers²⁷ zu Beginn nie so richtig in den Continuous Deployment-Modus gekommen sind. Siehe dazu das Risiko R10 „Schwierigkeiten beim Aufsetzen des Servers“ in der Risikoliste.

Zudem sind wir nicht wie vorgesehen in jedem Sprint die Risikoliste 1 Mal durchgegangen. Möglicherweise hätten wir uns dadurch erinnert gefühlt, die Qualitätsmassnahmen besser einzuhalten.

Weitere eingetretene Risiken sind die Nichterfüllung der nicht-funktionalen Anforderungen (siehe dazu das Testprotokoll in Abschnitt 5.11.2). Zudem haben wir die Bearbeitung der theoretischen und wirtschaftlichen Fragestellung sowie der Aufsetzung des Schlussdokuments mehrmals nach hinten verschoben zugunsten der Implementierung neuer Features. Alle eingetretenen Risiken sind in der Abbildung 55 aufgeführt.

²⁶Siehe Kapitel 8 „Qualitätsmassnahmen“ im Projektplan im Abschnitt 6.4 sowie Abschnitt „Kritischer Rückblick: Vergleich Ist vs. Soll“.

²⁷Der Docker-Port auf dem HSR-Server war gesperrt, sodass wir Alternativen suchen mussten, um unser Docker-Image auf den Server zu schicken.

Eingetretene Risiken im Projektverlauf					
Eintrittsdatum	Risikozahl	Beschreibung	Auswirkung	Maßnahmen	Stundenaufwandsabschätzung
13.12.2019	R2	Nicht-funktionale Anforderungen nicht erfüllt	-	Nach unserem Ermessen ist die Auswirkung dieses Risikos so gering, dass keine Maßnahmen getroffen werden.	0
28.11.2019	R9	Nichterhaltung Quelltextmaßnahmen	2. Livertest in der Umlangruppe 1 konnte nicht durchgeführt werden, da die notwendigen Softwareversionen gesperrt werden. Das erhaltene Docker-Image konnte vom IISR-Server aus nicht aus dem GitLab-Registry geholt werden.	Wir haben den Fehler eruiert und beheben. Das Projektteam wurde durch diesen Vorfall sensibilisiert, die Dokumentation zu verbessern. Zudem wurden Maßnahmen nötig, die Datenabwicklung aus dem Live-Test manuell gemacht werden musste.	ca. 12
22.10.2019	R10	Schwergkeiten beim Aulesetzen des Servers. Der für Docker notwendige Port war von der IISR blockiert.	Das erhaltene Docker-Image konnte vom IISR-Server aus nicht aus dem GitLab-Registry geholt werden.	Wir mussten eine externe Registry (Docker-Hub) verwenden. Das resultierte in Zeitsatzarbeit.	5
11.11.2019	R11	Dokumentation geht ins Hintertreffen	Die Dokumenten konnte nicht vorzeitig zu einem Review an den Betreuer gegeben werden.	-	0

Abbildung 55: Eingetretene Risiken.

Glossar

admin Beschreibt die Rolle des Spielleiters..

Aufwand Ein Aufwand beschreibt eine Aufgabe (Task) des Projektleiters. Die Gruppen entscheiden, wie viel Prozent der Projektleiter für welche Aufgaben investiert..

Employee Type Cost Beschreibt die Kosten pro Stunde für diesen Mitarbeitertypen.

Employee Type Productivity Beschreibt, wie viele Function Points der Mitarbeiter pro Stunde schafft..

Event Ein Event ist ein Risiko, welches im Verlauf eines Spiels eintreten kann.

HS Herbstsemester.

Integration-Test Ein automatisierter Test der Software, welcher im Backend startet und bis auf die Datenbank hinunterläuft. Zuvor kann ein Testszenario in die Datenbank geladen werden. Nach dem Test kann die Datenbank hinsichtlich der erwarteten Änderungen inspiziert werden..

Issues Entspricht einem Arbeitspaket in GitLab..

Live-Tests Hiermit beziehen wir uns auf die 3 Testspiele in den Übungsstunden des HSR-Moduls "Projekt- und Qualitätsmanagement".

Milestone Entspricht einem Sprint in GitLab..

Mocha Ein JS-Unit-Test-Framework für Node.js.

Personendaten alle Angaben, die sich auf eine bestimmte oder bestimmbare Person beziehen (Beispiele: Geburtsdatum, Adresse, Name, Geschlecht, etc.) Screenreader-Unterstützung aktiviert. .

PmQm HSR-Modul Projekt- und Qualitätsmanagement, welches vom Betreuer dieser Arbeit (u.a. im HS2019) unterrichtet wurde..

Product Owner Mit Product Owner beziehen wir uns im Rahmen dieses Projekts auf den Betreuer der Arbeit.

