

# Entwicklung eines Web Based Training Systems nach einem Lernmodell

**Studienarbeit**

für die Prüfung zum

**Bachelor of Science**

des Studienganges Angewandte Informatik

an der Dualen Hochschule Baden-Württemberg Karlsruhe

von

**Michael Gruben** Meyle+Müller GmbH+Co.KG (Pforzheim) &

**Julian Babics** MedicalCommunications Soft- und Hardware GmbH (Bruchsal) &

**Benjamin Merkle** SAP AG (Walldorf)

Mai 2013

**Bearbeitungszeitraum**

**Betreuer**

**Gutachter**

2,5 Monate

Dr. Kay Margareth Berkling

Prof. Dr. Johannes Freudenmann

# Erklärung

Wir erklären hiermit ehrenwörtlich:

1. dass wir unsere Studienarbeit mit dem Thema *Entwicklung eines Web Based Training Systems nach einem Lernmodell* ohne fremde Hilfe angefertigt haben;
2. dass wir die Übernahme wörtlicher Zitate aus der Literatur sowie die Verwendung der Gedanken anderer Autoren an den entsprechenden Stellen innerhalb der Arbeit gekennzeichnet haben;
3. dass wir meine Studienarbeit bei keiner anderen Prüfung vorgelegt haben;
4. dass die eingereichte elektronische Fassung exakt mit der eingereichten schriftlichen Fassung übereinstimmt.

Wir sind uns bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

Karlsruhe, Mai 2013

Michael Gruben & Julian Babics & Benjamin Merkle

Copyright (C) 2013 Michael Gruben, Julian Babics, Benjamin Merkle.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

## **Zusammenfassung**

Im Verlauf der Bearbeitung der Studienarbeit „Analyse und Vergleich von Autorensystemen für ein WBT zu Vorlesungsinhalten“ ist in der Vorlesung „Gamification“ ein Konzept für ein WBT<sup>1</sup>-System entstanden. Aus dieser Vorstellung ist die Idee, und damit die Motivation der Studienarbeit, entstanden es in die Realität umzusetzen.

Es handelt sich um eine Webapplikation, die diverse WBTs in entsprechenden Kategorien zum Bearbeiten anbietet. Das Lernmodell der Gebrüder Dreyfuß wird in diese verwoben. In dem Modell wird die Kompetenz in einem Fachgebiet auf zwei unterschiedlichen Ebenen betrachtet, die fachliche Kompetenz und die Fähigkeit erklären zu können.

Zunächst wird die fachliche Kompetenz betrachtet. Demnach bearbeitet ein Neuling auf dem ersten Kompetenzlevel eines bestimmten Fachbereiches ein grundlegendes WBT, dessen abschließende Fragen nach vorgegebenen Schemata und grundlegender Eigenschaften beantwortet werden. Ein Experte auf dem vierten Kompetenzlevel muss hingegen Antworten auf Fragen wissen, die ein wesentlich komplexeres Verständnis eines Sachverhaltes verlangen.

Um seine Fähigkeit erklären zu können unter Beweis zu stellen, engagiert man sich mit Hilfestellungen für niedrigere fachliche Level. Beurteilen diese die Hilfestellung als gut, kann der Mastery Rang erreicht werden, der sich noch über dem Experten befindet. Nach dem Dreyfuß-Modell dürfen sich Lernender und Lehrender durch maximal zwei Level unterscheiden. Der Mastery-Level ist hingegen ein „erklärender Experte“, der nicht nur fachlich höchst kompetent ist, sondern auch sehr gut auch für einen Anfänger erklären kann, ohne in fachliche Details abzuschweifen.

Das WBT-System, welches beide beschriebenen Ebenen der Kompetenz organisiert, wird unter einer freien Lizenz veröffentlicht werden. So kann das als noch sehr simpel und eingeschränkt erwartete Ergebnis der Studienarbeit als Community Projekt weiterleben und weiterentwickelt werden. Bereits vor Bearbeiten der Studienarbeit wird damit gerechnet, dass nur ein kleiner und spezieller aber funktionaler Teil des Konzeptes umgesetzt werden wird. Der Fokus liegt dabei grundsätzlich mehr auf Funktionalität,

---

<sup>1</sup> Web Based Training

einer leicht zu erweiternden Architektur der Software und einem benutzerfreundlichem Interface, als auf einem gut aussehendem Design.

*„Die meisten Menschen sind bereit zu lernen, aber nur die wenigsten, sich belehren zu lassen.“*

Winston Churchill, Britischer Politiker und Nobelpreisträger, 1874 - 1965.

# Inhaltsverzeichnis

<b>Abkürzungsverzeichnis</b>	<b>X</b>
<b>1. Einleitung</b>	<b>1</b>
<b>I. Vorbereitungen</b>	<b>2</b>
<b>2. Projektplanung</b>	<b>3</b>
2.1. Motivation und Notwendigkeit . . . . .	4
2.2. Abgrenzung . . . . .	5
2.3. Zielsetzung . . . . .	5
2.3.1. Operative Ziele . . . . .	5
2.3.2. Strategische Ziele . . . . .	6
<b>3. Theoretische Grundlagen</b>	<b>7</b>
3.1. Prinzipien und Theorien . . . . .	7
3.1.1. Lernen . . . . .	7
3.1.2. Motivation . . . . .	8
3.1.3. Lernmodelle . . . . .	9
3.1.4. eLearning . . . . .	11
3.1.5. Flow . . . . .	12
3.1.6. Gamification . . . . .	13
3.1.7. freie Lizenzen . . . . .	14
3.2. Konzepte und Implementierungen . . . . .	15
3.2.1. Learning Management System . . . . .	15
3.2.2. Autorenwerkzeug . . . . .	15
3.2.3. WBT . . . . .	16
3.2.4. SCORM . . . . .	16
3.2.5. Gestaltungskonzepte . . . . .	20
3.2.6. Autorisierung und Authentifikation . . . . .	24
3.2.7. Ruby on Rails . . . . .	24
3.2.8. REST . . . . .	26
3.2.9. MVC Architektur . . . . .	27

<b>II. Konzeption der Plattform</b>	<b>28</b>
<b>4. Das zugrundeliegende Prinzip</b>	<b>29</b>
4.1. Namensgebung . . . . .	29
4.2. Freie Software . . . . .	29
<b>5. Architektur</b>	<b>31</b>
5.1. Erweiterbarkeit . . . . .	31
5.2. Aufbereiten des Dreyfus-Modells . . . . .	31
5.2.1. Fachlicher Rang . . . . .	32
5.2.2. Ränge für Tutoren . . . . .	32
5.3. Rollen für Anwender . . . . .	33
5.3.1. Administrator . . . . .	34
5.3.2. Lernender . . . . .	34
5.3.3. Tutor . . . . .	34
5.4. Dokumentation der API . . . . .	34
5.5. Automatisch generierte Filterung von Ergebnissen . . . . .	35
5.6. Themen . . . . .	35
<b>6. Anwenderseite</b>	<b>36</b>
6.1. Gamification . . . . .	36
6.1.1. Belohnungskriterien . . . . .	36
6.1.2. Etappenweise Herausforderungen . . . . .	36
6.2. Generieren von Motivation . . . . .	38
6.2.1. Intrinsisch motivierende Aspekte . . . . .	38
6.2.2. Weitere extrinsisch motivierende Merkmale . . . . .	39
<b>III. Implementierung</b>	<b>40</b>
<b>7. Entwurf</b>	<b>41</b>
7.1. Realisierungsmethodik . . . . .	41
7.2. Internationalisierung . . . . .	42
7.3. Beschreibung des Entwurfsklassendiagramms und Use-Case . . . . .	42
7.3.1. Use-Case Diagramm . . . . .	42
7.3.2. Entwurfsklassendiagramm . . . . .	45
7.4. Komponenten . . . . .	46
7.4.1. Durcharbeiten von WBTs . . . . .	46
7.4.2. Tutorensuche . . . . .	46
7.4.3. Navigation, Impressum, Kontakt . . . . .	46
7.4.4. Themen . . . . .	46



<b>8. Umsetzung</b>	<b>47</b>
8.1. Nutzerverwaltung . . . . .	47
8.2. Lokationen . . . . .	47
8.3. Suche . . . . .	47
8.4. SCORM . . . . .	47
8.5. Lizenzierung . . . . .	48
8.6. Dokumentation . . . . .	48
8.7. Gestaltung . . . . .	49
8.8. Authentifizierung . . . . .	49
8.9. Themen . . . . .	49
 <b>IV. Reflexion</b>	 <b>50</b>
<b>9. Zusammenfassung der Ergebnisse</b>	<b>51</b>
<b>10. Fazit und Ausblick</b>	<b>52</b>
10.1. Ideen für weitere Versionen . . . . .	52
10.1.1. Gamification für Newbies . . . . .	52
10.1.2. Gamification für Regulars . . . . .	53
10.1.3. Gamification für Enthusiasten . . . . .	53
10.1.4. Weitere Möglichkeiten für Gamification . . . . .	53
10.2. Anschließende Vorhaben . . . . .	54
 <b>Verzeichnisse</b>	 <b>54</b>
<b>Abbildungsverzeichnis</b>	<b>i</b>
<b>Tabellenverzeichnis</b>	<b>ii</b>
<b>Listings</b>	<b>iii</b>
<b>Literaturverzeichnis</b>	<b>iv</b>
 <b>Anhang</b>	 <b>vii</b>
<b>A. Abbildungen</b>	<b>viii</b>
<b>B. Tabellen</b>	<b>ix</b>

<b>C. GNU Free Documentation License</b>	<b>xi</b>
1. APPLICABILITY AND DEFINITIONS . . . . .	xi
2. VERBATIM COPYING . . . . .	xiii
3. COPYING IN QUANTITY . . . . .	xiv
4. MODIFICATIONS . . . . .	xv
5. COMBINING DOCUMENTS . . . . .	xvii
6. COLLECTIONS OF DOCUMENTS . . . . .	xvii
7. AGGREGATION WITH INDEPENDENT WORKS . . . . .	xviii
8. TRANSLATION . . . . .	xviii
9. TERMINATION . . . . .	xix
10. FUTURE REVISIONS OF THIS LICENSE . . . . .	xix
11. RELICENSING . . . . .	xx
ADDENDUM: How to use this License for your documents . . . . .	xx

# Abkürzungsverzeichnis

<b>ADL</b>	Advanced Distributed Learning
<b>AGPL</b>	Affero GNU General Public License
<b>AJAX</b>	Asynchronous JavaScript and XML
<b>API</b>	Application Programming Interface
<b>CAM</b>	Content Aggregation Model
<b>CBT</b>	Computer Based Training
<b>CRUD</b>	Create Read Update Delete
<b>DE</b>	Distance Education
<b>DRY</b>	Don't repeat yourself
<b>F2F</b>	Face-To-Face
<b>GFDL</b>	GNU Free Documentation License
<b>GPL</b>	GNU General Public License
<b>GUI</b>	Graphical User Interface
<b>HTML</b>	Hypertext Markup Language
<b>KISS</b>	Keep it simple stupid'
<b>LMS</b>	Learning Management System
<b>MVC</b>	Model View Control
<b>OE</b>	Online Education
<b>ORM</b>	Object Relationship Management
<b>PIF</b>	Package Interchange File
<b>REST</b>	Representational State Transfer
<b>RoR</b>	Ruby on Rails
<b>RTE</b>	Run Time Environment

<b>SCO</b>	Shared Content Object
<b>SCORM</b>	Shared Content Object Reference Model
<b>URI</b>	Uniform Resource Identifier
<b>WBT</b>	Web Based Training
<b>WYSIWYG</b>	What You See Is What You Get
<b>XML</b>	eXtensible Markup Language
<b>YAGNI</b>	You ain't gonna need it

# 1. Einleitung

Basierend auf der Studienarbeit „Analyse von Autorensystemen für ein WBT zu Vorlesungszwecken von Michael Gruben (Gruben, 2012) wird in dieser Studienarbeit ein System aus WBT<sup>1</sup>s geschaffen. Das Konzept für das Produkt des Projektes ist im Rahmen der Vorlesung „Gamification“ entstanden.

Dabei handelt es sich grundsätzlich um eine Blended Learning Plattform, die interessierten Lernenden eine zentrale Anlaufstelle bietet. Es werden also eLearning und persönliches Lernen miteinander kombiniert. Umrahmt und gamifiziert wird die Idee mithilfe des Dreyfus fünf Etappen Modells mentaler Aktivitäten. Die in dieser Studienarbeit verwendeten Bezeichnungen unterliegen gegebenenfalls weiteren Änderungen und sind für die deutschsprachige Version der Plattform bestimmt.

