



Fachhochschule Köln

Cologne University of Applied Sciences

## Meilenstein 2 – Konzept

im Rahmen der Veranstaltung

“Entwicklung interaktiver Systeme”

im SS 2015

## **Learn2Quiz!**

von Bruno José Carvalho Gonzaga und Christian Ries

### **Betreuer**

Prof. Dr. Gerhard Hartmann

Prof. Dr. Kristian Fischer

Sheree Saßmannshausen

Franz Jaspers

Gummersbach, 29.06.15

# Inhaltsverzeichnis

<b>1 Exposé.....</b>	<b>4</b>
1.1 Problemstellung.....	4
1.2 Ziel.....	4
1.3 Lösung.....	4
1.4 Verteiltheit.....	4
<b>2 Zielhierarchie.....</b>	<b>5</b>
2.1 Strategische Ziele.....	5
2.2 Taktische Ziele.....	5
2.3 Operative Ziele.....	5
<b>3 Marktrecherche.....</b>	<b>6</b>
3.1 GoConqr.com.....	6
3.1.1 Karteikarten.....	6
3.1.2 Quiz.....	7
3.1.3 Zusammenfassung.....	7
3.2 Anki.com.....	8
3.2.1 Lernen & Abfragen.....	8
3.2.2 AnkiDroid.....	8
3.2.3 Besondere Merkmale.....	8
3.2.4 Zusammenfassung.....	9
3.3 Fazit.....	9
<b>4 Alleinstellungsmerkmale.....</b>	<b>10</b>
<b>5 Methodischer Rahmen.....</b>	<b>11</b>
5.1 Abwägung der Vorgehensmodelle.....	11
5.2 Methoden und Techniken.....	11
5.2.1 Discount Usability Engineering.....	11
5.2.2 Usability Engineering Lifecycle.....	12
5.2.3 Scenario Based Usability Engineering.....	12
5.3 Fazit.....	12
5.4 Vorgehensmodell.....	13
<b>6 WBA-Techniken.....</b>	<b>15</b>
6.1 Kommunikationsabläufe.....	15
6.1.1 Simple Interaktionen .....	15
6.1.2 Komplexe Interaktionen.....	15
6.2 Systemarchitektur.....	18
6.2.1 Verteilungsparadigmen.....	18
6.2.2 Systemkomponenten.....	18

6.2.3 Wahl der Middleware.....	19
6.2.4 Wahl der Service Architektur.....	19
6.2.5 Nachrichtenorientierte Middleware.....	20
<b>7 Geschäftsmodell.....</b>	<b>20</b>
7.1 Freemium.....	21
7.2 Einmalzahlung.....	21
7.3 Abonnements.....	21
<b>8 Risiken.....</b>	<b>21</b>
8.1 Geschäftsrisiken.....	21
8.2 Technische Risiken.....	22
8.3 Implementierungsrisiken.....	22
<b>9 Proof-of-Concepts.....</b>	<b>22</b>
9.1 Synchrone Kommunikation zwischen Client und Server & Speicherung der Daten in einer Datenbank.....	22
9.2 Asynchrone Nachrichten von Server zu Client & Client zu Client.....	23
9.3 Demo-Desktopanwendung mit libGDX (Java Game Library).....	23
9.4 Matchmaking (Einladungen an Benutzer senden).....	24

## Abbildungsverzeichnis

Abbildung 1: Ablauf des Gestaltungsprozesses.....	14
Abbildung 2: Simple Interaktion zwischen Spieler & System.....	16
Abbildung 3: Grundlegender Spielablauf zwischen Spieler & System.....	17
Abbildung 4: Kooperation zwischen Spielern während eines Spiels.....	18
Abbildung 5: Die Diskussionsrunde mit zentraler Einheit zwischen den Spielern.....	18

# **1 Exposé**

## **1.1 Problemstellung**

Die meisten Studenten und Schüler sind eher faul wenn es darum geht für anstehende Klausuren zu lernen und haben oft nicht die ausreichende Disziplin sich im Alleingang auf eine Klausur vorzubereiten. Schlechte Ergebnisse in den Klausuren demotivieren und fördern keineswegs den Lernprozess eines Studenten.

## **1.2 Ziel**

Das Ziel ist es, Studenten und Schülern möglichst strukturiert und ausreichend auf eine Klausur vorzubereiten. Dabei soll die intrinsische Motivation des Studenten durch eigene Erfolgserlebnisse gestärkt werden. Durch geeignete Belohnungsverfahren kann eine extrinsische Motivation geweckt werden.

Allgemein kann Motivation durch kollaboratives Lernen herbeigeführt werden. Der gegenseitige Austausch mit an anderen Studenten und Schülern fördert den eigenen Lernprozess und macht das Lernen interessanter.

## **1.3 Lösung**

Ein verteiltes System, welches Studenten und Schülern die Möglichkeit gibt gemeinsame Karteikarten-Pools anzulegen. Ein Pool von Karteikarten steht stellvertretend für ein bestimmtes Themengebiet.

Ein Nutzer kann diese Pools in Form von einem Quiz lernen. Dabei spielen die Nutzer innerhalb eines Teams mit und gegeneinander. Damit innerhalb des Teams der Fähigkeitslevel aller Spieler ungefähr auf dem selben Niveau liegt, greift das System auf ein Punktesystem von jedem einzelnen Spieler zu und bildet darauf basierend ein Team mit möglichst gleichstarken Spielern.

## **1.4 Verteiltheit**

Die Verteiltheit besteht darin auf den Clients den Fähigkeitslevel eines Nutzers anhand seiner Quiz-Ergebnisse zu ermitteln um dadurch automatisiert einen passenden Gegner für ein Quiz-Duell zu finden.

## **2 Zielhierarchie**

Die zu verfolgenden Ziele werden, aus der im Exposé genannten Problemstellung, abgeleitet. Die Ziele unterteilen sich in strategische (langfristige), taktische (mittelfristige), operative (kurzfristige) Ziele.

### **2.1 Strategische Ziele**

Das Ziel des Projekts ist die Entwicklung einer Anwendung, die es Schülern und Studenten ermöglicht, gemeinsam als Team für ein bestimmtes Themengebiet zu lernen. Das Lernen erfolgt in Form von digitalen Karteikarten die es innerhalb und mit Hilfe des Teams zu beantworten gilt.

### **2.2 Taktische Ziele**

Studenten sollen bestmöglich auf eine Klausur vorbereitet werden, indem die zu lernenden Informationen im Langzeitgedächtnis abgespeichert wird. Durch die Zusammenarbeit und das Vergleichen der eigenen Lernergebnisse mit anderen Studenten, sollen die Benutzer dazu angespornt weiter zu lernen. Ein geeignetes Punktesystem soll den Fähigkeitsgrad des Benutzers reflektieren. Eine einfache Nutzung soll es ermöglichen die angestrebten Ziele zu erreichen.

### **2.3 Operative Ziele**

Das Realisieren folgender Systemkomponenten wird als kurzfristiges Ziel des Projektes angesehen:

- Entwickeln des Spielmodus (Piktogramme, Benutzerführung)
- Spieldynamik (Regelwerk für ein Quiz-Duell)
- Ermitteln des Fähigkeitslevels eines Spielers
- Teamermittlung anhand des Fähigkeitslevel

Es muss ein verteiltes System entwickelt werden, welches das Konzept des systematischen Lernens mit Karteikarten implementiert. Um die Anforderungen der Nutzer zu erfüllen und das System gebrauchstauglich zu gestalten, muss ein menschenzentriertes Vorgehensmodell angewandt werden.

Der Nutzungskontext muss genau definiert werden; d.h. eine umfassende Domainenrecherche anstellen, mögliche Statistiken und wissenschaftliche Arbeiten heranziehen, eine genau Marktanalyse durchführen um eventuelle Ansätze für ein Alleinstellungsmerkmal zu finden, sowie eine ausführliche Stakeholderanalyse mit anschließender Benutzermodellierung. Die Nutzungsmotivation und die Anforderungen der Benutzer müssen ermittelt und beschrieben werden.

Zusätzlich zu den MCI-Abwägungen müssen die WBA-Techniken definiert werden. Es sollen mehrere Paradigmen der Kommunikation verglichen werden und durch Abwägung der einzelnen Technologien, eine passende Lösung gefunden werden.