Project Manager Task Beschreibt eine Tätigkeit des Projektleiters (Beispiel: Aufwand für Stakeholder-Kommunikation). Siehe auch unter Aufwand..

Project Manager Time Allocation Beschreibt, wie viel Zeit (in %) der Projektleiter für einen bestimmten Project Manager Task aufwendet.

Literatur

- [1] OECD, “Students, computers and learning.” <https://www.oecd-ilibrary.org/content/publication/9789264239555-en>, 2015.
- [2] A. Pham, “An education in child’s play,” *Los Angeles Times*, August 2011.
- [3] Globi-Verlag, “Globi cd-rom spiel 1,2,3 schwarze perle.” <https://www.globi.ch/globi/detail/globi-cdrom-spiel-123-schwarze-perle/502726/>.
- [4] N. Egger, M. Hermida, A. Cantieni, and D. Prasse, “Lernen und unterrichten in tabletklassen,” 2017.
- [5] Cesim Oy, “Compare cesim business simulations.” <https://www.cesim.com/simulations/compare-business-simulations>.
- [6] M. Barr, “Video games can develop graduate skills in higher education students: A randomised trial,” *Computers and Education*, vol. 113, pp. 86 – 97, 2017.
- [7] R. N. Landers, “Developing a theory of gamified learning: Linking serious games and gamification of learning,” *Simulation & Gaming*, vol. 45, no. 6, pp. 752–768, 2014.
- [8] N. Imlig-Iten, *Lernen mit digitalen Lernspielen im Unterricht*. PhD thesis, Universität Zürich, 2019.
- [9] R. Rosas, M. Nussbaum, P. Cumsille, V. Marianov, C. P. Flores, V. Grau, F. Lagos, P. Rodriguez, and M. Salinas, “Beyond nintendo: design and assessment of educational video games for first and second grade students,” *Computers and Education*, 2002.
- [10] STS Sauter Training and Simulation SA, “Simultrain.” <https://sts.ch/de/produkte/simulation/simultrain>.
- [11] “The project management game.” <http://thatpmpgame.com/>.
- [12] EDUardo Ltd., “Eduardo - the ultimate business simulation.” <https://education.com/>.
- [13] STS Sauter Training and Simulation SA, “User guide for the simultrain.” https://simultrain.swiss/doc/476_EN_User_manual_SimulTrain.pdf.

Abbildungsverzeichnis

1.	Screenshots Project Manager	v
2.	SWOT Analyse	vi
3.	Simultrain: Ansicht der Mitarbeiterzuweisung im Projekt	8
4.	Dashboard von Eduardo	11
5.	SWOT-Analyse unserer Projektmanagementsimulation.	13
6.	Liste der User Stories. Die grün markierten User Stories konnten im Rahmen der Arbeit umgesetzt werden.	18
7.	Liste der Use Cases. Die grün markierten Use Cases konnten im Rahmen der Arbeit umgesetzt werden. Die blau markierten Use Cases konnten im Rahmen der Arbeit teilweise umgesetzt werden.	19
8.	Die Abbildung zeigt die Abhängigkeiten der User Stories untereinander.	21
9.	Domain Model	37
10.	Die Gewichtung der nicht-funktionalen Anforderungen der Projektmitglieder, dem Resultat der Planungsphase und der effektiv implementierten Lösung.	39
11.	Datenmodell	48
12.	Schichtendiagramm	49
13.	Deployment-Diagramm	50
14.	Gui-Skizzen	51
15.	Black-Box-Testing gegen die Use cases.	54
16.	White-Box-Testing: Test-Coverage: Übersicht.	55
17.	White-Box-Testing: Test-Coverage: Controller-Schicht.	55
18.	White-Box-Testing: Test-Coverage: Service-Schicht.	56
19.	White-Box-Testing: Test-Coverage: Repository-Schicht.	56
20.	Testing der nicht-funktionalen Anforderungen.	57
21.	Die vier Stages der CI/CD Pipeline sind ersichtlich. Als letzten Schritt kann das Deployment manuell ausgelöst werden.	59
22.	Screenshot Filezilla	61
23.	Printscreen: Neues Spiel erfassen	62
24.	Printscreen: Gruppenzuteilung	63
25.	Landing page.	66
26.	Spielersicht: Einem Spiel beitreten.	66
27.	Spielersicht: Einer Gruppe beitreten.	67
28.	Spielleitersicht: Gruppenadministration.	67
29.	Spielleitersicht: Landing page.	68
30.	Spielleitersicht: Übersicht über die laufenden und beendeten Spiele.	69
31.	Spielleitersicht: Übersicht eines spezifisches Spiels.	70
32.	Spielleitersicht: Übersicht eines spezifisches Spiels (fullpage screenshot).	71
33.	Spielleitersicht: Erstellen eines neuen Spiels.	72
34.	Spielleitersicht: Neue Runde starten.	72
35.	Spielleitersicht: Erhalten eines Change Requests.	73