Inhalte der vorliegenden Studienarbeit sind Einblicke in die Entwicklung des ersten Prototyps. Dazu zeigt Kapitel II die Konzeption und damit die grundlegende Idee der Architektur. Daran anschließend wird in Kapitel 7 näher auf den tatsächlichen Entwurf eingegangen. Hier wird konkret auf Klassen und Methoden eingegangen, welche die Realisierung bestimmter Use-Cases zum Ziel haben. Kapitel 8 zeigt, wie der Entwurf letztlich realisiert wird. Hier sind auch erste Screenshots der Anwendung zu sehen. Um die vorrangegangenen Schritte zusammenzufassen und kurz auszuwerten, gibt Kapitel 9 einen Gesamtüberblick. Darauf aufbauend bietet Kapitel 10 eine Auswertung, die alle Aspekte des Projekts umfasst. Zusätzlich werden hier Ausblicke auf die weitere Verwendung des Projektergebnisses gegeben.

Am Ende des Projekts steht ein funktionierender Prototyp, der die wesentlichen Funktionen beherrscht. Weiterhin wird ein Konzept entwickelt worden sein, welches das Projekt an zentralen Stellen bekannt macht und so für eine rege Beteiligung sorgen soll. Mit der Namensgebung „Masterly Mate“<sup>2</sup> wurde bereits vor dem eigentlichen Projektstart ein wesentlicher Schritt zur Bekanntmachung getan.

---

<sup>1</sup> Web Based Training

<sup>2</sup> weitere Details zur Namensgebung in Abschnitt 4.1

# **Teil I.**

## **Vorbereitungen**

## 2. Projektplanung

Die Projektidee entstammt von studentischer Seite. In Abbildung 2.1 ist eine Wortwolke zu sehen, in der in Stichworten beschrieben ist, was sich unter Masterly Mate vorzustellen ist.



Abbildung 2.1.: Wortwolke über die „Masterly Mate“-Idee

Das Projekt nimmt sich eine Art Lernplattform zum Ziel, auf der Lernende auf Lehrende treffen sollen. Dabei entstehen Situationen, in der eine lehrende Person zu einer lernenden wird, und umgekehrt. Es sollen Diskussionen über bestimmte Fachgebiete stattfinden können und ideale Tutoren für bestimmte Fragestellungen gefunden werden. Da es nichts zu gewinnen gibt, engagiert sich jeder Teilnehmer freiwillig. Er erhält

Wissen und kann dieses im nächsten Moment an weitere interessierte Personen weitergeben, was sein eigenes Wissen erneut festigt. Lehrstunden sollen im gemütlichen Umfeld, wie Caffees oder Parks stattfinden. Masterly Mate bietet dazu eine regionale Suche an, mit deren Hilfe Lernende und Lehrende aufeinander treffen. Dem Duo steht es auch offen auf andere Kommunikationskanäle, wie Chat oder E-Mail, zu wechseln. Jeder Nutzer kann als Tutor für sein Fachgebiet oder seine Fachgebiete fungieren.

Um stets einen idealen Tutor zu finden, folgt die Idee dem Dreyfus fünf Etappen Modell mentaler Aktivitäten, welches in Abschnitt 3.1.3 näher beschrieben wird. So ist gewährleistet, dass ein Neuling die Inhalte von einer Person erklärt bekommt, die selbst noch im Lernprozess steckt und es können Inhalte, Tipps und Hinweise auf passendem Niveau ausgetauscht werden.

Letztlich soll das Ziel der Mitgliedschaft auf der Plattform nicht sein, der beste Guru eines Faches zu werden oder der beste Lehrer zu werden. Es geht darum Teil einer Bildungsgemeinschaft zu sein und sich gegenseitig engagiert zu unterstützen.

## **2.1. Motivation und Notwendigkeit**

Die Motivation zu dieser Idee entstand aus der interessanten und erwartungsvollen Kombination von eLearning und Gamification. Hinzu kommt die heute populäre Vorstellung von Blended Learning, wodurch das doch sehr trockene und eintönige Durcharbeiten von WBTs durch Lehreinheiten mit einem Tutor unterstützt wird.

Masterly Mate soll eine zentrale Anlaufstelle für diverse Weiterbildungs- und Lernangelegenheiten sein. Unabhängig davon, ob die Motivation privatem Interesse oder dem eigenen Bildungsweg entspringt, soll jeder Interessent wissen, dass die Plattform Antworten bietet.

Die Notwendigkeit resultiert aus der fehlenden Fähigkeit des Internets, Sachverhalte erläutern zu können. Heute ist es Usus beim Recherchieren das Internet zu gebrauchen, in dem rohe Daten und Informationen vorliegen. Wissen ist dort eher rar. Es gibt bisher nur wenige Plattformen, wie Wikipedia, die existieren, um Wissen zu publizieren, jedoch fehlt auch dort eine erklärende und erläuternde Komponente durch einen Menschen. Dieses Manko soll Masterly Mate ausgleichen.



## 2.2. Abgrenzung

Das Projektergebnis behauptet keinen Anspruch auf ein vollwertiges LMS<sup>1</sup>. Es fehlt die Komponente zur Organisation kompletter Lernpakete. In Masterly Mate stehen die WBTs unabhängig da. Sie sind allein Mittel zum Zweck als Beleg für die fachliche Kompetenz.

Weiterhin soll es Wikipedia nicht ersetzen. Das Projektergebnis bietet keine ausformulierten Texte oder Artikel zu Lerninhalten. Der Fokus liegt wesentlich stärker auf der Komponente Wissen zu vermitteln.

## 2.3. Zielsetzung

Da das Projekt insgesamt auf einen längeren Zeitraum angesetzt ist, lässt es sich nicht innerhalb der Bearbeitungszeit der vorliegenden Studienarbeit umsetzen. Aus diesem Grund sind die Ziele zum einen in operative und zum anderen in strategische zu unterteilen.

### 2.3.1. Operative Ziele

Wie in Kapitel 1 bereits angerissen wurde, soll zu Projektende ein funktionaler Prototyp stehen. Auch ist ein Konzept angedacht, welches der Bekanntmachung von Masterly Mate dient. Eventuell wird sich bis dahin eine kleine, lebendige Gemeinschaft gebildet haben, die die Plattform nutzt und um Inhalte erweitert. Die ersten Nutzer sollen automatisch unabhängig ihres didaktischen Grades<sup>2</sup> Autoren sein. So ist gewährleistet, dass Inhalte für neue Nutzer bereits existieren.

Das Projekt als Ganzes soll einen leichten Start haben. Sind die Ziele für die ersten Nutzer zu hoch gesteckt, resultiert aus der geringen Anzahl von Nutzern und der damit schwer erreichbaren nächsten Rängen Frustration und Unwille zur Nutzung der Plattform. Somit ist angedacht, die erforderliche Punktzahl für höhere Ränge (siehe Abschnitt 3.1.3) mit der Menge der Nutzer zu skalieren.

---

<sup>1</sup> Learning Management System

<sup>2</sup> näher erläutert in Abschnitt 5.2.2

### 2.3.2. Strategische Ziele

Längerfristig betrachtet soll eine rege und große Gemeinschaft entstehen, die Hilfsbereitschaft nicht scheut. Dazu wird bereits zu Beginn der Entwicklung eine Internationalisierung<sup>1</sup> berücksichtigt. Masterly Mate soll zur, in Abschnitt 2.1 beschriebenen, zentralen Anlaufstelle heranwachsen.

Dabei werden, um den Reiz am Lernen zu erhöhen, mit der Menge der Nutzer die Anforderungen für die jeweils nächsten Ränge erhöht. Denn je mehr Beteiligung die Plattform erfährt, desto wahrscheinlicher ist es, einen Tutor in der jeweiligen Region zu finden. Damit wird es auch immer einfacher, Punkte für den didaktischen Rang zu sammeln.

---

<sup>1</sup> siehe Abschnitt 7.2

## 3. Theoretische Grundlagen

Das Projekt basiert auf einigen theoretische Grundlagen. Dazu zählen neben neben Prinzipien und Lerntheorien einige technische Begriffe, deren Erläuterungen Inhalt dieses Kapitels sind. Dabei handelt es sich größtenteils um eine Kurzfassung der Beschreibungen aus (Gruben, 2012).

### 3.1. Prinzipien und Theorien

#### 3.1.1. Lernen

Das Lernen selbst wird heute als ein Prozess verstanden. Dabei wirken „mehrere zentrale psychologische Phänomene (Motivation, Emotion, Kognition)“ (Niegemann, 2004) zusammen.

Der Lernprozess besteht dabei aus drei Abschnitten:

1. Zunächst werden Eindrücke wahrgenommen. Dabei tragen neue oder vergessene Eindrücke zur Umstrukturierung im Gehirn bei.
2. Umstrukturieren bedeutet, dass Synapsen bewegt und andere Gehirnzellen angekoppelt werden.
3. Mit Wiederholungen wird Wissen persistiert. Es entstehen stabile Strukturen, welche einfach und schnell abrufbar sind.

In Abbildung 3.1 wird diese Umstrukturierung illustriert (Spitzer, 2012).

Von ein LMS wird erwartet, dass es diesen Lernprozess unterstützt. Anfänger sollen die Möglichkeit erhalten zunächst klein anzufangen und die Grundlagen eines bestimmten Sachverhaltes kennenzulernen. Fortschreitend können die Anforderungen und Herausforderungen gesteigert werden, um den Lernprozess zu unterstützten und zugleich die Motivation aufrecht zu erhalten.

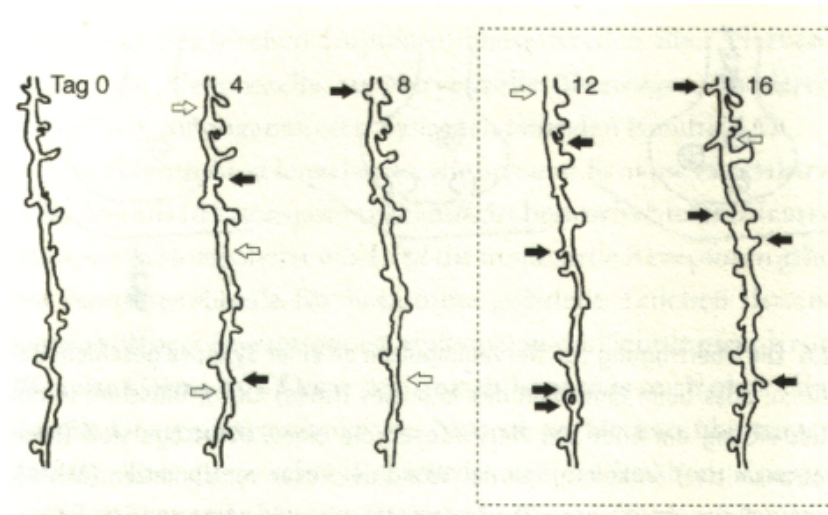


Abbildung 3.1.: Umstrukturierung von Synapsen <sup>1</sup>

### 3.1.2. Motivation

Die Motivation ist ein wesentliches Standbein des Lernprozesses. Fehlt sie, so ist es für Lernende bedeutend erschwert, Lerninhalte aufzunehmen, zu verarbeiten und zu verstehen. Mithilfe von Motivation wird ein charakteristisches Verhalten an den Tag gelegt, welches den Lernprozess aufrecht erhält (Jacobs und Preuße, 2010).

Im Mittelpunkt jeder Motivation steht stets das persönliche Glück (Stampfl, 2012). Dabei stehen Mittel zur Verfügung, die dem Lernenden auf unterschiedliche Weise unterstützen zu verstehen. Er kann zum einen intrinsisch und zum anderen extrinsisch motiviert werden.

#### Intrinsische Motivation

Die intrinsische Motivation wird auch als direkte Motivation bezeichnet. Damit wird der Lernende direkt angesprochen und in seinen Bedürfnissen befriedigt und seinen Wünschen wird unmittelbar nachgegangen. Ein intrinsisch motivierter Lernender geht einer Tätigkeit im eigenen Interesse nach, es sind keine externen Einflüsse nötig, die ihn zu seinen Handlungen erst bewegen müssen (Jacobs und Preuße, 2010).

<sup>1</sup> aus (Spitzer, 2012)

Jede Lernsoftware hat aus dem zuvor erwähnten Sachverhalt die intrinsische Motivation zum Ziel. Dazu werden nicht selten unter Anderem auch gamifizierende Inhalte verwendet (siehe 3.1.6).