Durch die Identifizierung und Beschreibung potentieller Risiken, sowie das Entwickeln von Proof-of-Concepts sollen die möglichen Projektrisiken gemindert werden. Zur Evaluation der Gestaltungslösung werden verschiedene Techniken wie z.B. Simplified thinking aloud und Cognitive Walkthrough sowie der ISO Standard 9241-110 Grundsätze der Dialoggestaltung herangezogen.

### 3 Marktrecherche

In der Marktrecherche werden bestehende Konkurrenzprodukte betrachtet und analysiert um bestenfalls daraus ein Alleinstellungsmerkmal für die eigene Anwendung ableiten zu können. Die vorhandenen Systeme sollen dabei den gleichen Problemraum abdecken und die selben Zielvoraussetzungen haben. Nach der Analyse sind die Vor und Nachteile gegenüber Learn2gether aufzuzeigen. Die Recherche umfasst neben Webanwendungen auch Applikationen aus dem Google Play Store und Apple App Store. Das Augenmerk richtet sich primär an Anwendungen, die zum Ziel haben, Nutzer anhand von digitalen Lernkarteikarten und Quiz, zum lernen zu motivieren. Da es eine sehr große Anzahl an bestehenden Lösungen gibt, wurden die zwei wichtigsten rausgefiltert und genauer unter die Lupe genommen.

#### 3.1 GoConqr.com

GoConqr ist momentan eine der umfangreichsten plattformübergreifende Lernplattform die der Markt zu bieten hat. Die Anwendung steht neben Browsern auch für sämtliche mobile Endgeräte zur Verfügung und ist auf der jeweiligen Website als auch im Google Play Store und Apple App Store erhältlich. Zu der Zielgruppe der Plattform gehören Schüler, Studenten und Lehrer. Mit GoConqr lassen sich ganz einfach Lernmaterialien wie zum Beispiel Notizen, Mindmaps, Lern-Quizze oder Karteikarten erstellen und mit anderen teilen. Es gibt auch eine Gruppen-Funktion mit der sich Lerngruppen bilden lassen in der dann alle Teilnehmer gemeinsam an Lernmaterialien arbeiten können. In diesen Lerngruppen können sich zum Beispiel auch Lehrer mit ihren Schülern austauschen, was die Möglichkeit mit sich bringt, Lerninhalte interaktiver gestalten zu können.

GoConqr gibt es in der Basic Variante die kostenlos nutzbar ist oder auch in der Premium und Premium+ Variante. Für Premium werden monatlich 3,95 € oder jährlich 19,95 € verlangt. Premium+ hingegen ist monatlich für 4,95 € zu haben oder ein ganzes Jahr für 29,95 €. Neben der Werbefreiheit die Premium User genießen, können diese deutlich mehr öffentliche und private Ressourcen erstellen. Zur Verdeutlichung: Die Erstellung eines Quiz oder eines Karteikarten Sets verbraucht dabei jeweils eine Ressource. Das gilt auch für das erstellen einer Mindmap und Notiz. Nutzer der Basic Version müssen zudem ganz auf die privaten Ressourcen verzichten. Es lassen sich hier also nur öffentliche Ressourcen erstellen und das heißt, das jeder der die Anwendung auch benutzt, Zugriff auf alle Lernmaterialien der Basic Nutzer hat. Auch die Anzahl der Fächer die sich erstellen lassen um mit passenden Lerninhalt zu füllen sind limitiert.

Im nachfolgenden werden nur zwei Teilfunktionen näher beschrieben, da nur diese im Vergleich zu den restlichen Funktionen, direkt mit den Funktionen unsere Applikation konkurrieren.

##### 3.1.1 Karteikarten

Erstellen von digitalen Lernkarteikarten an die sich Bilder anhängen lassen. Nachdem welche erstellt sind, kann der Nutzer mit dem lernen dieser digitalen Karteikarten anfangen. Dabei wird eine Frage angezeigt und der Lernende hat die Möglichkeit auf den "Ich weiß die Antwort" oder "Ich weiß die Antwort nicht" Button zu

klicken. Beide Möglichkeiten führen dabei zur nächsten Frage. Jede Karte lässt sich auch umdrehen um die Rückseite samt richtiger Antwort einzusehen. Damit nicht immer mit der gleichen Frage begonnen wird, gibt es eine Funktion, mit der die ganzen Karten vermischt werden. Ist ein Lerndurchgang abgeschlossen, bekommt der Nutzer am Ende numerisch wie auch grafisch in Form eines Kreisdiagramms aufgezeigt, wie viele Antworten er kennt und wie viele er nicht kennt. Nach jeden Lerndurchgang lässt sich auch eine Leistungstatistik aufrufen. Anhand der Leistungstatistik ist die Lernentwicklung für den Lernenden überprüfbar. Hier werden alle vergangenen Versuche samt Ergebnissen aufgezeigt und daraus ein durchschnittlicher Score berechnet. User können die Sets durch einen Stern-Punkte-System wie es bei Amazon verwendet wird bewerten und auch Kommentare hinzufügen. Zudem lassen sich auf Wunsch alle digitalen Karteikarten auch direkt ausdrucken.

### **3.1.2 Quiz**

Bei der Quizerstellung stehen verschiedene Fragetechniken zu Auswahl. Einer dieser Fragetechniken ist Multiple-Choice. Hierbei stehen bei jeder Frage mehrere vordefinierte Antworten zur Auswahl. Der Benutzer definiert also eine Frage und dazu alle möglichen Antworten. Es kann nur eine Antwort als die Richtige angewählt werden. Optional ist noch eine Erklärung zu den einzelnen Fragen angebbbar. Die Erklärung lässt sich am Ende des Quiz einsehen. Nach einem Quizdurchgang ist am Schluss zu sehen, wie viel Zeit für das Quiz benötigt wurde als auch die Anzahl der falschen Antworten und die erreichten Punkte für die korrekten Antworten. Alle Antworten lassen sich nun überprüfen, um sehen zu können, welche als falsch und welche als richtig beantwortet wurden. Wie bei den Karteikarten ist auch hier eine Leistungstatistik aufrufbar. Eine weitere Technik stellt die Checkbox dar. Im Vergleich zu den Multiple-Choice Fragen unterscheidet sich diese Methode lediglich darin, dass mehrere Antworten anwählbar sind und richtig sein können. Die letzte Variante nennt sich Richtig oder Falsch. Wie der Name schon sagt, können hier Fragen nur mit dem richtig oder falsch Button beantwortet werden. Wie beide vorherigen Varianten wird auch hier nach dem Ende des Lernduchrgans die benötigte Zeit, die Anzahl der falschen Antworten, die erreichten Punkte und die Leistungstatistik aufgelistet.

### **3.1.3 Zusammenfassung**

GoConqr bietet nicht nur Studenten sondern auch Schülern und Lehrern ein sehr umfangreiches Lernwerkzeug an. Das moderne Design und die einfache Gebrauchstauglichkeit überzeugen und machen Neulingen den Einstieg einfach.

Auf erstellte Lerninhalte ist der Zugriff von überall aus durch einen Browser oder mobilen Endgeräten möglich, was das ganze ziemlich komfortabel macht. Hat jemand aber ernsthafte Interesse GoConqr intensiv für das Lernen zu nutzen, wird er nicht drumherum kommen Geld dafür auszugeben und sich eine Premium Version zu holen, da die Basisversion nur eine begrenzte Anzahl an Ressourcen zu Verfügung stellt. Hinzu kommt noch eine beschränkte Privatsphäre was die Lerninhalte angeht. In der gratis Version lassen sich Lerninhalte nur als öffentlich makieren und das führt dazu, dass jeder User von GoConqr Zugriff drauf hat und diese selbst zum lernen verwenden kann.

GoConqr versucht dabei mit dem Gesamtpaket der bereitgestellten Lernfunktionen zu überzeugen und dadurch Nutzer zum lernen zu motivieren. Ob das langfristig ausreicht ist eine andere Frage. An sich lernt jeder Nutzer für sich alleine und hat dann eben die Möglichkeit seine Lernstatistik einzusehen und mit anderen Nutzern zu vergleichen. Aber führt dieses Feature wirklich zu einer Steigerung der Motivation um mehr zu lernen und



dadurch mehr Punkte erreichen zu wollen als der andere ? Zwar vergleicht man sich mit dem anderen und sieht wer besser ist, aber dadurch steht der Nutzer nicht automatisch in einem direkten Duell dem anderen gegenüber.