36.	Spielleitersicht: Notification oben rechts bei Erhalt eines neuen Change Requests.	73
37.	Spielleitersicht: Beispiel eines akzeptierten Change Requests.	74
38.	Gruppensicht: Statistikseite mit einem Change Request.	75
39.	Gruppensicht: Statistikseite mit Events.	76
40.	Gruppensicht: Startseite in einem laufenden Spiel. Von dieser Seite können die Management-Entscheidungen erreicht werden.	77
41.	Gruppensicht: Eingabe des Staffings für die aktuelle Runde.	77
42.	Gruppensicht: Eingabe Projektleiterzuteilung auf die verschiedenen Tasks.	78
43.	Gruppensicht: Eingabe der Investitionen für die Ausbildung und die Risikoreduktion.	79
44.	Spieler: Passwortwechsel mit interaktivem Feedback.	80
45.	Beispiel für die Ausgabe einer Fehlermeldung (oben links).	80
46.	Frontend: Source Lines of Code.	81
47.	Backend: Source Lines of Code total und Tests.	82
48.	Backend: Source Lines of Code - Details.	83
49.	Datenbank: Source Lines of Code.	84
50.	Dependency-Graph der Code-Struktur im Backend.	85
51.	Die wichtigsten Routes.	87
52.	Ausschreibungstext aus dem HSR-Arbeitsverwaltungstool.	89
53.	Projektcontrolling der Studienarbeit.	95
54.	Projektcontrolling der Studienarbeit.	97
55.	Eingetretene Risiken.	99

Tabellenverzeichnis

- 1. Changelog ii
- 2. Vergleich Funktionsumfang 10
- 3. Vergleich Kosten 12

A. E-Mail Verkehr

A.1. STS

RE: Informationsanfrage für Simultrain

gCH - STS Office <[REDACTED]@sts.ch>

Mi 11.12.2019 17:13

An: Iten Martin <[REDACTED]@hsr.ch>;

Guten Tag Herr Iten

Gerne beantworten wir Ihre Fragen.

- Gilt die Lizenzierung zeitlich unbegrenzt oder muss diese regelmässig erneuert werden? Eine Lizenz (für 3-4 Teilnehmer) ist für eine einmalige Simulation von Anfang bis Ende gültig. Danach ist die Lizenz 'abgelaufen' aber noch 3 Monate zur Ansicht der Resultate verfügbar. Die Lizenz kann innerhalb eines Jahres nach Kaufdatum benutzt werden.

- Kann der Trainer beliebige Spieler einer Simulation zuweisen oder muss für neue Spieler eine neue Lizenz erworben werden? Es muss für jeder neue Simulation eine neue Lizenz erworben werden.

Gerne stehen wir Ihnen für Fragen und weitere Informationen zur Verfügung und grüssen Sie freundlich aus Lausanne

Best regards / Freundliche Grüsse / Bien cordialement



[REDACTED] Account Manager

E : office@sts.ch **T** : +41 21 510 11 50 **F** : +41 351 86 83

STS SA - Av. de la Gare 10 - 1003 Lausanne - Switzerland **W** : [www.sts.ch]www.sts.ch



 Follow us on [LinkedIn](#) !

PART OF THE DEMOS GROUP
www.global.demosgroup.com

Gerne senden wir Ihnen im Anhang unseren Produktflyer zu SimulTrain, sowie folgende allgemeine Informationen: <https://sts.ch/de/produkte/simulation/simultrain>

SimulTrain ist ein Simulator, der in Ausbildungsworkshops eingesetzt wird. Er ist nicht für das Selbststudium geeignet.

Die Ausbildung erfolgt in Kleingruppen von 3-4 Personen, die gemeinsam die Rolle des Projektmanagers übernehmen. Vor dem Simulator wird die Gruppe mit vielen Situationen konfrontiert, die eine schnelle Entscheidung erfordern. Die Teilnehmer lernen also, als Team zusammenzuarbeiten und zu entscheiden.

Der Simulator simuliert realitätsnah den Ablauf eines mittelgrossen Projektes. Die Gruppe wird dabei mit unzähligen Situationen konfrontiert, die rasche Entscheidungen verlangen; Kosten und Termine müssen eingehalten werden - ohne die menschlichen Faktoren zu vergessen.

Am Ende der Simulation ziehen die Teilnehmer gruppenweise Bilanz aus den gemachten Fehlern und notieren sich die gelernten Lektionen. Der Kursleiter greift während der Simulation nur selten ein. Er wird vor allem in der abschliessenden Debriefing-Phase aktiv.