## **Extrinsische Motivation**

Die andere Seite der Motivation kommt von aussen. Es werden Belohnungen gegeben oder Strafe und negative Konsequenzen vermieden. Synonyme sind demnach „indirekte Motivation“, das „Butterbrot-und-Peitsche-Prinzip“ oder „Manipulation“ (Jacobs und Preuße, 2010).

Für Masterly Mate im Speziellen, kann der Tutor ein extrinsisch Motivierender Faktor sein. Hinzu kommen die gamifizierenden Elemente der zu erreichenden Punktzahl pro WBT und das Aufsteigen in Rängen. Implizit wird auch Strafe in der Form angewandt, dass es laut Konzept auch möglich ist, im Rang zu fallen (siehe dazu Tabelle B.1).

### **3.1.3. Lernmodelle**

Zum Zweck der Unterstützung des Lernprozesses und Förderung der Motivation haben sich einige Lernmodelle herauskristallisiert, die heute als gültig und vertretbar angesehen werden. In der Idee von Masterly Mate sind explizit zwei Lernmodelle verwoben. Das Dreyfus fünf Etappen Modell mentaler Aktivitäten und Blended Learning.

#### **Das Dreyfus fünf Etappen Modell mentaler Aktivitäten**

Inhalt des Dreyfus-Modells ist das Hinterfragen, welche Person einer anderen einen bestimmten Sachverhalt erklären sollte. Dabei wird insbesondere berücksichtigt, wie groß der Unterschied der Fachkompetenz zwischen Lernenden und Lehrenden ist. Es wurden insgesamt fünf Ränge<sup>2</sup> definiert, die den Lernweg von abstrakten Prinzipien hin zu konkreter Erfahrung mit der Aneignung von Wissen beschreiben (Dreyfus und Dreyfus, 1980).

Allgemein formuliert sollte kein Experte einem Neuling etwas erklären. Steigt man neu in ein Fachgebiet ein, so sind zunächst simple und einfache Beispiele verbunden mit

---

<sup>2</sup> Novice, Competence, Proficiency, Expertise, Mastery

einem engen Betrachtungswinkel des Sachverhalts sehr hilfreich. Ein Experte würde den Neuling mit unnötigen Details überhäufen.

Dazu staffelt sich der Lernerfolg in fünf Etappen:

- 1. Novize** Ein Novize ist auf grundlegende Anweisungen angewiesen. Er verfügt über kein Vorwissen und evaluiert sich nicht selbst. Mit extrinsischem Feedback wird dem Abkommen vom Regelwerk zugekommen.
- 2. Fortgeschrittener** Die Handlungen des Fortgeschrittenen sind gegenüber dem Novizen weniger kontextfrei. Sein weiterer Lernweg kann auf seiner kleinen Wissensbasis aufbauen. Er hat grundlegende Prinzipien verstanden und erkennt situationsbasierte Muster. Der Fortgeschrittene kann simple Beispiele anhand von Guidelines durchlaufen, er experimentiert jedoch nicht.
- 3. Erfahrener** Der Umgang mit typischen Situationen am ihm gegebenen System stellen keine Hürden für den Erfahrenen dar. Ihm ist es möglich neue Situationen anzuknüpfen, einzuordnen und sehr ähnliche bewusst zu unterscheiden. Der Erfahrene arbeitet nach selbst erschaffenen Maximen.
- 4. Experte** Der Experte ist kein geeigneter Lehrer für einen Novizen mehr. Er hat die grundlegenden Prinzipien verloren und arbeitet nach seiner Intuition, einer Mischung aus Regeln, Guidelines und Maximen. Lösungen für ungewohnte Situationen gehören stets zum Repertoire des Experten. Im Sinne der fachlichen Kompetenz ist dieser Grad der höchste.
- 5. Meister** Ein Meister zeichnet sich gegenüber dem Experten neben fachlicher Kompetenz durch herausragende didaktische Fähigkeiten aus. Er bleibt damit auch ein geeigneter Lehrer für Novizen.

Generell sollte sich ein Lehrer zwei Grade über seinem Schüler befinden oder Meister sein. Weitere detailliertere Erklärungen zu den Rängen finden sich in (Gruben, 2012).

## Blended Learning

Zweck des Blended Learning, zu deutsch auch Integriertes Lernen genannt, ist das Verschmelzen der Vorteile diverser Lernformen. Darunter befinden sich F2F<sup>3</sup>-Education,

---

<sup>3</sup> Face-To-Face

DE<sup>4</sup> und OE<sup>5</sup> (eLearning). Die jeweiligen Nachteile wurden dabei weitestgehend überwunden (Kröger und Reisky, 2004).

Masterly Mate verfolgt die Verschmelzung von F2F-Education, der Durchführung von Präsenzunterricht, mit eLearning, dem Durcharbeiten von WBTs. Die DE wird dabei nur am Rande betrachtet, da nur in Ausnahmefällen Unterweisungen über Chats oder ähnliche Kommunikationskanäle vonstatten gehen sollen.

### **3.1.4. eLearning**

Mit dem eLearning wird im Gegensatz zum regulären Lernprozess ein zusätzlicher Mittler, eine elektronische Komponente, zwischen die rohen Informationen und dem lernenden Individuum eingeschoben. Heute ist beispielsweise ein Webbrowser ein Wiedergabemedium von Vielen, welches der Demonstration von Informationen dient (Baumgartner, Häfele und Maier-Häfele, 2002).

#### **Vorteile**

eLearning ist grundsätzlich unabhängig von physischen Gegebenheiten, mithilfe von Software lassen sich sämtliche, auch fiktive, Szenarien darstellen. Der Kreativität sind keine Grenzen gesetzt. Auch ist eine enorm vereinfachte Auswertung von Prüfungen und Tests möglich, da Computer zur Automatisierung von Prozessen geschaffen sind.

Für Masterly Mate bedeutet das, dass das Aufsteigen in höhere Ränge automatisiert vonstatten gehen kann. Beim Überschreiten bestimmter Schwellwerte für Punkte steigt der Lernende wie von selbst einen Rang auf. Es wird zudem möglich, Statistiken für den Nutzer anzufertigen, was das Prinzip des gamifizierens (siehe Abschnitt 3.1.6) zusätzlich unterstützt. Das Masterly Mate selbst ein digitales Produkt ist, spielt darüber hinaus der eigenen vereinfachten Verbreitung über nationale Grenzen hinweg stark zu.

---

<sup>4</sup> Distance Education

<sup>5</sup> Online Education

## Nachteile

Grundsätzlich ist der Unterschied zwischen Mensch und Maschine der größte Gegner von eLearning. Ein Computer kann heute nur recht spärlich auf die Bedürfnisse des Lernenden eingehen.

Expertensystemen wird beispielsweise nur eine beratende Funktion zugeteilt. Weitere Beispiele sind neuronale Netze, welche zwar Lösungen entwickeln können, jedoch muss deren Erarbeitung überwacht und hinterfragt werden (Keller, 2000).

Letztlich kann das Lernen am Computer heute nicht die Qualität bieten, die ein Lernender mit einer Lehrkraft erfährt. Seit 1989 entstehen die selben Diskussionen um den Einsatz von eLearning in Schulen (Thomé, 1989).

Im Konzept für Masterly Mate werden diese Nachteile berücksichtigt. Wie in Abschnitt 3.1.3 beschrieben, baut der Ansatz nicht allein auf eLearning. Die Nachteile sind erkannt und werden soweit möglich durch die Verbindung mit Präsenzveranstaltungen gemildert.

### 3.1.5. Flow

Der Flow ist ein Gefühl des völligen Aufgehens in einer Tätigkeit, bei der die Handlungsschritte als einheitliches Fließen von einem Augenblick zum nächsten erlebt wird. Einem alltäglichen entropischen Zustand<sup>6</sup> des Bewusstseins steht im Flow ein negentropischer Zustand gegenüber. Daher lässt sich annehmen, dass eine Person im Flow sich auf ihrem höchsten Leistungsniveau befindet (Csikszentmihalyi und Schiefele, 1993).

## Komponenten

Nach diversen Befragungen und Untersuchungen ergaben sich vier Komponenten, die einen Flow charakterisieren (Csikszentmihalyi und Schiefele, 1993).

**Verschmelzen von Handlung und Bewusstsein** führt dazu, dass sich die Person als Teil der Handlung sieht. So fühlt sich beispielsweise ein Kletterer als Teil des Felsens an dem er klettert.

---

<sup>6</sup> ungeordnet und zufällig



**Zentrierung der Aufmerksamkeit auf einen beschränkten Umweltausschnitt** sorgt dafür, dass von der Handlung unabhängige Reize kaum ins Bewusstsein gelangen. Die Konzentration liegt im Wesentlichen auf der Gegenwart, während die Zukunft und Vergangenheit verschwimmt.

**Selbstvergessenheit** ist die Eigenschaft, die die Person im Flow sich selbst als wahrgenommene Steuerungsinstanz weitestgehend vergessen lässt. Dabei rücken Selbstzweifel und Sorgen, sowie selbstwertsteigernde Kognitionen in den Hintergrund.

**Ausüben von Kontrolle über Handlung und Umwelt** lässt die Person die Handlung als kontrolliert wahrnehmen.

## **Bedingungen**

Flow entsteht nicht von allein. In der Forschung wurden zwei wichtige Bedingungen gefunden, die einen Flow begünstigen (Csikszentmihalyi und Schiefele, 1993).

**Passung von Fähigkeit und Anforderungen** erfordert ein Gleichgewicht von Leistungsfähigkeit des Handelnden und die Anforderungen der Tätigkeit. Es ist stets eine Gradwanderung zwischen Langeweile und Angst aus subjektiver Sicht des Handelnden. Trifft niedrige Anforderung auf niedrige Fähigkeit, so führt dies zu Apathie als Gegenspieler von Flow.

**Eindeutigkeit der Handlungsstruktur** definiert ein klares Ziel für die handelnde Person. Dazu gehört, dass keine langwierigen Überlegungen über die Anforderungen oder mögliche Teilziele notwendig sind. Eine eindeutige Struktur führt klare Handlungsanforderungen und -möglichkeiten auf und sorgt für eindeutige und widerspruchsfreie Rückmeldungen.

### **3.1.6. Gamification**

Ganz nach dem Claim „Fun is just another word for learning“ (Koster, c2005), werden heute mithilfe von Gamification ernste Inhalte mit spielerischen Elementen versehen. Damit sollen diese dank geförderter intrinsischer Motivation (siehe Abschnitt 3.1.2)

einfacher zu vermitteln sein. Als ein modernes Buzzword zu diesem Themengebiet sind heute „Serious Games“<sup>7</sup> zu nennen.

In Bezug auf eLearning ist die Aufgabe der Gamification den Nutzer an die Anwendung zu binden. Möglich wird dies durch das okkupieren von Aufmerksamkeit und Bestrebungen mithilfe positiver Eindrücke oder Belohnungen. Als konkrete Mittel zählen:

- Ziele, die den Flow (siehe Abschnitt 3.1.5) des Nutzers unterstützen,
- regelmäßiges Feedback,
- eine Messung des Fortschritts,
- Belohnung des Aufwandes und der Planerfüllung, nicht nur des Erfolgs und
- Motivation von gleichgestellten Nutzern

Darüber hinaus kann ein Alleinstellungsmerkmal konzipiert und eine ansprechende Darstellung entworfen werden. So wird die Anwendung als exotisch angesehen, was bei einer stimmigen Menge an besonderen Eigenschaften die Motivation zur Nutzung fördert (Raymer, 2011). Im Gegenzug ist es eher nachteilig, wenn die Anwendung überladen oder aufdringlich wirkt. Es muss also ein gesundes Mittel gefunden werden.

Masterly Mate macht sich die motivierende Wirkung von Gamification zunutze. Es werden Fortschrittsbalken und Statistiken integriert. Zusätzlich erhält er mit höheren fachlichen Level mehr Möglichkeiten zur Gestaltung seines Profils oder als Tutor für Gleichgesinnte. Wie die Idee der Gamification in Masterly Mate konkret konzeptioniert wird, ist Inhalt von Abschnitt 6.1.

### **3.1.7. freie Lizenzen**

Das reguläre Urheberrecht bietet Autoren wenig Freiheit in Bezug auf die Freigabe ihrer Werke. Daher entstand aus der Idee der freien Software das Kreieren einer Lizenz, die dies ermöglicht.