### **3.2 Anki.com**

Anki 暗記, das japanische Wort für Auswendiglernen, ist eine quelloffene Lernkarteissoftware, die vorrangig auf das Lernen von Fremdsprachen abzielt. Durch verschiedene Einstellmöglichkeiten können allerdings auch unterschiedliche Inhalte gelernt werden. Anki implementiert den SuperMemo2 Algorithmus. SuperMemo implementiert die Methode des Wiederholens mit Lücken auch noch "Spaced Repetion" genannt.

Anki bietet die Möglichkeit sich öffentliche Kartenstapel auf den Rechner zu importieren. Grundsätzlich erstellt der Nutzer seine Kartenstapel selbst.

Dazu nutzt Anki Notizen mit vorgefertigten Feldern in welche die nötigen Informationen eingetragen werden. Aus diesen Notizen und anhand eines Notiztyps werden die Karten erzeugt. Zusätzlich zu den Feldern kann der Benutzer Schlagwörter angeben die es ermöglichen nur einen Teil eines Stapel zu lernen.

#### **3.2.1 Lernen & Abfragen**

Im Lernmodus werden dem Benutzer die Frage und die Antwort gleichzeitig angezeigt. Beim Abfragen seines Wissens wird nur die Frage angezeigt und die Antwort kann aufgedeckt oder eingegeben werden. Der SuperMemo Algorithmus kommt hier zum Einsatz. Der Algorithmus sorgt dafür, dass leichte Fragen nach längeren und schwere Fragen nach kürzeren Zeiträumen wiederholt werden. Das Intervall wird durch den Schwierigkeitsgrad einer Karte bestimmt. Die zusätzlichen Schlagwörter erlauben es bestimmte Themenbereiche auszublenden oder gesondert zu lernen. Der Nutzer kann die Menge an Karten und die dafür zu Verfügung stehende Zeit pro Tag anpassen.

#### **3.2.2 AnkiDroid**

AnkiDroid ist eine mobile Android-Implementation von Anki. Sie bietet die grundlegenden Funktionen wie das Erstellen und Bearbeiten von Karteikarten. Bereits erstellte Karteikarten können mit dem AnkiWeb-Server oder durch einfaches kopieren, synchronisiert werden. Durch den Einsatz eines Smartphones können die Karten mit Fotos und Tonspuren versehen werden. Zusätzlich kann der Benutzer sich die Karten in verschiedenen Sprachen vorlesen lassen. Dazu kann ein TTS (Text-to-Speech System), wie das bereits auf Androidgeräten vorhandene Google-TTS, verwendet werden.

#### **3.2.3 Besondere Merkmale**

Anki zeichnet sich dadurch aus, dass es zusätzlich zu reinen Textkarten noch Multimedia-Dateien unterstützt. Durch das anreichern mit Audio- und Video Material werden die Karten aussagekräftiger und einfacher zu lernen.

Besonders bei komplexen Darstellungen wie z.B Mathematischen Formeln, zeigt sich der Vorteil der LaTeX & HTML Kompatibilität. Wenn die Grundfunktionalitäten von Anki nicht ausreichend sind, dann können über die Plug-In Schnittstelle noch viele weitere Funktionen hinzugefügt werden.

### **3.2.4 Zusammenfassung**

Anki erreicht eine hohe Flexibilität beim Erstellen der Karteikarten. Das Speichern von Notiz-Typen erleichtert das Generieren von ähnlichen Karten. Die Unterstützung von Multimedia-Formaten ist ein sinnvolles Feature und wertet die Karteikarten auf. Die Umsetzung des Spaced Repetition mit Hilfe des SuperMemo2 macht die Kernfunktion und Motivation dieser Software aus. Anki besitzt eine große Community und leider nutzt die Software diese nicht auf um z.B sich mit anderen Gegnern messen zu können, in Gruppen zu lernen oder arbeiten. Lediglich der manuelle Austausch der eigenen Karteien mit Freunden ist möglich. Ein Feature welches ein vernetztes Lernen mit anderen Benutzern ermöglicht wäre eine sinnvolle Erweiterung. Die Abwesenheit dieser Funktion stellt jedoch keinen Nachteil für die Benutzung der Software dar.

### **3.3 Fazit**

Die Marktrecherche zeigt, dass ähnliche Anwendungen bereits bestehen und sich diese durch ihre Anzahl an Funktionen in der Recherche hervorheben. Teilweise werden spezielle Algorithmen eingesetzt, welche es dem Benutzer erleichtern sollen die Informationen auf den Karteikarten zu memorisieren. Die Hauptfunktion des Lernens mit Hilfe von Karteikarten ist in den Anwendungen gut umgesetzt, jedoch bieten diese nicht die Möglichkeit sich mit anderen Benutzern zu vergleichen oder gar gegen diese anzutreten.

Letztlich erfüllen alle analysierten Anwendungen ihren angestrebten Zweck, dennoch lernt der Benutzer immer noch im Alleingang ohne Interaktion mit anderen Nutzern. Durch entsprechende Features könnte die User-Experience und die User-Motivation verbessert werden.

## 4 Alleinstellungsmerkmale

Anhand die Marktanalyse wird ersichtlich, dass bereits einige Produkte existieren, welche dem Benutzer mit Hilfe von Karteisystemen das Lernen vereinfachen. Learn2Quiz! verfolgt jedoch den Ansatz, dass die Benutzer nicht nur im Alleingang lernen sollen, sondern die Möglichkeit haben sich an anderen Benutzern zu messen. Der Wettkampfgedanke und die daraus resultierende Motivation seine Gegner zu schlagen, sollen den Benutzer dazu anspornen öfters zu Lernen und somit sein Wissen zu verbessern.

Anhand die Marktanalyse wird ersichtlich, dass bereits einige Produkte existieren, welche dem Benutzer mit Hilfe von Karteisystemen das Lernen vereinfachen. Learn2Quiz! verfolgt jedoch den Ansatz, dass die Benutzer nicht nur im Alleingang lernen sollen, sondern gemeinsam im Team. Das System bildet dabei automatisiert anhand von den Fähigkeitslevel der einzelnen Spielern ein Team, bei denen möglichst alle Spieler eine ähnliche Spielstärke besitzen.

Dabei leitet sich der Fähigkeitslevel aus einem Punktesystem ab. Um die Anwendung und das Lernen interaktiver zu gestalten und somit von der Konkurrenz abzuheben, hat jeder Spieler während eines Quiz die Option, seine Teamglieder als Joker für eine Hilfestellung heranzuziehen, um eine gewünschte Frage zu lösen. Zudem können am Ende eines jeden Quiz offene Fragen von einzelnen Teammitglieder die während des Spiels entstanden sind, kollaborativ im Team ausdiskutiert und gelöst werden.

Jeder Spieler bekommt für richtig beantwortete Fragen sowie Hilfestellungen an andere Teammitglieder Punkte. Am Ende von jedem Spiel können kollaborativ im Team offene Fragen

Learn2Quiz! soll dem Benutzer, durch eine angemessene Gestaltungslösung sowie die Gebrauchstauglichkeit des Systems, ermöglichen sein Lernziel möglichst unkompliziert und kostengünstig zu erreichen.

## 5 Methodischer Rahmen

In der Mensch Computer Interaktion unterscheidet man grundsätzlich zwei verschiedene Ansätze zur Gestaltung interaktiver Systeme. Bevor die Wahl der Methoden getroffen werden kann, muss entschieden werden ob das Verfahren des benutzerorientierten oder nutzungsorientierten Design angewendet wird.

### 5.1 Abwägung der Vorgehensmodelle

Ein wichtiger Schritt beim Entwerfen von interaktiven Systemen, ist die Wahl des Vorgehensmodells für den anstehenden Gestaltungsprozess. Zunächst müssen die Unterschiede zwischen den Ansätzen des Benutzer- und Nutzungsorientierten Design aufgezeigt werden. Letztlich wird ein Verfahren gewählt durch welches sich die Ziele des zu entwickelnden Systems am besten umsetzen lassen.

Der Nutzungsorientierte Gestaltungsprozess nach Constantine und Lockwood basiert auf formalen und abstrakten Modellen die eher die Interaktion mit dem System beschreiben. Der Gestaltungsprozess bezieht den Nutzer mit seinen Zielen und Aufgaben nur teilweise mit ein.

Der Nutzerorientierte Ansatz verfolgt genau das gegenteilige Ziel. Der Nutzer wird aktiv in den Gestaltungsprozess mit einbezogen. Das präzise modellieren des Benutzer mit seinen Aufgaben, Zielen und Erwartungen, erlaubt das gestalten gebrauchstauglicher Anwendungen.