Weitere Informationen und Videos, auch für den Trainer, sowie eine Demoversion finden Sie auch unter www.simultrain.swiss

Einsatz von SimulTrain:

SimulTrain ist das ein interaktives Tool, das sich perfekt in ein Präsenzunterricht integrieren lässt. Eine komplette Simulation dauert ca 7 – 8 Stunden (inkl Diskussionen und Debriefing Phasen). Bei Bedarf können einige Parameter so eingestellt werden, dass die Simulation schneller verläuft.

Es ist unentbehrlich, dass der Lehrer / Trainer die Simulation gut kennt, damit er alle Parameter richtig einstellen kann und auch in der Debriefing Phase die richtige Auswertung macht.

Für Trainer, die SimulTrain® einsetzen möchten, organisiert STS kostenlose, online Einführungen „Entdecken Sie SimulTrain®“ von 1-2 Stunden. Am Ende dieser Einführung werden die Teilnehmer die Hauptfunktionalitäten, sowie die wichtigsten pädagogischen Aspekte von SimulTrain® kennen. Mehr Informationen und den Link zum Anmelden finden Sie hier: <http://sts.ch/en/events/products-presentation>

Preise:

Eine Lizenz SimulTrain kostet CHF 905.- (für max 4 Personen).

Öffentlichen Schulen und Hochschulen erhalten 75% Rabatt. Dieser Rabatt wird unter der doppelten Bedingung gewährt, dass die Studenten

für mindestens ein Jahr in der Erstausbildung immatrikuliert sind, und dass das Budget der Schule, staatlich kontrolliert wird.
Im Fall einer Weiterbildung (oder Executive Ausbildung) beträgt der Rabatt 40%.

Bei grösseren Bestellungen ab 11 Lizenzen gewähren wir Mengenrabatte.
Für 50 User pro halb Jahr benötigen Sie 13 Lizenzen. Falls Sie diese in einer einmaligen Bestellung erwerben, gewähren wir 5% Mengenrabatt.
Ab 26 Lizenzen (entsprechend 104 User) gewähren wir 10% Mengenrabatt.

Bei Fragen oder um das Konzept im Detail zu besprechen, steht Ihnen unser CEO Patrick Lehmann gerne auch telefonisch zur Verfügung.

Freudliche Grüsse



Account Manager

E : office@sts.ch **T** : +41 21 510 11 50 **F** : +41 351 86 83

STS SA - Av. de la Gare 10 - 1003 Lausanne - Switzerland **W** : [www.sts.ch]www.sts.ch



[Follow us on LinkedIn!](#)

PART OF THE DEMOS GROUP

www.global.demosgroup.com

Year-end Holiday Season – special opening hours

During the Holiday season, our offices will be **open on 23 and 30 December as well as 3 January**. We will be happy to respond to your emails and calls during our office hours (8:30 to 17:00 CET).

The offices are closed on the other days between 24 Dec. and 6 Jan. Please consider these opening days for your orders.

For urgent technical issues, please call +41 21 510 11 64.

A.2. EDUardo

EDUardo Business Simulation inquiry

[REDACTED]@edu-simulation.com>

Do 12.12.2019 19:01

An: Iten Martin <[REDACTED]@hsr.ch>;

2 Anlagen (2 MB)

EDUardo_Business_Simulation_introduction.pdf; EDUardo_for_Business_Schools_intro.pdf;

Dear Martin Iten,

Thank you for your interest in using EDUardo Business Simulation in your institution. As you requested, I hereby send you the previously discussed information via email.

Pricing:

The pricing of the EDUardo business simulation depends on several factors. We have multiple simulation modules available, which can all be customized for your specific needs if necessary. Pricing can also be affected by the number of end-users in question (20 to 30 in your case).

EDUardo's price is €35 per student, per simulation module, per semester. We can provide additional discounts for 100 students or more.

Customization options:

EDUardo simulation was built with customization in mind, therefore every knowledge element can be replaced or extended. If you want to upload some quizzes or e-learning materials like videos or text-based knowledge elements, we can arrange that without problems. Also, new customers receive their first customizations free of charge.

Apart from the Knowledge Base of the modules, the simulations themselves can be extended with in-game videos or messages upon request.

Your account:

I have created an account for you so that you can try out all the available simulation engines, and see for yourself which ones would suit your purposes the most. Please find the login details below:

Login URL: <https://app.edu-simulation.com/>

Email address: [REDACTED]@hsr.ch

Password: [REDACTED]

B. Projektplan

Projektplan „Project Manager“

HS19 Hochschule Rapperswil

Marco Keller Martin Iten

21. Oktober 2019

Version 1

Inhaltsverzeichnis

1	Änderungsverlauf	3
2	Einführung	4
2.1	Zweck	4
2.2	Gültigkeitsbereich	4
3	Projektübersicht	5
3.1	Zweck	5
3.2	Aufgabenstellung	5
3.3	Ziel	5
3.4	Lieferumfang	6
4	Projektorganisation	7
4.1	Übersicht	7
4.2	Externe Schnittstellen	7
5	Managementabläufe	8
5.1	Iteration	8
5.2	Kostenvoranschlag	8
5.3	Zeitliche Planung	8
5.4	Meilensteine	9
5.5	Besprechungen	9
5.6	Review-Termine	9
6	Risikomanagement	11
6.1	Risiken	11
6.2	Umgang mit Risiken	11