Drei Faktoren zeichnen eine freie Lizenz aus. Autoren erhalten „ein leicht handhabbares und frei verfügbares Instrument“ (Kreutzer, 2011). Damit können sie ihre Werke mehr oder weniger frei verbreiten. Nutzer erhalten „sehr weit gehende Freiheiten der

---

<sup>7</sup> mitunter populäre Spiele, wie Assassins Creed, in denen Lerninhalte im Spielkontext verwoben sind (Breitlauch, 2013)

hierunter stehenden Werke“(Kreutzer, 2011). Darüber hinaus „sind die Rechte und Pflichten, die durch die Open-Content-Lizenzen aufgestellt werden, im Normalfall transparent und leicht zu verstehen“(Kreutzer, 2011).

Demnach fasst eine freie Lizenz analog zum Urheberrecht Spielregeln im Umgang mit urheberrechtlich geschütztem Material dar. Der Unterschied bei der freien Lizenz liegt dennoch in einer einfacheren Handhabbarkeit und der Übergabe von Rechten an Nutzer.

Das Projekt zur vorliegenden Studienarbeit bedient sich den Eigenschaften einer freien Lizenz, um es weiter wachsen zu lassen. Mit der freien Verbreitung soll sich das Konzept und die Qualität von Masterly Mate mit der Beteiligung interessierter Personen weiter fortentwickeln.

## **3.2. Konzepte und Implementierungen**

### **3.2.1. Learning Management System**

Ein LMS unterstützt das selbstgesteuerte Lernen. Ein Nutzer arbeitet sich, möglichst intrinsisch motiviert, durch die ihm dort gebotenen Inhalte (Wendt, 2003).

Wie in der Einleitung beschrieben ist Masterly Mate selbst ein LMS. Der Lernende entscheidet sich selbst für ein Themengebiet (Topic), welches ihn interessiert. Dazu findet er WBTs, die ihm bestimmte Sachverhalte auf seinem fachlichen Niveau näher bringen. Masterly Mate geht mit der Vermittlung von Tutoren über die eigentliche Definition des LMS hinaus, was es von existierenden OpenSource-LMS, wie Moodle abgrenzt.

### **3.2.2. Autorenwerkzeug**

Masterly Mate erfordert das Einbinden von WBTs. Autorenwerkzeuge als WYSIWYG<sup>8</sup>-Editoren dienen deren Erstellung und machen laut Definition das Einbringen von Multimedia möglich (Niegemann, 2004).

---

<sup>8</sup> What You See Is What You Get

In vielen LMS sind heute simple Autorenwerkzeuge integriert. Externe Lösungen hingegen lassen sich je nach ihren Möglichkeiten und der Handhabung in professionelle Autorensysteme, WYSIWYG-Editoren und Rapid Content Development klassifizieren (Niegemann, 2004). Masterly Mate unterstützt mit der Einbindung von SCORM (siehe Abschnitt 3.2.4) alle Varianten und bietet demgegenüber kein eigenes Werkzeug zum Erstellen von Inhalten an.

### 3.2.3. WBT

Beim Begriff des WBTs handelt es sich um eine Software, die Web-Technologien nutzt, um eLearning zu realisieren. Blickt man tiefer in die Definition, so gehen die Meinungen heute auseinander. Eine Variante ist eine Erklärung von Peter Baumgartner: „WBT umfasst die internetgestützte Form des Fernlernens mit und ohne Betreuung durch Tutoren“ (Baumgartner et al., 2002). Dem hinzuzufügen ist, dass Web-Applikationen im Allgemeinen auch ohne eine Anbindung an das Internet, beispielsweise lokal oder in einem kleinen Firmennetzwerk, brauchbar sind.

In einer weiteren Definition taucht das WBT in einer Klassifikation zwischen virtuellem Klassenzimmer und CBT<sup>9</sup> auf. Es kann damit auf eine tutorielle Betreuung verzichten und ist stark an die Verfügbarkeit in einem Computernetzwerk gebunden (Schleifer, 2003).

Historisch ist WBT aus DE, „Computer-conveyed education“ und diversen Internet Technologien entstanden, deren Technologien, Traditionen und Techniken den Grundstein bilden. Daraus wurden bei der Konzeption von WBT die Vorteile extrahiert und die Nachteile versucht zu vermeiden (Horton, 2000).

### 3.2.4. SCORM

SCORM<sup>10</sup> ist eine Entwicklung der ADL<sup>11</sup>-Initiative. Es soll möglich sein, auf einfache Art und Weise Trainingseinheiten in LMS einzubinden. Zweck von SCORM und der damit in Verbindung stehenden API<sup>12</sup> ist demnach das Schaffen einer Ebene zwischen einem WBT und einem LMS.

---

<sup>9</sup> Computer Based Training

<sup>10</sup> Shared Content Object Reference Model

<sup>11</sup> Advanced Distributed Learning

<sup>12</sup> Application Programming Interface

## Aufbau

Dazu ist SCORM zunächst als ein Paket zu verstehen. Für gewöhnlich erfolgt die Einbindung in das LMS durch einen Import als ZIP-Datei, welche als Container fungiert. In Abbildung 3.2 ist die Hierarchie von Verschachtelungen von links nach rechts dargestellt.

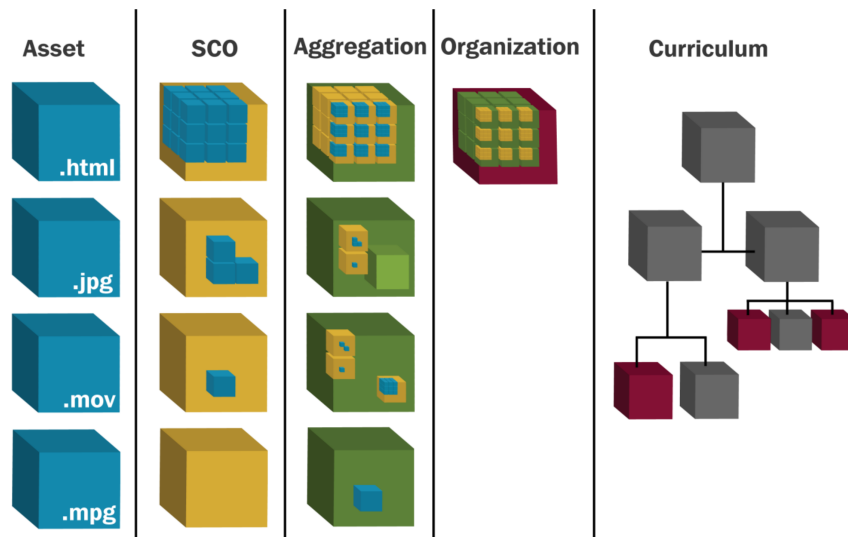


Abbildung 3.2.: Komponenten eines SCORM-Paketes<sup>13</sup>

**Asset** Assets sind elektronische Ressourcen für die Verwendung in SCOs. Beispiele sind Medien, Texte, Bilder, Klänge und Töne, HTML<sup>14</sup>-Seiten, Assessment Objekte und andere elementare Teile. Sie kommunizieren nicht direkt mit dem LMS und werden in einem SCO nur verlinkt, nicht direkt eingebunden. Assets befinden sich quasi in einem Lager, in dem Ressourcen für den Einsatz in einem SCO bereit liegen. Aufgrund der Verlinkungen hat das Editieren eines Assets Auswirkung auf sämtliche Stellen, an denen es eingesetzt ist (ADL, 2011).

**SCO** Ein SCO<sup>15</sup> ist die kleinste logische Einheit von SCORM. Aus Sicht der Designer von Lerneinheiten beinhalten SCOs das eigentliche lehrreiche Material. Programmierer würden sie eher als Web-Applikation sehen, welche über Schnittstellen für die Kommunikation mit einem LMS verfügt (ADL, 2011).

<sup>13</sup> aus (ADL, 2011)

<sup>14</sup> Hypertext Markup Language

<sup>15</sup> Shared Content Object

**Aggregation** Eine Aggregation oder ein Cluster ist ein Verbund von zusammengehörigen Aktivitäten. Sie kann SCOs oder andere Aggregationen enthalten. Eine Aggregation ist keine physische Datei, vielmehr ist sie eine Struktur mit der die Planung einer Reihenfolge zur Abarbeitung von SCOs und Aggregationen möglich wird. In der SCORM Manifest Datei kann eine Aggregation zusammengesetzt werden (ADL, 2011).

**Organization** Jede SCORM-Datei enthält eine Organisation. Diese hat gegenüber Aggregationen keine inhaltlichen Unterschiede. Organisationen zeichnen sich dadurch aus, dass sie die Wurzel eines jeden SCORM-Paketes bilden, sie enthalten sämtliche Regeln zur Abarbeitung des WBTs. Daher werden sie auch Root-Aggregation genannt. (ADL, 2011)

**Curriculum** Curricula gehen über die Definition von SCORM hinaus und gehören demnach nicht mehr zum Standard. Sie werden eher von einem LMS zusammengestellt und überwacht (ADL, 2011). Ohne SCORM wäre die Zusammenstellung Curricula jedoch stark erschwert, es mangelte an einer Schnittstelle, welche die Kommunikation zwischen WBT und LMS bereitstellt und damit einer Komponente zur Auswertung der Etappen.

## Manifest-Datei

Jedes SCORM-Paket ist laut Standard dazu angehalten eine Manifest-Datei mit dem namen *imsmanifest.xml* zu enthalten. Darin sind wesentliche Informationen der Lerninhalte für das LMS enthalten. Es kommuniziert welche Inhalte wann, wie eingebunden werden sollen. Das Schema der SCORM-Manifest-Datei ist in Abbildung 3.3 abgebildet.

Die Metadaten am Kopf der Datei enthalten zusätzliche Informationen zum WBT, die nach dem CAM<sup>16</sup> aufgebaut sind. Da die Unterschiede zwischen SCORM-Versionen sehr groß ausfallen, wird im Header neben dem eigentlichen Schema<sup>17</sup> weiterhin die Version mit angegeben<sup>18</sup>, um Kompatibilität zu gewährleisten.

---

<sup>16</sup> Content Aggregation Model

<sup>17</sup> meistens „ADL SCORM“

<sup>18</sup> zum Beispiel „2004 4th Edition“

<sup>19</sup> aus (ADL, 2011)

Darunter finden sich die im vorigen Abschnitt erläuterten Komponenten eines SCORM-Paktes wieder. Unter dem Tag *organizations* wird die Root-Aggregation mit ihren Regelungen und der Reihenfolge der SCOs aufgeführt. Ein Item steht dabei jeweils für ein SCO. Ursprünglich war die Unterstützung mehrerer Organisationen geplant, heute wird jedoch nur eine einzige unterstützt.

Unter *resources* werden sämtliche Assets aufgezählt, die ein SCO oder Asset benötigt. Eine Ressource ist als eine Gruppierung von Assets zu verstehen. So werden nur die Assets geladen, die für das aktuelle Modul benötigt werden. Der Identifier kann in einer Organization oder in einer anderen Ressource als Abhängigkeit aufgenommen werden.

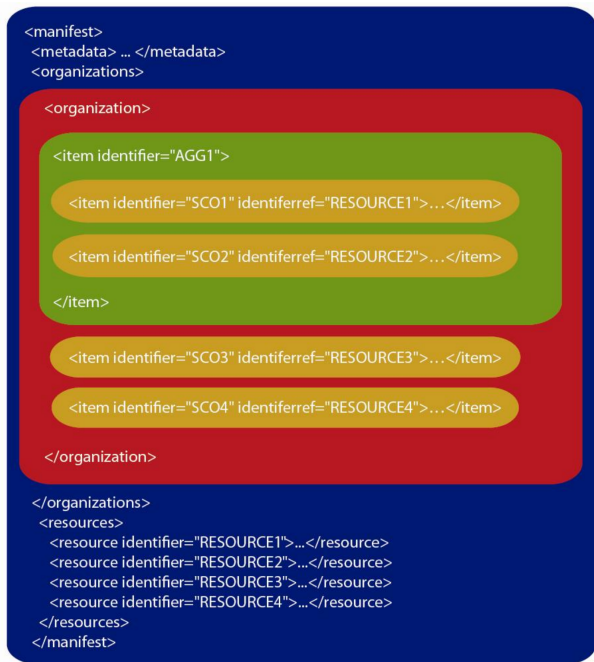


Abbildung 3.3.: Schema der SCORM-Manifest Datei<sup>19</sup>

## SCORM-API

Die SCORM-API ist notwendig, um die Kommunikation zwischen LMS und WBT zu vereinheitlichen. So können WBTs eingebunden werden, die aus dem Autorenwerkzeug als SCORM-Datei exportiert wurden. Abbildung 3.4 zeigt, wie die SCORM-API zwischen dem LMS und dem WBT steht. Der Client verfügt über einen Web-Browser zum betrachten und Durcharbeiten des WBTs, der in einer Aggregation angeordneten SCOs. Das LMS stellt als Server die WBTs bereit. Es verfügt über Nutzerinformationen und Informationen über Lerninhalte. Daneben bietet es dem Lernenden einen Weg für das Erlangen von Wissen in einem Fachgebiet – ein Curriculum – an.