Aus der Problemstellung und der Zielsetzung (s. Exposé) lässt sich das eigentliche Ziel des zu entwickelnden Systems ableiten. Der Student als Nutzer des Systems verfolgt die Absicht sich weiter zu bilden um entsprechend gut für einen Test vorbereitet zu sein. Dabei soll der Gedanke des kollaborativen Lernens & der Motivation, ihn dabei unterstützen dieses Ziel zu erreichen. Das Hauptaugenmerk liegt also in diesem Fall auf dem Benutzer und seinen individuellen Eigenschaften. Dadurch, dass sich die primären Stakeholder präzise definieren lassen, erlaubt das Benutzerorientierte Verfahren die Ausarbeitung einer entsprechend guten Gestaltungslösung für diese Zielgruppe. Zumal reale Personen in den Prozess einbezogen werden können, kann die Lösung in Anbetracht mehrerer Iterationen des Prozesses verfeinert werden.

Die entgültige Entscheidung fällt dementsprechend auf das Benutzerorientierte Verfahren, da es die Zielsetzung des Projekt unterstützt und die entsprechenden Methoden und Techniken anbietet eine gebrauchstaugliche Lösung in Zusammenarbeit mit dem Nutzer zu entwickeln.

### 5.2 Methoden und Techniken

Der Benutzer-zentrierte Gestaltungsprozess umfasst mehrere Methoden und Techniken

#### 5.2.1 Discount Usability Engineering

Bei dieser Methode, entwickelt von Jakob Nielsen, handelt es sich um eine Evaluationsmethodik, die schnell und einfach in der Handhabung ist und zudem noch kostengünstig. Bei der kurzen Entwicklungszeit die das Projekt mit sich bringt und die nicht so erfahrenen Entwicklern was die Gestaltung von Mensch und Computer interaktiven Systemen angeht, ist diese Methodik auf jeden Fall in Betracht zu ziehen. Mit

Hilfe der darin verwendeten drei Stufen der Evaluation, dem Designprozess und der Analyse können schon viele Fehler des Systems aufgedeckt werden. Ein Nachteil ist der, dass die Ausführungsschritte der Methoden genau vorgegeben sind. Es würde sich somit anbieten, als Ergänzung die Norm DIN ISO 9214 Teil 210 heranzuziehen. Diese gibt keinerlei Vorgaben bezüglich Methoden und Techniken an und kann somit besser an das Projekt angepasst werden. Dadurch sollen bessere Ergebnisse im Gestaltungsprozess erreicht werden, was die Bedürfnisse und Erwartungen der Benutzer angeht, die ja expliziter Bestandteil der Entwicklung sind.

### **5.2.2 Usability Engineering Lifecycle**

Das Modell des Usability Engineering Lifecycle nach Deborah Mayhew ist, verglichen mit dem Discount Usability Engineering, ein sehr kostenintensives und komplexes Modell. Der Entwicklungsprozess wird in 3 Stufen eingeteilt: Am Anfang steht die “requirements analysis”, dann das “design, testing and development” und zuletzt die “installation”. Für den Teil des Design, Testen und Entwickeln muss ein großer Zeitaufwand eingeplant werden, da diese Etappe sehr komplex ist und zusätzlich müssten starke Anpassungen vorgenommen werden um dieses Modell in der angestrebten Projektdauer sinnvoll einsetzen zu können. Desweiteren legt dieses Modell verstärkt Wert auf die Analyse des Nutzungskontexts als auf die Analyse der Nutzeranforderungen. Bei Learn2Quiz! sollten die Benutzeranforderungen jedoch detaillierter betrachtet werden um den Bedürfnissen des Nutzers gerecht zu werden.

### **5.2.3 Scenario Based Usability Engineering**

Wie der Name schon andeutet, basiert das Scenario Based Usability Engineering nach Rosson und Carroll auf der Erstellung von Szenarien. Diese Szenarien werden auf allen Modellstufen in den Gestaltungsprozess, unter Anbetracht des jeweiligen Nutzungskontextes, einbezogen. Obwohl dieses Modell nicht so kostenintensiv ist wie das Usability Engineering Lifecycle, stellt die Fülle der zu erstellenden Szenarien jedoch einen erheblichen Aufwand dar. Aus diesem Grund ist dieses Modell eher ungeeignet für Learn2Quiz!, da der zeitliche Aufwand den Rahmen des Projekts überschreiten würde.

[http://edutechwiki.unige.ch/en/Scenario-based\\_usability\\_engineering](http://edutechwiki.unige.ch/en/Scenario-based_usability_engineering)

## **5.3 Fazit**

Letztlich lassen sich alle genannten Techniken für den Gestaltungsprozess einsetzen, jedoch sind das “Usability Engineering Lifecycle” und das “Scenario Based Usability Engineering” sehr kostenintensiv und teilweise sehr komplex und eignen sich aus dem Grund eher weniger für ein Projekt welches zeitlich sehr begrenzt ist.

Die Wahl fällt dementsprechend auf das Discount Usability Engineering nach Jacob Nielsen. In Kombination mit der DIN EN ISO 9241 Teil 210/110 bietet dieses Modell die bestmögliche Voraussetzung, eine kostengünstige und dennoch detaillierte Gestaltungslösung zu entwickeln. Der Fokus liegt dabei auf der Gebrauchstauglichkeit sowie der Verfolgung der angestrebten Ziele. Nachfolgend werden die verwendeten Techniken und der Ablauf des Gestaltungsprozesses genauer beschrieben und festgelegt.

## 5.4 Vorgehensmodell

Das folgende Schaubild erläutert die einzelnen Phasen des menschenzentrierten Gestaltungsprozesses. Dadurch dass die DIN EN ISO 9241-210 lediglich ein Rahmenwerk vorgibt, wird zusätzlich das Discount Usability Engineering mit seinen Methoden hinzugezogen. Daraus ergibt sich ein Vorgehensmodell welches in 3 Phasen gegliedert ist.