7	Infrastruktur	12
7.1	Gitlab	12
7.1.1	CI/CD Pipeline	12
8	Qualitätsmassnahmen	13
8.1	Definition of Done („Definition Fertigstellungsgrad der Arbeitspakete“) .	13
8.2	Generelle Qualitätsmassnahmen	13
8.3	Code-Qualität	14
8.4	Testing	15

1 Änderungsverlauf

Datum	Version	Änderung	Autor
2019-03-01	1	Initiale Version des Dokumentes	Marco Keller, Martin Iten

Tabelle 1: Changelog

2 Einführung

2.1 Zweck

Dieser Projektplan bildet die Grundlage für die Studienarbeit „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 Projektmanagementplanspiel, welches im Rahmen der Studienarbeit an der HSR als software-basierte Lösung erstellt wird. Neben dem praktischen Teil werden auch eine theoretische sowie betriebswirtschaftliche Fragestellung untersucht.

3.1 Zweck

„Project Manager“ soll den im HSR-Modul „Projekt- und Qualitätsmanagement“ erlernten Stoff spielerisch vertiefen.

3.2 Aufgabenstellung

1. Gibt es in der Literatur oder bei anderen Instituten Hinweise darauf, um wieviel effizienter spielerisches Lernen ist?
2. Können die Anforderungen des Dozenten in einer Software umgesetzt werden? Begründung, wenn Anforderungen nicht wie gefordert umgesetzt werden konnten.
3. Könnte die HSR diese Software in Form einer Trainingsdienstleistung im freien Markt anbieten, um Softwareunternehmen im Bereich Projektmanagement fitter zu machen?

Die komplette [Aufgabenstellung ist im Google Drive](#)¹ einsehbar.

3.3 Ziel

Wir setzen uns als Projektteam folgende Ziele:

1. Bestehen des Moduls Studienarbeit.
2. Untersuchung und Beantwortung der theoretischen Fragestellung.
3. Untersuchung und Beantwortung der betriebswirtschaftlichen Fragestellung.
4. Konzeptioneller Teil: saubere Anforderungsanalyse und Architekturdesign, Vorgehensmethode definieren und während der gesamten Projektlaufzeit konsequent befolgen, alle Dokumentationsrichtlinien der HSR einhalten.
5. Praktischer Teil: lauffähige software-basierte Lösung erarbeiten.

¹<https://drive.google.com/open?id=1XlxRk1cfGvM6ONUQZ07wGhcGMadzghX9>

3.4 Lieferumfang

Am Ende des Projekts werden folgende Arbeitsprodukte abgegeben:

- Source Code
- Projektdokumentation gemäss HSR-Richtlinien²

²Siehe dazu das Dokument [Anleitung: Dokumentation Studien- und Bachelorarbeiten](#). Dieser Link ist nur innerhalb des HSR-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. Die einzelnen Mitglieder unterstützen sich auch interdisziplinär. Dazu treffen wir uns zwei bis dreimal pro Woche an der HSR.

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

4.2 Externe Schnittstellen

- Projektbetreuer: Claudio Fuchs, claudio.fuchs@hsr.ch
- Serverbetreuung: IT Service Desk HSR, i-support@hsr.ch

5 Managementabläufe

Wir setzen für dieses Projekt auf Scrum + Unified Process³. Dies erlaubt eine agile Entwicklung unter Einbezug einiger Aspekte von Unified Process. Wir haben uns gegen ein vollständig agiles Vorgehen entschieden, da die Projektlaufzeit mit 14 Wochen kurz ist. Zudem ist die Art und der Umfang der Dokumentation von der HSR vorgegeben und bietet wenig Spielraum. Bei einem vollständig agilen Vorgehen befürchten wir, den Fokus zu verlieren. Entsprechend setzen wir uns mit den Phasen Inception, Elaboration, Construction und Transition aus Unified Process 4 grobe Meilensteine.⁴

5.1 Iteration

Für dieses Projekt setzen wir auf zweiwöchige Sprints. An den zweiwöchentlichen Meetings wird die letzte Iteration abgeschlossen, besprochen und ausgewertet.