Technisch realisiert ist die SCORM-API mit JavaScript, einer populären Programmiersprache für Webanwendungen auf Clientseite. Sie besteht aus zwei Teilen, einen auf der Client- und einen auf der Serverseite.

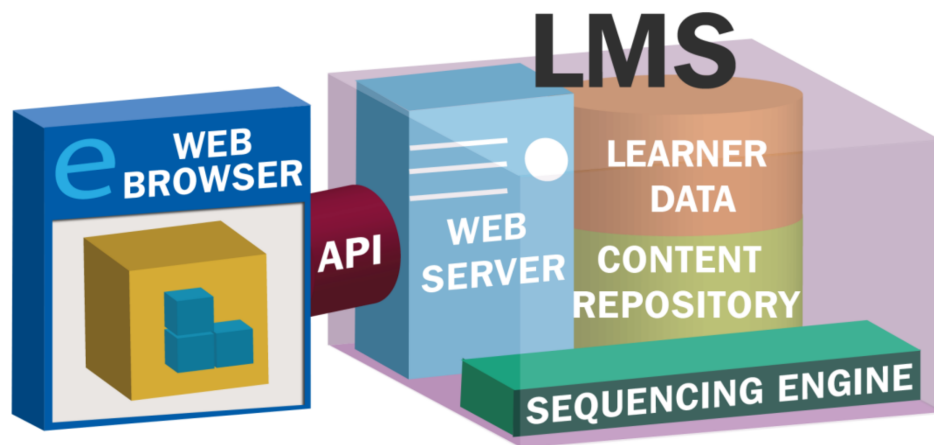


Abbildung 3.4.: SCORM-API Anbindung<sup>20</sup>

**API-Wrapper** Der Client bindet für jedes SCO einen API-Wrapper ein, der die Zugehörigkeit zu einem SCORM-Paket sichert. Dieser enthält darüber hinaus alle Funktionen, die für die Kommunikation der Inhalte mit dem LMS erforderlich sind. Ein SCO muss dabei mindestens zwei Aufrufe tätigen. Die Funktion *doInitialize()* initiiert die Verbindung zwischen dem LMS und dem SCO, *doTerminate()* hingegen trennt die Verbindung noch vor dem Schließen eines SCO.

**SCORM-RTE** Im Verantwortungsgebiet für das RTE<sup>21</sup> von SCORM liegt das Starten von Inhaltsobjekten, das Herstellen einer Kommunikation zwischen LMS und SCO und das Verwalten von Informationen über den Lernfortschritt eines Anwenders. Das RTE definiert ein Modell, welches seine Arbeit aufnimmt wenn ein spezifisches Inhaltsobjekt zum starten identifiziert wird (ADL, 2009).

### 3.2.5. Gestaltungskonzepte

Da die Arbeit mit Maschinen nicht dem natürlichen Verhalten von Menschen entspricht, ist es notwendig Oberflächen von Software zu gestalten, welche von Menschen verstanden und benutzt werden können. Es ist dabei besonders wichtig die Menschen, welche mit der Software arbeiten nicht mit zu vielen Informationen oder 4kompliziertem Verhalten zu überfordern und somit zu frustrieren. Daher haben sich im Laufe der Zeit

<sup>20</sup> aus (ADL, 2011)

<sup>21</sup> Run Time Environment



einige Gestaltungskonzepte herausgebildet auf die im Folgenden eingegangen werden soll.

## **Die Sinne des Menschen**

Der Mensch verfügt über ein breite Spektrum an Sinnesorganen. Mit diesen kann er sich in seiner Umwelt zurechtfinden um mit ihr interagieren zu können. Im Bezug auf die Arbeit mit Software wird jedoch meist nur ein Teil der zur Verfügung stehenden Sinne genutzt. Zu diesen zählt neben dem Hören vor allem das Sehen, da heutzutage die Kommunikation der Maschine zum Menschen über einen Bildschirm realisiert wird.

Wie bereits erwähnt erhält der Mensch die Informationen, welche eine Maschine ihm zur Verfügung stellt meist über seinen Visuellen Sinn. Damit Menschen diese Informationen ohne größere Schwierigkeiten verarbeiten können, ist es wichtig die einzelnen Objekte sinnvoll anzuordnen. Hierfür gibt es einige Grundsätze.

### **Gesetz der Nähe**

Das Menschliche Gehirn erkennt Gruppen von Objekten bereits ohne ihre eigentlichen Eigenschaften zu kennen. Dies liegt daran, dass Objekte welche nahe beieinander liegen vom Gehirn als zusammengehörend identifiziert. Es leuchtet daher ein, dass eine falsche Anordnung von Objekten dem Gehirn falsche Zugehörigkeiten vorgaukelt. Der Mensch wird daher Objekte die logisch zusammengehören aber falsch positioniert wurden auch an den falschen Stellen suchen. Würde man z.B. in einer Benutzeroberfläche, die Eingabe von Vor- und Nachname an weit voneinander entfernte Positionen schreiben, so würden die Benutzer eines der beiden Felder erst spät oder möglicherweise gar nicht finden. Dies würde die Arbeit mit der Software erschweren und sie zäh und ineffizient machen.

### **Gesetz der Gleichheit**

Dieses Gesetz wirkt nicht so stark, wie das Gesetz der Nähe. Dennoch sollte es nicht außer acht gelassen werden. Das Gehirn gruppiert neben nah benachbarten Objekten auch gleichförmige und Gleichartige. So erkennt das Gehirn, dass Objekte gleiche Farbe in einem direkten Zusammenhang stehen müssen und wird sie daher entsprechend

logisch verknüpfen. Bei der Farb-/Formgebung ist daher darauf zu achten. Die Verwendung von Farbe sollte ohnehin nur mit Maß und Ziel umgesetzt werden. Darauf wird später näher eingegangen werden.

## **Software Ergonomie**

Es gibt einige Grundsätze im Bezug auf die Ergonomie von Software. Die folgenden sind unter anderem Teil der ISO 9241.

**Aufgabenangemessenheit (ISO)** Es ist wichtig, dass eine Benutzeroberfläche so gestaltet ist, dass die Benutzer ihre Aufgaben effektiv und effizient erledigen können. D.h. die Software muss ihnen die benötigten Werkzeuge an die Hand geben um die Aufgaben zu erledigen und obendrein müssen diese auch so einfach und intuitiv zu bedienen sein, dass diese Erledigung in einem zeitlich Vertretbaren Rahmen passieren kann. Die Software muss daher den Benutzer auf sinnvolle Art und Weise unterstützen. Beispielen für eine solche Unterstützung wäre die Verwendung von Standardwerten in Textfeldern und/oder Drop-Down-Menüs.

**Selbstbeschreibungsfähigkeit (ISO)** Damit ein Benutzer die Software gut verstehen und damit auch gut bedienen kann, muss die Software in der Lage sein, ihre Funktionsweise selbst zu beschreiben. Der Benutzer sollte in die Lage versetzt werden, dass er bei jeder ausführbaren Funktion genau weiß, was diese bewirkt.

**Steuerbarkeit (ISO)** Im Innern fast eines jeden Menschen befindet sich der Wunsch Macht zu haben und bestenfalls auch auszuüben. Machtlosigkeit hingegen findet der Mensch belastend. Daher sollte eine Software einem Benutzer nicht das Gefühl geben, keine Kontrolle über sie zu haben. Der Benutzer muss in die Lage versetzt werden, jeden Schritt, den er mit der Software unternimmt wieder rückgängig machen oder gar vollständig abbrechen zu können.

**Erwartungskonformität (ISO)** Der Mensch hängt, so anpassungsfähig er auch sein mag, Gewohnheiten nach. Daher sollte eine Software ihn nicht verwirren indem sie altbekannte und weitgehend allgemeingültige Muster verändert. Dazu zählen z.B. die

Menus "Datei" oder "Bearbeiten". Gut funktionierendes und bekanntes sollte nicht durch experimentelles ersetzt werden.

**Fluchlinien** Fluchlinien sind keine direkt sichtbaren Linien, sondern treten implizit durch die Anordnung von Objekten auf der Benutzeroberfläche auf. Prinzipiell kann man sagen, dass eine Oberfläche mit sehr vielen Oberflächen unordentlicher und unübersichtlicher wirkt. Daher sollte eine Benutzeroberfläche möglichst wenige Fluchlinien aufweisen. Fluchlinien entstehen durch die Außenkanten von Objekten. Dies bedeutet, dass eine Anordnung bei denen die Kanten der Objekte auf der gleichen Linien liegen, wenige Fluchlinien erzeugen.

**Schrift** Die Schrift sollte angemessen und gut lesbar sein. Bei langen Texten strengt eine serifenlose Schrift das Auge an. Bei Benutzeroberflächen hingegen wirken diese angemessener. Außerdem ist es wichtig, dass die Schrift gut lesbar ist.

**Farben** Farben sind grundsätzlich sparsam einzusetzen. Außerdem ist beim Einsatz der Farben zu beachten, dass diese eine psychologische Wirkung haben. So wird rot in aller Regel mit Gefahr assoziiert. Man sollte diese Farbe daher auch dafür einsetzen. Des Weiteren ist zu vermeiden Grundfarben nebeneinander zu verwenden.

## **GUI-Bloopers**

Ein Blooper bezeichnet einen Missgriff, in diesem Abschnitt geht es daher um Missgriffe bei der Gestaltung von Benutzeroberflächen. Einige werden im folgenden vorgestellt.

**Dynamische Menus** Menus die im Verlauf der Arbeit mit der Software ihren Inhalt ändern. Dies mag auf den ersten sehr innovativ wirken, verwirrt den Benutzer aber nur unnötig.

**Checkboxes und Radiobuttons falsch einsetzen** Checkboxes sollten nur verwendet werden, wenn es sich bei der abgebildeten Funktionalität tatsächlich um eine An-/Auswahl handelt.

## 3.2.6. Autorisierung und Authentifikation

Uebernimmt Julian?

## 3.2.7. Ruby on Rails

RoR<sup>22</sup> ist ein in der Programmiersprache Ruby programmiertes Framework für die einfache Entwicklung von Web-Applikationen. Das Design folgt Annahmen über grundlegende Voraussetzungen, welche jeder Programmierer zu Beginn benötigt. Darüber hinaus ist weniger Programmcode zum Erreichen von Zielen erforderlich (RailsGuides, 2013).

### Handhabung

Das Erstellen einer RoR-Applikation folgt einem recht eigensinnigen Weg. Ein „bester“ Weg wird als Annahme vom Framework unterstützt. Alternativen führen demgegenüber gelegentlich zum Gegenteil und die Implementierung wird erschwert. Ein Programmierer ist daher angehalten sich den „Rails Way“ anzunehmen, um eine beschleunigte Produktivität im Gegensatz zu weniger produktiven Ergebnissen zu erreichen. Die drei folgenden Prinzipien sind in RoR fest eingewoben und sollten verinnerlicht werden (RailsGuides, 2013):

**DRY** „Don’t Repeat Yourself“ empfiehlt es, den selben Programmcode nicht immer wieder erneut zu schreiben. Eine durchdachte Architektur macht Objekte und Methoden leicht wiederverwendbar.

**Convention over Configuration** RoR macht annahmen darüber, wie Ziele erreicht werden wollen und beugt so das schier endlose Durcharbeiten von Konfigurationsdateien vor.

**REST** mit dem REST<sup>23</sup>-Muster (siehe Abschnitt 3.2.8), mit dem eine Web-Applikation aus Ressourcen und standard HTML gefertigt wird, macht diese performanter

**MVC** wird in Abschnitt 3.2.9 näher erläutert

---

<sup>22</sup> Ruby on Rails

<sup>23</sup> Representational State Transfer

## Komponenten

RoR selbst ist sehr modular aufgebaut, um die gewünschten Anforderungen bestmöglich erfüllen zu können. Die einzelnen Komponenten werden „gems“ genannt. Jeder Interessierte kann nach belieben eine neue gem schaffen oder eine vorhandene verbessern. Tabelle 3.1 zeigt die gems aus denen RoR selbst besteht (RailsGuides, 2013).