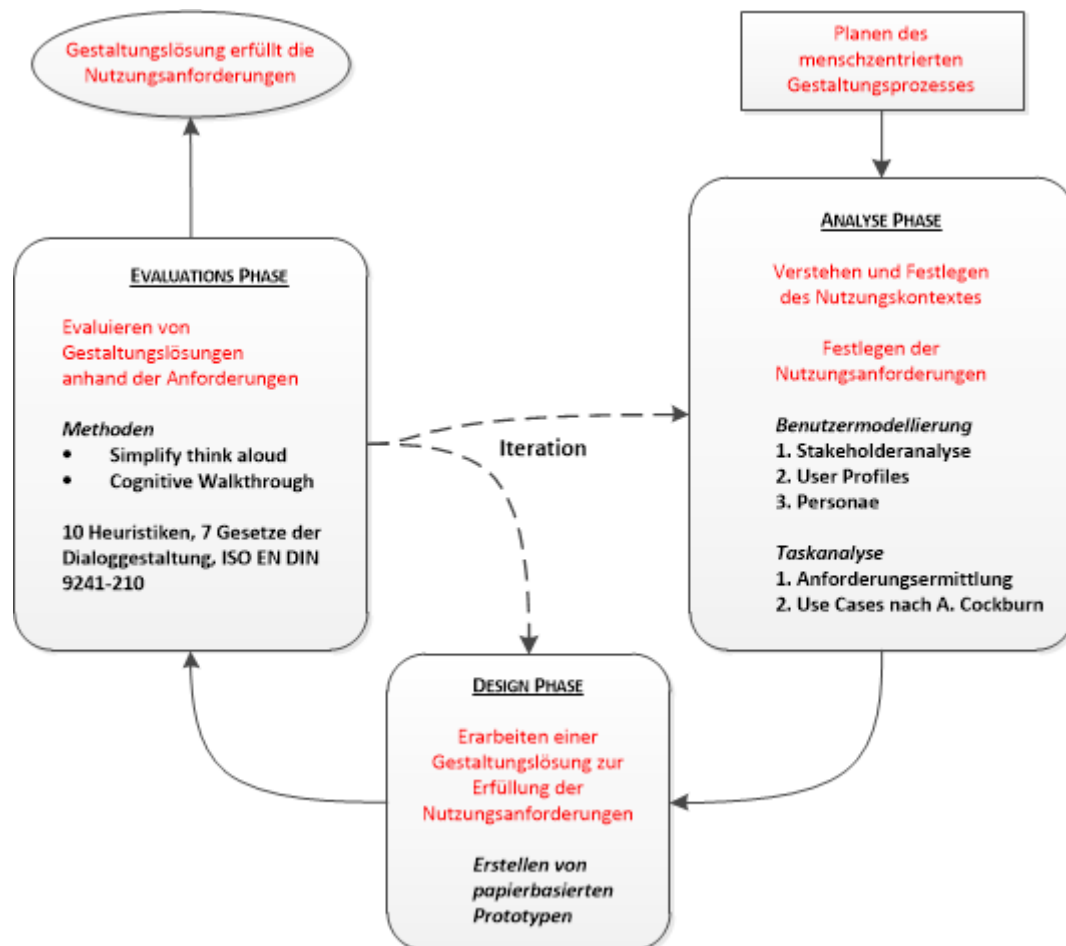


Abbildung 1: Ablauf des Gestaltungsprozesses

An erster Stelle steht die Analysephase. Sie dient dazu den Nutzungskontext sowie die Nutzungsanforderungen des Benutzer zu verstehen und festzulegen. Das Discount Usability Engineering setzt dabei auf Szenarien und Storyboards um die Benutzer zu modellieren. Jedoch beschreiben diese Techniken den Benutzer nur sehr grob und ermöglichen deshalb keine detaillierte Benutzermodellierung. Aus diesem Grund werden die Benutzer anhand von User Profiles und Personae beschrieben. Bereits in der Analysephase sollen reale Personen in den Prozess mit eingebunden werden um zusätzliche Informationen über die Stakeholder zu erhalten. Die Nutzungsanforderungen werden anhand von Use Cases nach Alistair Cockburn erstellt. Diese Use Cases dienen im weiteren Verlauf des Projektes als Anhaltspunkte für das Design und die Evaluation der Gestaltungslösung.

In der Designphase geht es primär darum, Gestaltungslösungen anhand eines papierbasierten Prototypen zu erstellen. Dazu werden die in der Analysephase ermittelten

Use Cases hinzugezogen. Die Prototypen dienen dazu die Gestaltungslösung mit realen Personen zu evaluieren und sich dabei auf die wesentliche Benutzerführung zu konzentrieren.

Letzlich werden in der Evaluationsphase die Gestaltungslösungen von realen Benutzern getestet. Das Discount Usability Verfahren bietet dafür die Simplify think aloud-Methode und zusätzlich die 10 Heuristiken nach Jacob Nielsen. Um weitere Fehler in den Gestaltungslösungen zu finden wird zusätzlich die Methode des cognitive walkthrough hinzugezogen. Weiterhin dienen die 7 Grundsätze der Dialoggestaltung als Anhaltspunkte für die Evaluation der Gestaltlösungen.

Grundsätzlich sind in der Evaluationphase mehrere Iterationen vorgesehen. Um den Nutzungsanforderungen gerecht zu werden und diese zu erfüllen, reichen diese Iterationen zurück in die Analyse- bzw. Designphase. Dieses Vorgehen ermöglicht es noch ihm späteren Verlauf des Gestaltprozesses, anfängliche Fehlentscheidungen zu korrigieren. Die letzte Phase gilt als beendet wenn eine Gestaltungslösung vorliegt, den alle Anforderungen erfüllt.

## 6 WBA-Techniken

### 6.1 Kommunikationsabläufe

Bevor die Systemarchitektur ausgearbeitet werden kann, muss sich Gedanken über die im System auftretenden Interaktionen gemacht werden. Die nachfolgenden Abbildungen illustrieren die Semantik der wichtigsten Kommunikationsabläufe zwischen den einzelnen Benutzern und den zentralen Einheiten des Systems.

#### 6.1.1 Simple Interaktionen

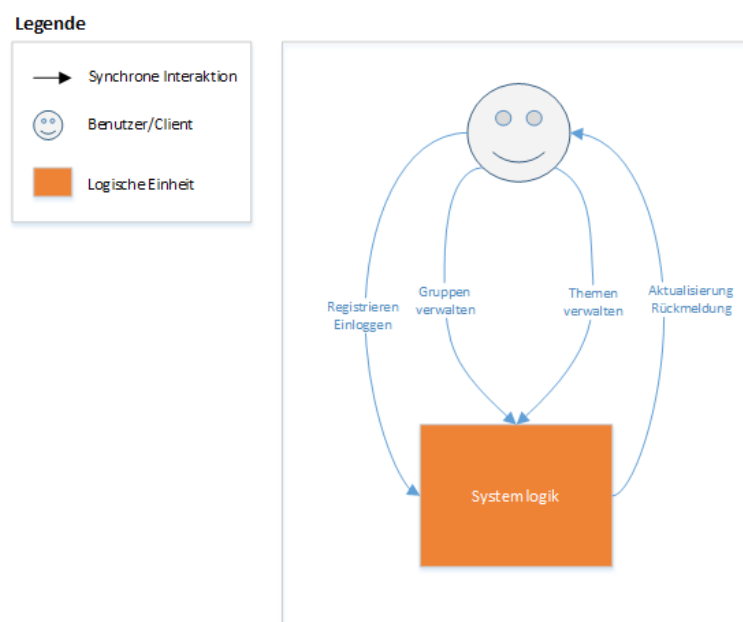


Abbildung 2: Simple Interaktion zwischen Spieler & System

Unter simplen Interaktionen verstehen sich in diesem Fall alle Funktionen welche zum Verwalten der Systemressourcen dienen. Dazu gehört das Registrieren und Anmelden mit Benutzerdaten, sowie die Verwaltung der Gruppen, Themengebiete und Karteikarten, d.h das Erstellen und Bearbeiten dieser Ressourcen. Das System gibt in allen Fällen eine einfache Rückmeldung bzw. Aktualisierung der veränderten Ressourcen zurück. Die Anmeldung erfordert zusätzlich ein Handshaking zwischen dem Benutzer und dem System.

#### 6.1.2 Komplexe Interaktionen

Die komplexen Interaktionen bezeichnen alle Funktionen die sich auf den Spiel Ablauf und die Kooperation zwischen den Spielern beziehen. Die Kommunikation teilt sich 3 grundlegende Szenarien auf. Zuerst wird die Interaktion zwischen den Spielern und dem System in einer Spielsession betrachtet werden.



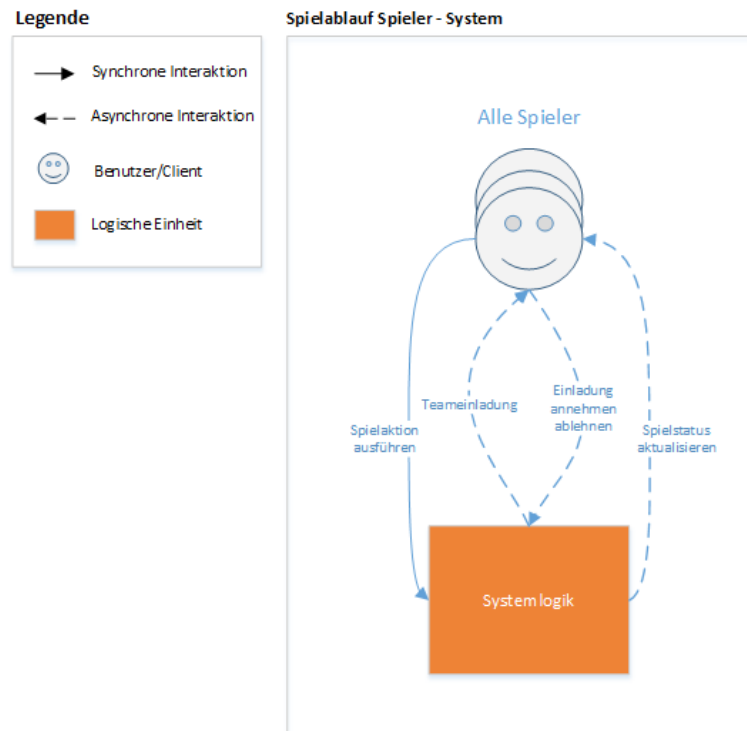


Abbildung 3: Grundlegender Spielablauf zwischen Spieler & System

Eine Spielsession startet dadurch dass ein Spieler sich zum Spielen bereit erklärt, er befindet sich somit in einer Warteschlange. Das System ermittelt nun eine bestimmte Anzahl von möglichen Mitspielern und sendet eine asynchrone Einladung zum Team an diese Spieler. Die Spieler können je nach Laune die Einladung annehmen oder ablehnen. Wenn das Team voll ist, wird das Team geschlossen und keine Spieler können mehr beitreten. Das System meldet den Spielern den bevorstehenden Spielbeginn und startet ein neues Quiz. Nun werden allen Spielern eine bestimmte Anzahl von gleichen Karten in der gleichen Reihenfolge angezeigt. Zu jeder Karte muss der Spieler, in einer festgelegten Zeit, die richtige Antwort geben. Kennt er die richtige Antwort nicht, dann kann er die anderen Spieler zur Hilfe ziehen. Diese Interaktion wird durch *Abbildung 4: Kooperation zwischen Spielern während eines Spiels* beschrieben. Das Spiel endet wenn alle Spieler die Karten bearbeitet haben. Nach dem Spiel startet dann die Diskussionsrunde welche in *Abbildung 5: Die Diskussionsrunde mit zentraler Einheit zwischen den Spielern* dargestellt ist.