5.2 Kostenvoranschlag

Für das Projekt stehen insgesamt 14 Wochen zur Verfügung. Jeder Student erhält 8 European Credit Transfer System-Punkte (ECTS) für die Studienarbeit. 1 ECTS entspricht einem Arbeitsaufwand von 30 Stunden⁵ Total stehen somit 480 Arbeitsstunden zur Verfügung:

$$2 \text{ Projektmitglieder} \cdot 30 \text{ Stunden} \cdot 8 \text{ ECTS} = 480 \text{ Arbeitsstunden}$$

Ein Sprint dauert jeweils zwei Wochen. Pro Sprint stehen somit 68.57 Arbeitsstunden zur Verfügung:

$$2 \text{ Projektmitglieder} \cdot 17.14 \text{ Stunden} \cdot 2 \text{ Wochen pro Sprint} = 68.57 \text{ Arbeitsstunden}$$

In diesen 68.57 Arbeitsstunden ist jeweils die Zeit für die Teamsitzung und Risikoabdeckung bereits mit eingerechnet⁶. Total stehen somit 480 Arbeitsstunden zur Verfügung. In diesen 480 Arbeitsstunden ist jeweils die Zeit für die Teamsitzung und Risikoabdeckung bereits mit eingerechnet.

5.3 Zeitliche Planung

Die Planung wird laufend aktualisiert und den gegebenen Umständen angepasst. Der aktuelle Zeitplan ist [hier](#)⁷ erreichbar.

³An der HSR unter Scrum+ bekannt

⁴Das Vorgehen nach Wasserfall haben wir verworfen, da wir zu Beginn den exakten Funktionsumfang und die einzusetzende Technologie noch nicht exakt definiert hatten.

⁵siehe HSR-Studienführer Informatik, https://www.hsr.ch/fileadmin/user_upload/1_studium/1.1_bachelor/informatik/Studienfuehrer_Informatik_2016_WEB_URL.pdf.

⁶Risiken werden nach Eintrittswahrscheinlichkeit sowie max. erwarteter Schaden gewichtet. Daraus kann der gewichtete Schaden abgeleitet werden. Dieser wird in der Zeitplanung berücksichtigt.

⁷https://docs.google.com/spreadsheets/d/1uZ_kEq7Ugfo4GSLp9czpcPMX_gKOLgP_roistcYGA_o

Die Zeitschätzung erfolgt im Plenum. Jedes Projektmitglied gibt eine Zeitschätzung für das Arbeitspaket ab. Alle Schätzungen werden gemittelt und als Zeitschätzung verwendet. Alternativ erfolgt die Zeitschätzung mittels eines „educated guess“ durch ein einzelnes, erfahrenes Projektmitglied. Falls ein Paket risikobehaftet ist, wird die Zeitschätzung um 50% erhöht.

Die Zeitschätzung der Projektmitgliedern für die User Stories ist [hier](#)⁸ einsehbar.

Die Zeitauswertung inklusive einem Controlling-Cockpit ist [hier](#)⁹ zu finden.

Die Liste mit allen Arbeitspaketen des Projekts ist auf [GitLab](#)¹⁰ einsehbar.

5.4 Meilensteine

Aufgrund unserer agilen Arbeitsweise sowie der kurzen Projektlaufzeit verzichten wir auf Meilensteine. Wir orientieren uns stattdessen an den vier Phasen Inception, Elaboration, Construction und Transition.

5.5 Besprechungen

An der HSR führen wir im Team regelmässige Besprechungen durch. Dies sind vor allem unsere Daily Besprechungen am Montag und Mittwoch, sowie die Sprint Reviews am Ende eines Sprints.

Sämtliche Termine sind in unserem [Zeitplan](#)¹¹ festgehalten. Ebenso führen wir für jede Besprechung ein [Protokoll](#)¹².

5.6 Review-Termine

Im zweiwöchentlichen Turnus treffen wir uns mit unserem Projektbetreuer Claudio Fuchs an der HSR oder falls kein Termin möglich ist per Skype. Diese Termine dienen dazu den Stand der Arbeiten zu präsentieren und die Prioritäten den nächsten Sprint zu definieren.

Folgende Termine wurden hierzu festgelegt:

- 19.09.2019 - 12:10 Uhr
- 03.10.2019 - 11:50 Uhr
- 10.10.2019 - 11:40 Uhr
- 24.10.2019 - 12:10 Uhr
- 06.11.2019 - 14:00 Uhr (Skype)

⁸https://docs.google.com/spreadsheets/d/1mm13X81N9_p3myCDcdZFQUQNhqghTPDKnF24FN9A00

⁹<https://docs.google.com/spreadsheets/d/1501KZfXeqGxtyebH1VM7NRMu5fkeEXf9hb7nzleprX4>

¹⁰<https://gitlab.dev.ifs.hsr.ch/sa-keit/project-manager/issues>

¹¹https://docs.google.com/spreadsheets/d/iuZ_kEq7Ugfo4GSLp9czpcPMX_gKOLgP_roistcYGA_o

¹²https://drive.google.com/drive/folders/1_-MQJBeMxk94vi_DMJFBc_86HU_I8wEP

- 21.11.2019 - 12:10 Uhr
- 05.12.2019 - 12:10 Uhr

6 Risikomanagement

6.1 Risiken

Das Risikomanagement ist [hier](#)¹³ zu erreichen.