Tabelle 3.1.: Ruby on Rails Komponenten<sup>24</sup>

<b>Action Pack</b>	<p>Dieses gem realisiert das VC in MVC. Es besteht aus Action Controller, Action Dispatch und Action View.</p> <p>Der <b>Action Controller</b> verwaltet die Controller in einer Rails Applikation. Das zugehörige Framework vermittelt eingehende Anfragen zu einer Rails Applikation, extrahiert Parameter und sendet sie zur vorgesehenen Action. Die bereitgestellten Dienste des Action Controllers umfassen das Sitzungsmanagement, das rendern von Templates und die Verwaltung von Umleitungen.</p> <p><b>Action View</b> Demgegenüber ist Action View verantwortlich für die Verwaltung der Views der Applikation. Es ist standardmäßig in der Lage HTML und XML<sup>25</sup> zugleich auszugeben. Es rendert Templates und inkludiert eingebettete Templateteile<sup>26</sup> und die eingebaute AJAX<sup>27</sup>-Unterstützung.</p> <p><b>Action Dispatch</b> verwaltet das Routen von Web-Anfragen und leitet diese beliebig an die eigene oder eine andere Applikation weiter.</p>
<b>Active Mailer</b>	<p>Dieses Framework stellt E-Mail Dienste her. Eingehende E-Mails können verarbeitet werden oder neue E-Mails können als einfacher Text oder als mehrteilige Dokumente aus flexiblen Templates generiert und schließlich gesendet werden.</p>
<b>Active Model</b>	<p>Active Model stellt ein definiertes Interface zwischen den Action Pack Diensten und Mapping von Objektbeziehungen bereit. Dieses gem erlaubt RoR die Verwendung eines alternativen ORM<sup>28</sup>-Frameworks, falls dies für die Applikation benötigt wird.</p>
Fortsetzung auf der nächsten Seite ...	

Tabelle 3.1 (Fortsetzung)

<b>Active Record</b>	Die Basis für Models in RoR ist Active Record. Es bietet unter Anderem Datenbankunabhängigkeit, grundlegende CRUD <sup>29</sup> -Funktionalität, erweiterte Suchmöglichkeiten und die Fähigkeit Relationen zwischen Models herzustellen.
<b>Active Resource</b>	Die Verwaltung der Verbindung zwischen Objekten und RESTful Webservices stellt Active Resource bereit. Es implementiert einen Weg webbasierte Ressourcen an lokale Objekte mittels CRUD Semantiken zu binden.
<b>Active Support</b>	Active Support ist eine umfangreiche Sammlung von Werkzeugklassen und standard Ruby Bibliothekserweiterungen, welche in RoR Verwendung finden.
<b>Railties</b>	Railties ist der Kerncode von RoR, welcher neue Applikationen baut und die aufgeführten Frameworks und Plugins zu einer allumfassenden Applikation verbindet.

### 3.2.8. REST

REST ist die Basis für eine von Roy Fielding entworfene RESTful Architektur (Fielding, 2000). Demnach besteht REST in Bezug auf RoR im Wesentlichen aus zwei Prinzipien: Es werden zum einen Identifizierungen für Ressourcen analog URI<sup>30</sup>s verwendet, um Ressourcen zu repräsentieren. Zum anderen werden Repräsentationen des Status der Ressource zwischen Systemkomponenten transferiert.

Beispielsweise wird unter dem HTTP Request *DELETE /photos/17* verstanden, dass zu einer Ressource Photo mit der ID 17 verwiesen werden soll. Zusätzlich wird eine gewünschte Aktion indiziert, die das Löschen der Ressource veranlasst. REST ist damit

<sup>24</sup> aus ((RailsGuides, 2013))

<sup>25</sup> eXtensible Markup Language

<sup>26</sup> genannt „partials“

<sup>27</sup> Asynchronous JavaScript and XML

<sup>28</sup> Object Relationship Management

<sup>29</sup> Create Read Update Delete

<sup>30</sup> Uniform Resource Identifier

ein recht natürlicher Stil für die Architektur von Webapplikationen. RoR greift REST auf und versteckt gleichzeitig einige REST-Komplexitäten und Browser Eigenarten (RailsGuides, 2013).

### 3.2.9. MVC Architektur

RoR baut auf der MVC<sup>31</sup>-Architektur, dessen Komponenten und Eigenschaften in Tabelle 3.2 zu sehen sind, auf. Damit ist es möglich, die Programmcodes für Programmlogik und Nutzerinterface sauber zu trennen. So ist es einfacher Programmteile zuzuordnen, was eine einfachere Wartung garantiert. Darüber hinaus kann so das DRY<sup>32</sup>-Prinzip sehr einfach umgesetzt werden (RailsGuides, 2013).

Tabelle 3.2.: Komponenten der MVC-Architektur und deren Eigenschaften<sup>33</sup>

<b>Model</b>	Ein Model repräsentiert die Informationen (Daten) der Applikation und die Regeln zu deren Manipulation. In RoR werden Models primär für die Handhabung von Regeln zur Interaktion mit der korrespondierenden Datenbanktabelle verwendet. Üblicherweise wird ein Model auf eine Datenbanktabelle abgebildet. Der Großteil der Programmlogik wird in Models abgebildet.
<b>View</b>	Views repräsentieren das Nutzerinterface der Applikation und stellen Informationen für den Web-Browser oder andere Werkzeuge bereit, welche Anfragen an die Applikation stellen. In RoR sind dies meistens HTML-Dateien mit eingebettetem Ruby-Programmcode, der allein für das Darlegen von Informationen zuständig ist.
<b>Controller</b>	Controller sind quasi der Kleber zwischen Model und View. In RoR nehmen die Controller die Anfragen an den Webserver entgegen, sie fragen Daten aus der Datenbank bei den Models ab und übergeben diese an die Views zur Präsentation.

---

<sup>31</sup> Model View Control

<sup>32</sup> Don't repeat yourself

<sup>33</sup> aus (RailsGuides, 2013)

## **Teil II.**

# **Konzeption der Plattform**



## 4. Das zugrundeliegende Prinzip

Ziel des Systems ist die Vermittlung von Lerninhalten in einer sich gegenseitig unterstützenden Gemeinschaft. Zu diesem Zweck folgt das Konzept einer Art Mischung aus Lern- und Datingplattform – es werden Lerninhalte bereitgestellt, zu denen Tutoren vermittelt werden.

### 4.1. Namensgebung

Der Name „Masterly Mate“ entstand aus der Bezeichnung des höchsten Rangs im Dreyfus-Modell (siehe Abschnitt 3.1.3) und dem Namen eines beliebten Getränks im Informatikerkreis, beziehungsweise dem englischen Begriff für Kumpel oder Kamerad.

So lässt sich der Name frei als meisterlicher Kamerad übersetzen, was die erwünschte offene und freundliche Kommunikation auf der Plattform ausdrücken soll.

### 4.2. Freie Software

Um Offenheit gleich bei der Entwicklung zu berücksichtigen, wird Masterly Mate unter einer freien Lizenz, wie sie in Abschnitt 3.1.7 beschrieben ist, veröffentlicht werden. Damit kann jeder interessierte den Quelltext einsehen und bei Bedarf selbst Hand anlegen, um Funktionalitäten zu verbessern oder neue hinzuzufügen. So gibt es auch keine Falltüren im Sinne von ungewünscht übermittelten und verwendeten Informationen und damit fehlender Transparenz, wie beispielsweise bei Facebook oder Google.

Auch das vorliegende Dokument wird unter einer freien Lizenz, der GFDL<sup>1</sup> zur Verfügung gestellt. So befindet sich nach der Eigenständigkeitserklärung ein Lizenzhinweis.

---

<sup>1</sup> GNU Free Documentation License

Nach den Vorgaben wird im Anhang ab Seite xi die komplette Lizenz aufgeführt. Daraus folgt, dass jeder Interessierte das Projekt und dessen Ursprünge verfolgen kann. Auch bietet das Dokument Einblicke in Ideen für weitere Versionen in Abschnitt 10.1 und anschließende Vorhaben in Abschnitt 10.2.

Hier noch viel mehr beschreiben!

Ein Anwender, der eine fachliche Herausforderung sucht oder sich in einem Fach weiterbilden möchte, wird sich dem Bearbeiten von WBTs widmen.

, indem er WBTs durcharbeitet und mit Bestehen der darin enthaltenen Quizes Punkte für seinen fachlichen Rang sammelt. Unter Umständen nimmer er Hilfe von einem Tutor in Anspruch

Er erhält gegebenenfalls eine gute Bewertung und bessert damit seinen didaktischen Rang auf.

# 5. Architektur

## 5.1. Erweiterbarkeit

Bei der Konzeption wird insbesondere darauf geachtet, dass die Anwendung einfach erweiterbar ist und den Ansprüchen vielseitiger Szenarien genügt. So wird beispielsweise auf die Unterstützung von WBTs aus einem bestimmten Autorenwerkzeug verzichtet. Viel mehr steht das Einbinden von SCORM auf dem Plan. Darüber hinaus berücksichtigt das Klassenmodell (siehe Abschnitt 7.3) eine Many-to-Many Beziehung zwischen Themen und WBTs. So ist gewährleistet, dass es allgemeine WBTs geben kann, die mehreren Themen zugehörig sind.

## 5.2. Aufbereiten des Dreyfus-Modells

Für das Produkt der vorliegenden Studienarbeit wird das in Abschnitt 3.1.3 erläuterte Dreyfus-Modell angepasst. So ergeben sich vier fachliche Ränge. Hinzu kommen Ränge für Tutoren, welche einem Nutzer den fünften Rang nach dem Dreyfus-Modell erreichen lässt. Hinzu kommt, dass der fachliche Rang regelmäßig vom Nutzer bestätigt werden muss. Nach einem Jahr im selben Rang wird der Nutzer aufgefordert einen Test zu absolvieren. Besteht er den Test nicht, oder ignoriert er diesen, so fällt der Nutzer automatisch um einen Rang. Da es keinen Rang unterhalb von „novice“ gibt, werden Nutzer automatisch gelöscht, die den Test für den untersten Rang nicht bestehen. Diese Vorgehensweise wird so umgesetzt, da davon ausgegangen werden kann, dass ein aktiver Nutzer innerhalb eines Jahres den nächst höheren Rang erreicht. Weiterhin werden so inaktive und nicht interessierte Nutzer automatisch entfernt, was in einer regen Gemeinschaft resultiert. Dem Problem von Accounts, hinter dem kein aktiver Nutzer<sup>1</sup> mehr steht, wird somit vorgebeugt.

---

<sup>1</sup> sogenannten Zombies

### 5.2.1. Fachlicher Rang

Mit dem Durcharbeiten von WBTs kann ein Nutzer im fachlichen Rang aufsteigen. Der Hintergrund ist, dass er mit korrekten Antworten im Prüfungsteil der WBTs seine fachliche Kompetenz unter Beweis stellt. Demgegenüber werden bei falschen Antworten im Quiz keine negativen Punkte angerechnet. Je nachdem, wie gut ein Test ausfällt, erhält er eine bestimmte Anzahl an Punkten. Abhängig vom Grad des aktuellen fachlichen Rangs wird auch die notwendige Punktzahl für den nächsten Rang erhöht. Der Aufbau folgt also analog einer Exponentialfunktion. Wie in Abbildung 5.1 zu sehen ist, benötigt man im Vergleich mit den didaktischen Rängen im fachlichen Level mehr Punkte für den nächsten Rang. Im Gegensatz dazu wird hier maximal der Experten-Rang erreicht. In der Abbildung ist der Rang des Experten nicht zu sehen, da dieser das Erreichen der notwendigen kompletten Punktzahl symbolisiert.

Selbstverständlich können weitere WBTs durchgearbeitet werden, diese bessern jedoch nicht das Punktekonto für den fachlichen Rang auf. Dem Anwender ist freigestellt, ob er sich nun, wo er Experte in einem Fachgebiet ist, einem anderen Wissensgebiet widmet, um dort als Neuling von Vorn anzufangen.

### 5.2.2. Ränge für Tutoren

**Sternverlust nach einem Quartal ohne Unterweisung -> ansonsten könnte man sich ja auf seinen 5 Sternen ausruhen** Als Tutor wird man von den Lernenden beurteilt, die man in einem gewissen Fachgebiet unterstützt hat. Im Gegensatz zu den fachlichen Rängen sind die didaktischen Ränge vom Fach unabhängig. Auch bleiben sie über alle fachlichen Ränge hinweg erhalten. Ein weiterer Unterschied ist, dass man als Tutor nicht Punkte, sondern Sterne sammelt. Jede gute Bewertung (daumen rauf) gibt einen Schritt in Richtung weiteren Stern. Eine schlechte Bewertung (daumen runter) stellt dazu einen direkten Gegensatz dar. Beide Bewertungsrichtungen verhalten sich ausgeglichen und es kristallisieren sich Tutoren heraus, die fachliche Inhalte für jedermann verständlich zu erklären wissen. So ist auch gewährleistet, dass sich meisterliche Tutoren nicht auf ihren vier Sternen „ausruhen“.