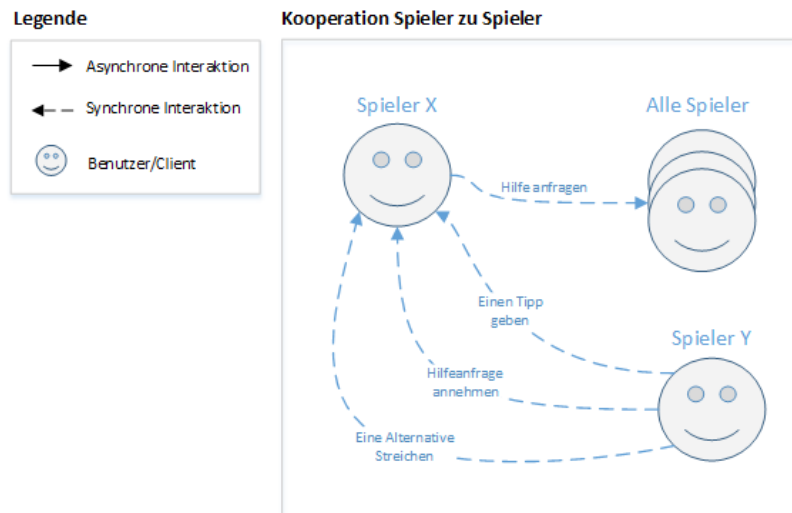


Abbildung 4: Kooperation zwischen Spielern während eines Spiels

Abbildung 4: Kooperation zwischen Spielern während eines Spiels zeigt die bereits erwähnte Kooperation zwischen den Spielern. Sollte Spieler X eine Antwort nicht kennen, kann er Hilfe bei seinen Mitspielern anfragen. Der Spieler der am schnellsten auf die Anfrage reagiert, darf dem anderen Spieler eine Hilfestellung geben. Je nach gewählten Joker, wäre das das Streichen einer Alternative oder das Auswählen eines Tipp. Die Interaktionen sind in diesem Fall alle asynchron, da der Spieler nicht beim Spielen unterbrochen werden soll.

Die übertragenen Daten sind als flüchtig, also nicht-persistent anzusehen und dienen nur dem Ablauf innerhalb des Spiels.

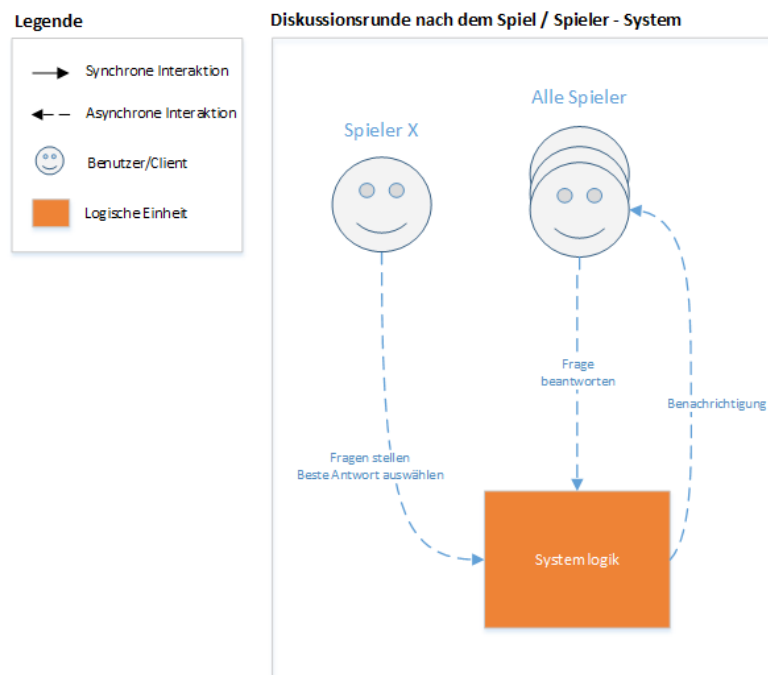


Abbildung 5: Die Diskussionsrunde mit zentraler Einheit zwischen den Spielern

Die Diskussionsrunde nach einem Spiel dient dazu etwaige Fragen zu Karteikarten innerhalb des Teams zu klären. Die Systemlogik wird hierbei zur zentralen Verwaltung dieser Diskussion genutzt, da die dort anfallenden Daten bzw. Inhalte persistent und für jeden zugänglich gespeichert werden. Ein Spieler X kann zu einer gegebenen Karteikarte eine Frage stellen. Das System benachrichtigt alle Spieler im Team über den Eingang einer Frage und bietet die Möglichkeit die Frage zu beantworten.

## **6.2      *Systemarchitektur***

Es soll eine angemessene Systemarchitektur entworfen werden, die der Zielsetzung von Learn2Quiz! gerecht wird. Das bedeutet, dass die Benutzer eine Clientanwendung benötigen, über welche sie mit dem System interagieren können. Diese Anwendung wird auf einem Desktop PC ausgeführt. Geplant ist die Unterstützung von mobilen Endgeräten sowie eine plattformunabhängige Nutzung des Systems. Eine zusätzliche Anforderung ist eine möglichst geringe Antwortzeit in dem System um das Spielerlebnis nicht zu beeinträchtigen.

### **6.2.1      Verteilungsparadigmen**

Nach dem die Kommunikationsabläufe untersucht wurden, wird nun die Art der Verteiltheit der Anwendungslogik bestimmt. Dazu werden das Peer-to-peer, Client-Server und Publish-Subscribe Paradigma auf ihre Tauglichkeit für den Einsatz bei Learn2Quiz! untersucht.

In einem Peer-to-peer-Netz sind alle Systemkomponenten gleichberechtigt. Es gibt in solchen Strukturen keine "festen" Vermittler, also Server, sondern jeder Teilnehmer kann als Server fungieren. Learn2Quiz! verfolgt jedoch den Ansatz eines kollaborativen Quiz-Spiels, welches in einer Spielsession ausgeführt wird. Diese Spielsession stellt die zentrale Instanz dar und kann somit eher auf ein Client-Server-Paradigma abgeleitet werden. Zusätzlich gibt es in P2P-Netzen keine zentrale Datenhaltung, welche jedoch in diesem Fall nötig wäre um die Spielstammdaten persistent zu speichern. Ein Client müsste also in einem P2P-Netz über jegliche Daten verfügen oder er müsste diese mit den anderen Nutzern austauschen, was einen erheblich größeren Kommunikationsaufwand mit sich bringt.

Ein Publish-Subscribe Paradigma würde sich generell eignen, jedoch stellt die lose Kopplung zwischen Publisher und Subscriber in sofern ein Problem dar, dass der Publisher seine Subscriber nicht wirklich kennt. Learn2Quiz! benötigt jedoch die Möglichkeit dass der Benutzer einen direkten Austausch mit dem System vornehmen kann, er also Anfragen an das System sendet und auf eine Antwort vom System wartet bis er weiterarbeiten kann. Dazu muss der Nutzer im System bekannt und erreichbar sein. Eine reine Publish-Subscribe Lösung wäre also eher ungeeignet.

Eine Client-Server Architektur erscheint am Sinnvollsten. Jedoch benötigt Learn2Quiz! zusätzlich zu synchronen Interaktionen ebenso die Möglichkeit asynchrone Interaktionen zwischen den Komponenten zu ermöglichen. Es ist also sinnvoll die Vorteile vom Client-Server Prinzip zusammen mit der der losen Kopplung des Pub-Sub-Modells zu vereinen.