6.2 Umgang mit Risiken

- Strategiewahl: vor allem reduzieren und vermeiden; wenig selbst tragen; kein Abwälzen (Versichern)
- Pauschal 50% mehr Zeit einplanen bei kritischen Arbeitspaketen.
- Gutes Verhältnis und kurze Kommunikationswege zum Betreuer der Studienarbeit halten, um bei Eintritt eines Risikos schnell um Rat fragen zu können.
- Die Risikoliste wird 1x pro Sprint überprüft und (wenn nötig) aktualisiert.
- Risikobehaftete Arbeitspakete werden mit einem Risikolabel versehen.
- Ob ein Ticket das Risikolabel bekommt, wird im Plenum besprochen.

¹³<https://docs.google.com/spreadsheets/d/1zMzaLQaSfh4JvKQWg1oSP1Nh3QYMwbIKikd54J5Pa18>

7 Infrastruktur

- Gitlab
 - Gitlab CI für Versionsverwaltung und CI/CD
- Google Drive
 - Google Docs
 - Google Sheets
- Automatic Build (Build Server) und Continuous Integration: Gitlab

7.1 Gitlab

Gitlab verwenden wir als Hosting-Plattform für den Code und die LaTeX Dokumente, zur Zeiterfassung sowie zur Verwaltung der Arbeitspakete.

7.1.1 CI/CD Pipeline

Wir haben uns für Gitlab CI/CD entschieden, damit wir möglichst wenig unterschiedliche Tools verwenden müssen. Diese Plattform hat sich bereits im Engineering-Projekt 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 beschreibt, wie wir die Arbeitsqualität über die gesamte Laufzeit des Projekts sicherstellen möchten. Dieser Abschnitt beschreibt **nicht** die Qualitätsanforderungen an die Software, denn diese werden in den nicht-funktionalen Anforderungen beschrieben.

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

Für Arbeitspakete mit dem Label „#Development“:

- Funktionalität ist implementiert
- Die Dokumentation ist auf dem aktuellen Stand
- Style-Guidelines eingehalten

Für Arbeitspakete mit dem Label „#Review“ und „#Testing“:

- Alle Unit Tests bestanden
- Alle Integrations Tests bestanden
- GUI Test bestanden
- Review durchgeführt

Arbeitspakete, welche **keines** der Label „#Development“, „#Review“ und „#Testing“ enthalten, werden vor dem Schliessen des Arbeitspaketes gemeinsam besprochen und erst geschlossen, wenn alle Projektmitglieder mit der Schliessung einverstanden sind.

8.2 Generelle Qualitätsmassnahmen

- Regelmässige Besprechungen (innerhalb des Teams als auch mit dem Betreuer) stellen sicher, dass die aktuellen Fortschritte und Rückschläge zeitnah ausgetauscht werden.
- Wir richten einen separaten digitalen Kommunikationskanal für dieses Projekt ein (Slack, Telegram oder ähnlich). Alle Projektmitglieder haben darauf jederzeit Zugriff.
- Wir nehmen uns Zeit für die Sprint Retrospektive.
- Für jeden Sprint definieren wir ein Sprint Goal. Das Sprint Goal halten wir in GitLab prominent fest.

- Wir erstellen eine Übersichtsseite in GitLab ([Wiki](#)¹⁴) mit den Direktlinks zu allen wichtigen Dokumenten. Das Wiki wird fortlaufend aktualisiert.
- Alle Dokumente liegen zentral entweder im Google Drive oder auf GitLab. Alle Projektmitglieder haben jederzeit Zugriff auf alle Dokumente.
- Beim Erstellen der Arbeitspakete in GitLab nutzen wir die Möglichkeit, digitale Checklisten in der Arbeitspaketbeschreibung („Description“) zu erstellen, um den Fortschritt des Arbeitspakets festzuhalten.
- Bei Arbeitspaketen mit dem Label „#Development“, „#Review“ oder „#Testing“ erfassen wir in der Beschreibung die Definition of Done als Checkliste.