Meisterliche Tutoren verfügen auch über das Privileg eigene WBTs in die Plattform einbringen zu können. Den niederen Rängen ist dies verwehrt, da diese unter Umständen Sachverhalte nicht allgemeinverständlich zu erläutern wissen. Auch sind meisterliche Tutoren dazu privilegiert sämtliche fachliche Ränge unterrichten zu können, während

für gewöhnlich Lernende nur von Tutoren unterwiesen werden, die maximal zwei fachliche Ränge über ihnen stehen.

Gegenüber den fachlichen Rängen ist in Abbildung 5.1 zu sehen, dass für den nächsten Rang bzw. Stern vergleichsweise weniger Punkte zu erreichen sind. Demgegenüber lässt sich nur als Tutor der Rang des Meisters, der vier Sternen entspricht, erreichen. Dieser Rang ist in der Abbildung nicht zu sehen, da er analog zum fachlichen Rang den Erhalt aller möglichen Punkte symbolisiert.

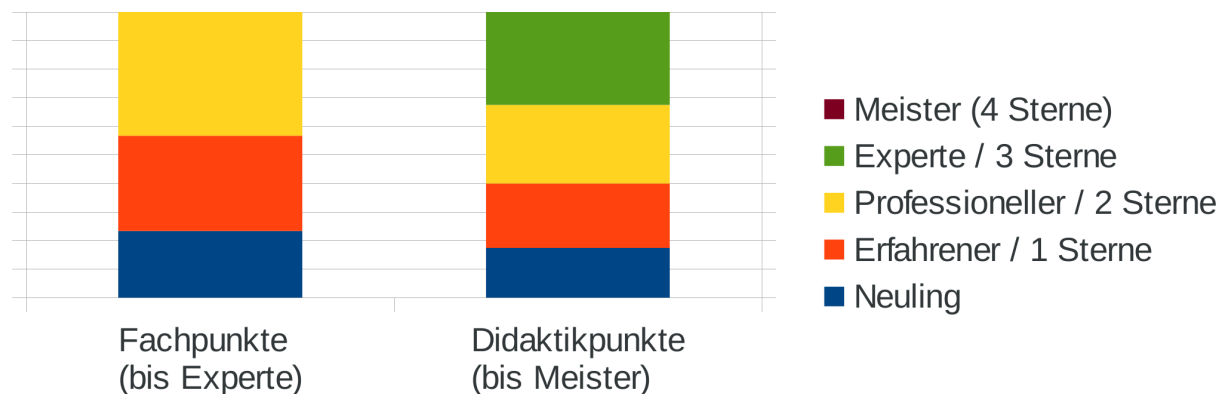


Abbildung 5.1.: Verteilung der Punkte

Ein Meister hat durch das Erhalten der höchsten Wertung für die didaktische Fähigkeit bereits bewiesen, dass er Spaß an der Vermittlung von Wissen hat. Demnach bedarf er keiner weiteren Motivation eines höheren Ranges. Vielmehr möchte er keine negativen Bewertungen seiner Lernenden erhalten und bemüht sich der weiteren hochwertigen Qualität seiner Lerneinheiten.

### 5.3. Rollen für Anwender

Passend zum zuvor beschriebenen Konzept werden drei Rollen für Anwender definiert. Dabei ist das Innehaben mehrerer Rollen zur gleichen Zeit ein Teil des Modells. Zusammenfassend sind die Rechte und Pflichten eines Nutzers in den verschiedenen Rollen in Tabelle B.1 aufgeführt. Das dort aufgeführte Forum ist nicht Teil der ersten produktiven Version (siehe Abschnitt 10.1)

### 5.3.1. Administrator

Der Administrator ist der Verwalter der Plattform und damit für den reibungslosen Ablauf seitens der Nutzer verantwortlich. Dazu kontrolliert er den Zusammenhalt des Systems und greift bei Inkonsistenzen oder Fehlern ein. Auch bildet er die Schnittstelle zur Community, die sich in der Weiterentwicklung von Masterly Mate engagiert.

### 5.3.2. Lernender

Der Lernende bildet die Hauptzielgruppe des Systems. Er soll WBTs finden, diese durcharbeiten können und sich an Tutoren wenden, falls er auf ein Problem oder Unklarheiten stößt. Dazu bietet Masterly Mate ihm das Auffinden eines an sein Fachwissen angepassten Trainings. Weiterhin kann er Tutoren kontaktieren, die in seinem Umkreis wohnen und passend zu seinem Rang Inhalte zu erläutern verstehen. Die Lokation wird anhand der Postleitzahl festgemacht.

Ein Lernender kann in seinem fachlichen Level bis zum Experten aufsteigen. Nähere Erläuterungen zu den fachlichen Rängen wurden in Abschnitt 5.2.1 aufgeführt.

### 5.3.3. Tutor

Ein Lernender kann ab dem zweiten fachlichen Rang des Dreyfus-Modells (siehe 3.1.3) in seinem Profil die Einstellung „Tutor“ anwählen. Damit erscheint er unter den Suchergebnissen für Lernende, die einen Tutor suchen. Als Tutor wird man von Lernenden gefunden, die Unterstützung in einem Fachgebiet suchen.

## 5.4. Dokumentation der API

mit „rake doc:app“ lässt sich die Dokumentation erstellen

Verfügbar machen mit Link auf „doc/app/index.html“

Uebernimmt Julian?

## 5.5. Automatisch generierte Filterung von Ergebnissen

- Profilinfos werden ausgelesen und zur Eingrenzung der Ergebnisse verwendet
- daher kein Suchfeld nötig

Die Suche von WBTs wird über reine SQL-Abfragen realisiert. Dies bedeutet, dass bei einem Aufruf der WBT-Liste, die Daten des betreffenden Users ausgelesen werden. Entsprechend des Levels eines Users im betreffenden Themengebiet werden die dafür vorgesehenen WBTs, welche er noch nicht absolviert hat angezeigt.

## 5.6. Themen

Die Themen (oder Topics/Themengebiete) grenzen die WBTs voneinander ab und sind neben den Levels ein Suchkriterium. Jedes WBT ist Teil mindestens eines Themengebieten. Außerdem sind die Ränge an die Themen gebunden. Der Benutzer kann WBTs in jedem Themengebiet absolvieren und im Rang aufsteigen. Dabei ist zu beachten, dass ein Benutzer einen bestimmten Rang nur in einem Themengebiet haben kann. Das bedeutet, dass er zwar in einem Gebiet den Status eines Expertise haben kann, in einem ganz anderen jedoch ein absoluter Newbie ist. Ist ein WBTs zwei (oder mehr) Themen zugeordnet, so erhält der Benutzer bei Abschluss die Punkte für beide Themengebiete. Themen können Unterthemen haben, daher kann ein Benutzer innerhalb eines Überthemas unterschiedliche Ränge haben. So kann z.B. ein Benutzer im Thema Informatik ein Expertise sein und zugleich in den Unterthemen Datenstrukturen und Algorithmen Specialist bzw. Newbie sein. Die Beziehung von Benutzer zu Themengebiet ist m:n, ebenso ist die Beziehung zwischen Thema und WBT m:n. Die Beziehung von Themen zu ihren Unterthemen ist 1:n. Das hier Beschriebene ist der Stand der Alpha-Version. Eine mögliche Änderung für die Zukunft wäre, dass erreichte Punkte eines Unterthemas auch im Überthema gutgeschrieben werden.

## 6. Anwenderseite

### 6.1. Gamification

In Abschnitt 3.1.6 wurden die wesentlichen Aspekte von Gamification angesprochen. Hier erfolgt die Konkretisierung derartiger Eigenschaften in die Konzeption von Masterly Mate.

#### 6.1.1. Belohnungskriterien

Um die Motivation stets auf einem hohen Niveau zu halten, bedient sich Masterly Mate einer Auswahl an Belohnungskriterien.

Die **erreichte Punktzahl** im Quiz eines WBTs wird für das Erreichen des nächst höheren Levels hinzugenommen. Analog dazu erhalten Tutoren **Sterne** für qualitativ hohe Unterweisungen. Zuzüglich zu den regulären Punkten für die fachlichen und den tutoriellen Rang können **Wertmarken** gesammelt werden mit denen einem Avatar oder dem GUI<sup>1</sup> mehr Gestaltungsmöglichkeiten verliehen werden.

#### 6.1.2. Etappenweise Herausforderungen

Ein weiteres Mittel, um die Motivation zu fördern sind Spielmechaniken, die in die Anwendung eingebracht werden. Für diese ist es erforderlich, dass für jede Art von Nutzer Anreize bestehen.

Die drei Ränge Neuling, regulärer Nutzer und Enthusiast sind nicht zu verwechseln mit den Rängen des Dreyfus-Modells aus Abschnitt 5.2. Die hier betrachteten Ränge in Bezug auf Gamification berücksichtigen die Handhabung der Software als solche. Ein

---

<sup>1</sup> Graphical User Interface



Neuling verwendet die Anwendung zum ersten mal oder bisher nur wenige male. Der reguläre Nutzer ist ein Stammgast der Plattform, während der Enthusiast schier nicht schlafen kann, ohne die Anwendung täglich verwendet zu haben.

Allen Nutzern ist gemeinsam, dass sie sich selbst gut evaluieren können. Sie erhalten die Möglichkeit Statistiken einzusehen. Für die erste Version von Masterly Mate ist eine Art Ladebalken vorgesehen, welcher je nach Füllstand zeigt, wie groß die Erfüllung der erforderlichen Gesamtpunktzahl für einen Rang ist. Über seine Wertung kann sich ein Nutzer stets im Feld aller Nutzer einordnen. Dabei kann jeder für sich individuell entscheiden, ob es für ihn wichtig ist, in die Top-Ten zu gelangen (Gruben, Merkle und Babics, 2012).

## **Neuling**

Als erstmaliger Nutzer einer Plattform oder Anwendung benötigen Neulinge einen vereinfachten Einstieg und eine leicht verständliche Anleitung. Auch muss die GUI übersichtlich gestaltet sein, sodass nicht bereits nach wenigen Klicks Frustration entsteht.

In Masterly Mate wird daher von Beginn an eine Möglichkeit zur Internationalisierung (siehe Abschnitt 7.2) umgesetzt. So sieht der Anwender das Interface stets in seiner Sprache der Wahl und stößt somit nicht auf Verständnisprobleme.

Weiterhin erhält ein neuer Nutzer analog seines fachlichen Rangs Zugang zu einfachen WBTs, die leicht verständlich sind. Stößt er hier bereits auf Probleme, so kann er sich einen Tutor zu Rate ziehen (Gruben et al., 2012).

Eine Anleitung im Sinne eines Handbuches ist für die erste produktive Version nicht angedacht.

## **Regulärer Nutzer**

Nutzer, die die Anwendung in moderater Weise nutzen, sind grundsätzlich zufrieden mit dem, was ihnen geboten wird. Sie sind aber noch begeisterungsfähig.

Ein solcher Stammgast kann in Masterly Mate auf noch höhere Ränge aufsteigen, denn ihm fällt es leichter den in Abschnitt 5.2.1 angesprochenen jährlichen Test zu bestehen. Als Tutor strebt er eventuell danach ein Meister zu werden.

Um ein Erlebnis analog des in Abschnitt 3.1.5 beschriebenen Flows zu unterstützen, steigt auch das Niveau der zur Verfügung stehenden WBTs. So kann der Stammnutzer bei Bedarf stets neue Herausforderungen suchen (Gruben et al., 2012).

### **Enthusiast**

Ein Enthusiast wird alles daran setzen seinen Experten oder meisterlichen Rang zu behalten. Er wird also weiterhin Unterweisungen geben und genießt sein Privileg, WBTs editieren zu können. Je nach Interesse kann er seinen Meister Rang dazu nutzen in andere Fachgebiete einzusteigen und dort für sämtliche Mitglieder unterhalb seines zugehörigen fachlichen Rangs mit Rat zur Seite zu stehen (Gruben et al., 2012).

## **6.2. Generieren von Motivation**

Zusammen mit der im vorangegangenen Abschnitt beschriebenen Konzeption von Gamification in Masterly Mate wird an dieser Stelle die Integration der in Abschnitt 3.1.2 beschriebenen Motivation erläutert.