### **6.2.2      Systemkomponenten**

Das System besteht aus zwei Hauptkomponente, dem Server und mehreren Clients. Der Server implementiert die Spiellogik, also die Verwaltung von Spielsessions, die Anfrage

von Nutzeraktionen und die Verwaltung der Karteikarten. Zur Spiellogik gehört zudem das automatische Ermitteln von passenden Teammitgliedern für ein Spiel. Um diese Verteilung von Spielern zu ermöglichen, benötigt der Server den Fähigkeitslevel eines jeden Spielers. Die Ermittlung des Fähigkeitslevels gehört zu der Anwendungslogik des Clients, welcher die Informationen an den Server weitergibt. Zusätzlich implementiert der Client die Teamfunktionen, also das gegenseitige Helfen zwischen den Nutzern. Damit Client und System miteinander kommunizieren können wird ein Webservice benötigt. Dieser ermöglicht die synchronen Interaktionen, also das Ausführen von Spielaktionen. Zusätzlich können zwischen die Nutzer untereinander oder mit dem Server über asynchrone Interaktionen kommunizieren. Dazu wird eine passende Middleware benötigt. In den folgenden Abschnitten werden mögliche Webservices und Middlewares beschrieben und nach bestimmten Kriterien bezogen auf Learn2Quiz! Bewertet.

### 6.2.3 Wahl der Middleware

Um die Kommunikation zwischen den Systemkomponenten synchron als auch asynchron abzuwickeln, wird eine Middleware benötigt. Dementsprechend stehen folgende Middlewarearten zur Auswahl:

- Dienst-orientiert
- Nachrichten-orientiert
- Objekt-orientiert
- TCP/Sockets (low-level)

Eine Dienst-orientierte (service-oriented) Middleware eignet sich in erster Linie zur Realisierung von synchronen Kommunikationen und bietet eine gute Interoperabilität von verschiedenen Systemen. Eine Plattformunabhängigkeit ist wünschenswert, um spätere Implementierung von Learn2Quiz! auf mobilen Geräten unterschiedlicher Anbieter zu ermöglichen.

Diesen Vorteil bieten auch Nachrichten-orientierte Middleware. Diese ermöglichen asynchrone, bzw. ereignisgesteuerte Kommunikation in verteilten Systemen.

Objekt-orientierte Middleware bietet den Vorteil dass ihre Komplexität gering ist und sich technisch relativ leicht umsetzen lässt. Nachteil ist leider die geringe Interoperabilität, welche eine plattformübergreifende Nutzung von Learn2Quiz! per se ausschließt.

Letztlich bietet sich die Möglichkeit eine Low-level Middleware mit Hilfe von TCP-Sockets umzusetzen. Hinsichtlich Desktop-Clients sind TCP-Sockets eine einfache Lösung, jedoch gestaltet sich eine spätere Anbindung von mobilen Geräten etwas schwieriger. Ausschlaggebend sind dabei mögliche Konnektivitätsprobleme die durch eine schlechte Netzinfrastruktur auftreten können. Für Learn2Quiz! erscheint außerdem eine High-Level-Implementierung ausreichend zu sein, deshalb werden TCP-Sockets als Middleware nicht weiter in Betracht gezogen.

Dieser Vergleich macht deutlich, dass sich für die asynchrone Kommunikation eine nachrichten-orientierte Middleware eignet. Die asynchronen Interaktionen zeigen ein eher geringes Datenvolumen auf, da sie hauptsächlich für Benachrichtigungen und den Austausch zwischen den Nutzern verwendet werden. Wohingegen der datenlastigere Verkehr, also der Austausch von Spiel-Stammdaten zwischen Server und Client über einen Webservice realisiert werden soll. Das System wird durch diese Kombination flexibler und kann im späteren Verlauf gut auf mobile Endgeräte angepasst werden.

### 6.2.4 Wahl der Service Architektur

Die einzelnen Systemkomponenten müssen über das Internet mit einander kommunizieren. Hierfür stehen Webservices im REST- oder im SOAP-Stil zur Verfügung. Es soll nun entschieden werden welche Architektur sich am Besten für Learn2Quiz! eignet.

Nachteilig bei SOAP-Architekturen ist die allgemein höhere Bandbreite gegen über von RESTful-Webservices. Der Overhead ist bei SOAP wesentlich höher und das Datenformat ist auf XML beschränkt.

RESTful-Webservices bieten den wesentlichen Vorteil, dass das Datenformat nicht auf XML festgelegt ist. Dies ermöglicht es, wesentlich kleinere Nachrichten zu versenden.

Grundsätzlich es es möglich beide Architekturstile zur Implementierung eines Webservice zu verwenden. Jedoch eignen sich RESTful-Webservices sehr gut für kleinere Projekte und dennoch bleibt das System, mit Blick auf die Zukunft, flexibel und skalierbar. Die Möglichkeit ein anderes Datenformat als XML einzusetzen um somit den Overhead möglichst gering halten, unterstützt die Auswahl eines RESTful-Webservice.

### 6.2.5 Nachrichtenorientierte Middleware

XMPP ist ein Protokoll welches XML als Datenformat nutzt. Das bedeutet, dass die Nachrichten unter XMPP größer sind als dies bei den binärkodierte Protokollen AMQP und MQTT der Fall ist. XMPP bietet jedoch den Vorteil dass es auf den Benutzer bezogen ist. So ist jeder Nutzer durch eine eindeutige ID ermittelbar. MQTT und AMQP benötigen jedoch keine 'Roster'-Verwaltung was sich bei Peer-to-Peer Kommunikationen als vorteilhaft erweist. MQTT benötigt, gegenüber von XMPP (XEP-0060), keine Erweiterung um Pub-Sub-Services zu implementieren.

Die Wahl des Protokolls gestaltet sich schwierig, da jedes Protokoll seine Vor- und Nachteile hat. Nachteilig bei XMPP ist die Festlegung auf das Datenformat XML. Unter MQTT und AMQP können verschiedene, teilweise sparsamere Protokolle eingesetzt werden. Jedoch hat das Entwicklerteam keine Erfahrungen um Umgang mit MQTT und AMQP, was mit sich bringt, dass eine gewisse Einarbeitungszeit eingeplant werden muss und somit der Zeitrahmen überschritten werden könnte. Da bereits Erfahrungen im Umgang mit XMPP bestehen, fällt die Wahl auf dieses Protokoll. Möglicherweise entspricht dies nicht der optimalen Wahl, jedoch sind die Nachteile unter Anbetracht des Zeitaufwandes zu vernachlässigen.

## 7 Geschäftsmodell

Um die Entwicklungskosten abzudecken und bestenfalls auch noch einen Gewinn erwirtschaften zu können, müssen verschiedene Geschäftsmodelle in betracht gezogen werden. Dabei ist abzuwägen, welches dieser Geschäftsmodelle die beste Finanzierungsquelle/Lösung für das Projekt dargestellt. Werbung

Die Anwendung könnte Nutzern in vollem Umfang kostenlos zur Verfügung gestellt werden. Dabei müssten diese nur die Schaltung von Werbung zum Beispiel in Form eines Banners in Kauf nehmen. Zu berücksichtigen wäre, dass die Schaltung der Werbung nicht während eines Lernvorgangs stattfinden dürfte, da es sonst einen Störfaktor darstellt und der Schüler / Student abgelenkt wird. Es müssten passende Werbepartner gefunden werden, die zwangsläufig Studenten und Schüler zur Zielgruppe haben. Ob sich da

genügend finden würden ist ungewiss, zumal auch der Inhalt der zuschaltenden Werbung, nicht mit den eigentlichen Zielsetzungen der Anwendung kollidieren sollte. Es würde eben keinen Sinn machen für Computerspiele zu werben, wenn diese für das lernen kontraproduktiv sind.

### **7.1 Freemium**

Das Geschäftsmodell von Freemium basiert darauf, dass das Basisprodukt kostenlos ist und das Vollprodukt hingegen kostenpflichtig. Nutzer sollen so zunächst angelockt werden und Ihnen dann Premiumdienste angeboten werden. Im Falle von Learn2Quiz könnte die Basisversion einen eingeschränkten Kapazitätsumfang enthalten. Ein User kann zum Beispiel nur eine limitierte Anzahl von Gruppen erstellen und beitreten, während in der kostenpflichtige Version keine Limitierungen enthalten sind. Eine andere Alternative wäre die Basisversion kostenlos mit Werbung anzubieten und für einen Aufpreis werbefrei. Da aber auch hier wie schon oben erwähnt, die Problematik bestehen würde einen passenden Werbepartner zu finden, fällt die Entscheidung bei dieser Methode auf die Variante mit dem eingeschränkten Kapazitätsumfang und ist dem Geschäftsmodell der reinen Finanzierung durch Werbung zu bevorzugen. Somit kommt Freemium für das Projekt eher in Frage.