8.3 Code-Qualität

- Wir führen Unittests und Integrationstests durch
 - Wir streben eine Test-Coverage von 65% an.¹⁵
- Wir versuchen, Kommentare im Code zu vermeiden.
 - Wenn wir uns gezwungen sehen, Kommentare im Code zu machen, werten wir dies als Hinweis auf schlechten Code und besprechen diesen Code in einem Code Review.
- Wir orientieren uns am Github-Flow¹⁶.
- Wir verwenden Merge-Request-Templates
- Wir verwenden ein 4-Augen-Prinzip für Merges
 - Ein Projektmitglied erstellt einen neuen Merge Request
 - Ein anderes Projektmitglied macht einen Review und hinterlässt gegebenenfalls Kommentare zur Überarbeitung in GitLab.
 - Der Ersteller des Merge Request merged **nie** seinen eigenen Merge Request (insbesondere für staging/release und master-branch).
 - Der Merge-Button wird von einem Teammitglied als Zeichen des Reviews gedrückt.
- Wir wenden Continuous Integration an.
 - Wir halten die einzelnen Commits klein.
 - Nach jedem „git push“ wird das CI getriggert und die Software im Repo überprüft (z.B. Linter, Tests, Metriken, etc.).

¹⁴<https://gitlab.dev.ifs.hsr.ch/sa-keit/project-manager/wikis/home>

¹⁵Die Unittests werden mit Mocha durchgeführt.

¹⁶<https://guides.github.com/introduction/flow/>

- Wir verwenden einen Code-Styleguide
 - AirBnB Javascript Style (eslint mit airbnb).

8.4 Testing

Im Projektverlauf geplante Tests:

- Unit-Tests und Integration-Tests: siehe 8.3
- Performance-Test: Prüfung, ob die nicht-funktionalen Anforderungen eingehalten werden
- Usability-Test: Live Testing der Applikation im PmQm-Unterricht.
- System-Test (Blackbox-Testing): koordiniert mit allen Projektmitgliedern (inklusive Product Owner) durchzuführen.

Zum Performance-Test, Usability-Test und System-Test erstellen wir jeweils eine Testspezifikation und ein Testprotokoll.

C. Nicht-funktionale Anforderungen: Grafik Rohdaten

Rohdaten der Priorisierung der nicht-funktionalen Anforderungen:

Kategorie	Skala von 0 bis 4 (5 mögliche Werte)	Subkategorien			Kategorien			
		Subkategorie	Marco Keller	Martin Iten	Resultat Planungsphase	Marco Keller	Martin Iten	Resultat Planungsphase
Functional Stability	Functional Completeness	2	4	1	2.0	3.7	1.7	1.3
	Functional Correctness	4	4	4				
	Functional Appropriateness	0	3	0				
Performance efficiency	Time-Behavior	1	2	2	0.7	1.3	1.0	0.7
	Resource Utilization	0	1	0				
	Capacity	1	1	1				
Compatibility	Co-existence	0	0	0	0.0	0.0	0.0	0.0
	Interoperability	0	0	0				
Usability	Appropriateness recognizability ("Obvious Applicability")	0	2	1	1.8	2.7	1.7	1.2
	Learnability	0	2	1				
	Operability ("Intuitiveness")	4	4	3				
	User error protection	4	2	3				
	User interface aesthetics ("Aesthetics", "Look & Feel")	2	4	2				
	Accessibility	1	2	0				
Reliability	Maturity	2	3	0	1.5	2.5	0.8	0.6
	Availability	1	4	1				
	Fault Tolerance	1	2	0				
	Recoverability	2	1	2				
Security	Confidentiality	2	1	1	0.8	1.6	0.6	0.6
	Integrity	1	3	2				
	Non-repudiation	0	0	0				
	Accountability	0	2	0				
	Authenticity	1	2	0				
Maintainability	Modularity	4	2	2	2.4	1.8	1.4	1.3
	Reusability	2	2	0				
	Analyzeability	2	0	0				
	Modifiability	3	3	4				
	Testability	1	2	1				
Portability	Adaptability	3	1	4	2.7	2.0	2.3	2.3
	Installability	3	3	3				
	Replaceability	2	2	0				

D. Einschätzung der Metrik Lines of Code

Einstufung DLOC (Fortsetzung)

Die Angabe (D)LOC sagt gar nichts über die Qualität aus (ausser wenn pro Person aussergewöhnlich viele Zeilen Code erzeugt wurden, dann liegt der Verdacht nahe, dass entweder viel Copy/Paste gemacht wurde oder dann die Qualität zu Gunsten der Menge stark vernachlässigt wurde).

Pro Programmierer pro Monat sind etwa folgende Zahlen zu erwarten:

- 80-150 LOC bei schwierigen Echtzeit-Projekten (Zahl v. IBM, Space Shuttle)
- 300-500 LOC im Schnitt mit einem guten, nicht zu grossen Team
- bis ca. 1000 LOC nur mit kleinen Spitzenteams
- über 1500 LOC ist unglaublich (Pfusch, viel Copy/Paste, generiert)

Bei der **Messung der LOC pro Person und Monat** nimmt man die Leistung des ganzen Entwicklungsteams, inkl. Requirements, technische Leitung, Grafiker, Tester über der ganzen Projektdauer, von Kickoff bis Abnahme.