### **7.2 Einmalzahlung**

Eine Einmalzahlung stellt auch eine Interessante Option für das Produkt da. Ein Nutzer würde mit einer einmaligen Zahlung das Produkt erwerben und könnte es danach mit allen seinen Funktionen nutzen. Der Preis müsste dabei so gewählt werden, dass die Entwicklungskosten auf jeden Fall abgedeckt werden, aber auch noch mögliche Folgekosten. Das birgt jedoch die Gefahr, dass der anzusetzende Preis um diesen Anforderungen gerecht zu werden, für den Endkunden den Schülern und Studenten einfach zu hoch ist, da diese im Normallfall nicht so viel Geld zu Verfügung haben.

### **7.3 Abonnements**

Diese Variante stellt auch ein interessantes Geschäftsmodell für das Projekt da. Statt das Produkt mit einer einmaligen Zahlung zu erwerben, könnten Abonnements angeboten werden die von Interessenten auf nur eine bestimmte Zeit abgeschlossen werden. Ein Vorteil wäre sicherlich der, dass gegenüber der Einmalzahlung, die Preise für die Abonnements niedriger angesetzt werden könnten und so erschwinglicher für Studenten und Schüler sind. Außerdem garantiert diese Form einen kontinuierlichen Umsatz.

## **8 Risiken**

Die Entwicklung von neuen Systemen ist immer mit Risiken verbunden die es zu identifizieren gilt. Dabei wird zwischen Projektinternen und Projektspezifischen Risiken unterschieden. Es ist zu beachten, dass im laufe des Entwicklungsprozesses noch weitere Risiken hinzukommen können.

### **8.1 Geschäftsrisiken**

Die Marktrecherche ergab, dass es schon ein Vielzahl von ähnlichen Anwendungen auf den Markt gibt. Wird sich also das neue System neben den bestehenden überhaupt etablieren können oder greifen Nutzer zu den altbewährten Lösungen. Wie sollen mögliche Nutzer in

der Fülle von Konkurrenzprodukten bei Markstart vom System erfahren und dazu gebracht werden es zu testen? Es besteht die Gefahr in diesem Markt einfach unterzugehen. Diese Problematik wird durch die gewählten Alleinstellungsmerkmale adressiert. Diese sollen dafür sorgen dass sich Learn2Quiz! auf dem Markt behaupten kann. Zusätzlich kann ein gutes Geschäftsmodell die Risiken noch weiter minimieren.

## **8.2 Technische Risiken**

Durch den Einsatz bereits bekannter Technologien und Techniken sollen die technischen Risiken auf ein Minimum reduziert werden. Die genutzten Technologien sind außerdem ausreichend dokumentiert und erfordern keine aufwendige Einarbeitung seitens der Entwickler.

## **8.3 Implementierungsrisiken**

Die Implementierung des System ist ein wesentlicher Faktor der das Projekt zum scheitern bringen kann. Das Ziel des Projektes ist die Entwicklung eines kollaborativen Quiz-Lernspiels. Dadurch dass der Erfolg des Systems von den Benutzern abhängt, müssen die funktionalen Anforderungen genausten ermittelt werden. Zusätzlich spielen psychologische Faktoren eine wichtige Rolle. Das Systems sollte dem Benutzer einerseits Freude bereiten und anderseits ein nützliches Lerninstrument darstellen. Die Gebrauchstauglichkeit des Systems ist ausschlaggebend für die User-Experience, also müssen dementsprechend die MCI-Verfahren detailliert ausgeführt werden.

# **9 Proof-of-Concepts**

In der Proof of Concepts werden die wichtigsten Kernfunktionen getestet um absehen zu können, ob sich diese dann in der Realität auch umsetzen lassen.

## **9.1 Synchrone Kommunikation zwischen Client und Server & Speicherung der Daten in einer Datenbank**

- Was ist das Ziel des PoC?

Es soll möglich sein über eine Client-Desktopanwendung eine Nachricht im JSON Format über die POST-Methode an den Server zu übermitteln. Die übermittelten Daten sollen in einer MySQL Datenbank abgespeichert werden. Es muss ein RESTful Webservice in Java umgesetzt werden, dazu wird JAX-RS und ein Grizzly-Server eingesetzt. Das Un-/Marshalling der übertragenen Daten wird durch JAXB erledigt.

- Exit-Kriterium

Erfolgreiche Übermittlung von Dummydaten vom Client zu Server, mit anschließender persistenter Speicherung dieser Daten in der Datenbank.

- Fail-Kriterium

Daten können nicht erfolgreich übermittelt oder gespeichert werden.

- Mögliche Alternativen

Architektur abwägen, synchrone Kommunikation über XMPP realisieren, andere Datenbanken in Betracht ziehen

geplanter Termin und Status-

08.05.2015 / Status: offen

## **9.2      *Asynchrone Nachrichten von Server zu Client & Client zu Client***

- Was ist das Ziel des PoC?

Es soll möglich sein vom Server an den Client sowie von Client zu Client asynchrone Nachrichten zu verschicken und zu empfangen.

Es soll ein Openfire XMPP-Server und eine Client-Anwendung mit der Smack API entwickelt werden. Die Daten werden in XML übertragen. Es soll eine Pub-Sub Architektur mit Hilfe der XMPP Erweiterung XEP-0060 entworfen werden. Dazu sollen Dummy-Topics eingesetzt werden um die Funktweise zu testen.

- Exit-Kriterium

Erfolgreicher Übermittlung und Empfang der asynchronen Nachrichten von Server zu Client und von Client zu Client.

- Fail-Kriterium

Die übermittelten Daten konnten nicht versendet und empfangen werden.

- mögliche Alternativen

Architektur überdenken. Alternativ zu XMPP könnte MQTT verwendet werden.

- geplanter Termin und Status

08.05.2015 / Status: offen

## **9.3      *Demo-Desktopanwendung mit libGDX (Java Game Library)***

- Was ist das Ziel des PoC?

Es soll ein Demo-Spiel entwickelt werden, an welchem das Grundlegende Spielgeschehen ersichtlich wird. Dazu wird das Java-Spielframework libGDX verwendet. Es handelt sich dabei um ein Cross-Plattform Framework welches es ermöglicht Spiele für Desktop und mobile Clients schnell und einfach zu entwickeln. Das Spiel sollte die Grundfunktionen, also das Anzeigen und beantworten von Fragen ermöglichen. Es sollte eine Kommunikation mit Hilfe der PoC 1&2 zwischen Server und Client eingerichtet werden.



- **Exit-Kriterium**

Der PoC ist erfolgreich wenn: Fragen vom Server an den Client geschickt werden und dort mit Hilfe von libGDX angezeigt werden können. Zusätzlich muss der Benutzer eine Antwort auswählen können welche dann an den Server zurückgeschickt wird. Der Server überprüft die Antwort und gibt eine geeignete Rückmeldung.

- **Fail-Kriterium**

Wenn kein lauffähiger Client mit libGDX entwickelt werden kann und so das Spiel nicht testbar ist. Die Kommunikation zwischen Client und Server nicht besteht oder funktioniert.

- **Mögliche Alternativen**

Systemarchitektur und Technik überdenken, andere Spiel-Frameworks in Betracht ziehen, möglicherweise auch ganz ohne Framework

- **geplanter Termin und Status-**

08.05.2015 / Status: offen

## **9.4 Matchmaking (Einladungen an Benutzer senden)**

- **Was ist das Ziel des PoC?**

Das System sollte eine bestimmte Anzahl Benutzer zu einer neuen Spielsitzung einladen können. Die Benutzer erhalten eine asynchrone Benachrichtung und können dem Spiel beitreten. Die Wahl der Spieler bezieht sich auf ihr Fähigkeitslevel. Das System sollte eine gesunde Mischung von starken und schwachen Gegnern ermitteln.

- **Exit-Kriterium**

Das System bildet selbstständig Teams in dem es Benutzer zu einem Team einlädt und solange wartet bis die min. Anzahl von Spielern erreicht ist.

- **Fail-Kriterium**

Automatische Einladungen werden nicht empfangen oder versendet

- **Mögliche Alternativen**

Manuelles Einladen von Spielern

- **geplanter Termin und Status**

08.05.2015 / Status: offen