In diesem Abschnitt wird ein Blick auf die motivierenden Merkmale aus Anwendersicht beschrieben, die im Wesentlichen bereits im vorangegangenen Abschnitt für die Umsetzung von Gamification besprochen wurden. Dieser wird separiert in intrinsische und extrinsische Motivation dargestellt.

### **6.2.1. Intrinsisch motivierende Aspekte**

Ein Nutzer von Masterly Mate strebt von sich aus nach Weiterbildung und damit nach einem höheren fachlichen oder didaktischen Rang. Beinahe nebenbei steigt er dafür immer tiefer in fachliche Themen ein. Für seine Leistungen möchte er Feedback erhalten und mit gleichgesinnten reden oder helfen.

Tutoren haben zusätzlich die Möglichkeit ihr Langzeitgedächtnis mithilfe von immer wiederkehrenden Unterweisungen zu schulen. Damit zusammenhängend trainieren sie sich in ihren didaktischen Fähigkeiten.

### **6.2.2. Weitere extrinsisch motivierende Merkmale**

Zu den intrinsischen Motivatoren kommen weitere extrinsische Motivatoren. Dazu gehören der Erhalt von Belohnungen, mehr Möglichkeiten in der Gestaltung des Interfaces oder Avatars oder eine Bestenliste zum Leistungsvergleich mit anderen Nutzern.

Macht man sich in der Plattform einen Namen als Tutor oder beteiligt sich an regen Diskussionen im Forum, so erfährt man Lob und Kritik von gleichgesinnten. Insgesamt lässt Masterly Mate eine persönliche Analyse in Form von Statistiken zu.

# **Teil III.**

## **Implementierung**

## 7. Entwurf

Der in diesem Kapitel beschriebene Entwurf zeigt konkret, wie das Konzept von Masterly Mate umgesetzt werden wird. Hier werden Schemata und Architekturen entwickelt, die im Kapitel 8 in Programmcode umgesetzt werden.

### 7.1. Realisierungsmethodik

Nachdem in den vorangegangenen Kapiteln das Konzept von Masterly Mate erläutert wurde, stellt sich die Frage nach einer geeigneten Möglichkeit zur Umsetzung. Da die Anwendung stets verfügbar, leicht erreichbar, modular und einfach zu verwalten sein soll, ist RoR das Mittel der Wahl. Dieses bietet viele interessante Features in Form von sogenannten gems, die dank einer regen Community stets aktualisiert und erweitert werden. Zudem unterstützt es moderne Programmierparadigmen, wie DRY und KISS<sup>1</sup>. Dadurch bleibt die Anwendung aus Sicht der Programmierer übersichtlich und erscheint sehr strukturiert. Das das Framework der MVC-Architektur folgt, schafft einen weiteren Grundstein zur Trennung von Zuständigkeiten<sup>2</sup> und sorgt auch damit für Übersichtlichkeit.

Weiterhin ist dieses Framework für die Weiterentwicklung im OpenSource-Bereich prädestiniert, da damit bisher populäre Webanwendungen, wie Twitter, realisiert wurden.

Darüber hinaus wird darauf geachtet, die Komponenten nach und nach nur dann zu entwickeln, wenn sie tatsächlich gebraucht werden. Diese Vorgehensweise nach dem YAGNI<sup>3</sup>-Prinzip beugt ein überlaufenes, unübersichtliches und schwer zu wartendes Produkt vor.

---

<sup>1</sup> Keep it simple stupid'

<sup>2</sup> bekannter unter „separation of concerns“

<sup>3</sup> You ain't gonna need it

## 7.2. Internationalisierung

Keine Internationalisierung in WBTs! (WBTs brauchen Attribut: Sprache) Was ist die Default-Language? -> Sprache des Nutzers (User auch Attribut: Muttersprache oder Herkunftsland)

Ueberrnimmt Julian?

## 7.3. Beschreibung des Entwurfsklassendiagramms und Use-Case

### 7.3.1. Use-Case Diagramm

Im obigen Use-Case Diagramm, sind die einzelnen benötigten Komponenten enthalten, wie sie die Analyse ergeben hat. Zunächst einmal lassen sich Aktoren ausmachen, welche in System-/Software-Aktoren und menschliche Aktoren aufteilen lassen.

Bei den Softwareseitigen Aktoren ergab die Analyse als Aktoren den Webserver, ein Einzelnes WBT (sie sollen nach Abschluss die Punkte im System eintragen), das Autorenwerkzeug und den Webbrowser. Auf der Seite der Aktoren ließe sich der Lernende (Standard-User), Tutor, Autor und Admin ermitteln. Zu beachten ist, dass die Aufteilung der menschlichen Aktoren auch eine Berechtigungshierarchie darstellt, wobei ein Lernender die geringsten und ein Admin die meisten Berechtigungen hat.

Die einzelnen Use Cases werden nun jeweils im folgenden kurz beschrieben. Zu beachten ist dabei noch, dass diese Analyse sich nicht zwingend mit dem Ergebnis aus der Studienarbeit deckt, da Funktionalitäten auf spätere Releases verschoben worden sind.

### WBT durcharbeiten

Agierende Aktoren:

- WBT
- Lernender

- Tutor
- Autor
- Admin

Ein Benutzer startet ein WBT und schließt es erfolgreich ab bzw. scheitert oder beendet es vorzeitig. Im Anschluss daran, werden die erreichten Punkte im System eingetragen. In der Version der Studienarbeit, wird dies noch vom Benutzer selbst durchgeführt. Später soll dies das WBT übernehmen.

## **Bewerten**

Agierende Akteure:

- Lernender
- Tutor

Den Use Case Bewerten gibt es in zwei Ausprägungen. Die erste Ausprägung ist die Bewertung eines Tutors. Hat ein Lernender oder eine anderer Tutor (im folgenden beide als "Lernende" bezeichnet) von dem Betreffenden Benutzer Hilfe erhalten, so kann der Lernende im Anschluss daran diese Hilfe im System bewerten. Dies kann er mit Sternen und wahlweise mit einem Kommentar tun (Kommentar noch nicht in der Alpha-Version). Die Zweite Ausprägung ist die Bewertung eines WBTs. Ein Benutzer muss das WBT dafür abgeschlossen haben. Ebenso wie beim Tutor kann er dies über Sterne und Kommentare tun.

## **Sprache verwalten**

Agierende Akteure:

- Alle menschlichen Akteure (Ausprägung Sprache hinzufügen kann nur der Admin)

Ebenso wie im obigen Use Case gibt es hiervon zwei Ausprägungen. Nummer Eins ist der Use Case Sprache wechseln. Dieser kann von jedem Benutzer durchgeführt werden und bewirkt, dass alle Zeichenketten in MasterlyMate, welche in der betreffenden Sprache vorhanden sind, ausgetauscht werden. Use Case Nummer Zwei kann nur von einem Administrator durchgeführt werden. Und bewirkt die Erzeugung einer weiteren

Sprache, welche eingestellt werden kann. Gegenwärtig wird dies noch direkt durch eine Änderung der Sprachdateien getan.

## **Profil verwalten**

Agierende Aktoren:

- Alle menschlichen Aktoren

Auch dieser Use Case verfügt über zwei Ausprägungen. Der erste Use Case ist das bearbeiten eines Profils. Dies kann nur der Besitzer des jeweiligen Profils. Den anderen Use Case kann hingegen jeder Benutzer durchführen. Es handelt sich dabei um das Betrachten eines Profils. Dies ist insbesondere für die Kontaktaufnahme eines Benutzers zu einem Tutor notwendig.

## **Suchen**

Agierende Aktoren:

- Alle menschlichen Aktoren

Es können sowohl Tutoren als auch WBTs gesucht werden. Dabei wird jeweils berücksichtigt welches Themengebiet gefragt ist.

## **Themengebiet abfragen**

Agierende Aktoren:

- Webserver

Der Webserver überprüft bei einer Suchanfrage die einzelnen Themen.

## **Themen verwalten**

Agierende Aktoren:

- Admin
- Webserver



Nur der Admin kann die beiden Ausprägungen dieses Use Cases durchführen, welche das Hinzufügen und Löschen eines Themas darstellen.

## **WBT verwalten**

Agierende Akteure:

- Admin
- Autor
- Webserver

Dieser Use Case besitzt drei Ausprägungen. Admin und Autor können WBTs einbinden. Es kann jedoch nur der User, welcher ein WBT eingebunden hat, dieses auch wieder entfernen bzw. ersetzen. Dies gilt nicht, wenn der betreffende Benutzer ein Administrator ist. Dieser kann jedes WBT löschen oder ersetzen.

### **7.3.2. Entwurfsklassendiagramm**

In dem Entwurfsklassendiagramm, sind die Models und ihre Beziehungen zueinander dargestellt. Diese Modelle bilden einzelne Konzepte im System ab. So stellt User ein Zentrales Modell dar, welches als Abbildung eines Benutzers im System für die Authentifizierung verantwortlich trägt. Außerdem werden dem User verschiedene Themen und WBTs zugeordnet. Diese Verbindungen benötigen jeweils Assoziationsklassen. Dies liegt daran, dass es in dieser Verbindung Attribute gibt, welche sich nicht eindeutig User oder WBT bzw. Thema zuordnen lassen. Bei der Verbindung User-Thema benötigt man ein Attribut für die Punkte, welche bisher von dem User in dem System erzielt worden sind. Außerdem muss klar sein welchen Rang ein User in dem jeweiligen Thema inne hat. Die Verbindung User-WBT dagegen benötigt eine Variable um zu hinterlegen ob ein User das WBT bereits einmal abgeschlossen hat und wenn ja, wie viele Punkte er erreicht hat. Das WBT-Model verfügt über eine ID, sowie über einen Pfad zur eigentliche SCORM-Datei. Wie bereits erwähnt ist es einem oder mehreren Themen (Topics) zugeordnet.

## 7.4. Komponenten

Struktur hier unklar. . . noch warten auf die anderen Inhalte von JeyB und Benni. Prinzipiell ist Masterly Mate aus zwei Komplexen aufgebaut. Zum einen kann sich ein Nutzer fachlich weiterbilden. Zum Anderen bietet ein Nutzer als Tutor seine Hilfe für ein bestimmtes Fachgebiet an.

### 7.4.1. Durcharbeiten von WBTs

### 7.4.2. Tutorensuche

### 7.4.3. Navigation, Impressum, Kontakt

Uebernimmt Julian

### 7.4.4. Themen

Themen fungieren für den Benutzer zum Einen als Suchfilter bei der Suche nach den WBTs. Dies sorgt dafür, dass der Benutzer ohne umschweife auf WBTs zugreifen kann, welche für ihn interessant sind. Außerdem dienen Themen dazu, die unterschiedlichen Ränge, welche ein Benutzer zur selben Zeit in Masterly Mate haben kann, voneinander abzugrenzen. Jeder Benutzer kann in einem Thema nur genau einen Rang inne haben. Umgekehrt kann er jedoch in beliebig vielen Themen einen Rang haben.

## 8. Umsetzung

### 8.1. Nutzerverwaltung

Uebernimmt Julian?

### 8.2. Lokationen

Uebernimmt Julian?

### 8.3. Suche

Uebernimmt Julian/Benni?

### 8.4. SCORM

Die SCORM-Funktionalität wird in dem Objekt „wbt“ realisiert, welches in Abschnitt 7.3.2 beschrieben wurde.

Die darin enthaltene Upload-Methode, die beim Einfügen und Ändern eines WBTs aufgerufen wird, speichert das WBT auf dem lokalen Speicher des Servers. Dabei wird das PIF<sup>1</sup> mithilfe der Funktionen aus einem scorm-gem<sup>2</sup> direkt entpackt. Dabei ist mit der Berücksichtigung der Validierung stets ein Fehler aufgetreten. Für die erste Version wird daher keine Validierung unterstützt, das PIF wird ohne Prüfung entpackt (siehe

---

<sup>1</sup> Package Interchange File

<sup>2</sup> Ressource: <https://rubygems.org/gems/scorm>

9). Die Fehlerursache liegt unter Umständen an dem Alter des gems. Eine genauere Betrachtung ist für spätere Versionen von Masterly Mate angedacht.

Mit dem Hochladen und Entpacken werden die nötigen Attribute für den Start des WBT gepflegt. Dies ist zum einen der Paketname selbst und zum anderen der Pfad zur Start-Datei des Root SCO. In einer start-Methode werden diese Attribute ausgelesen und das WBT wird in einem neuen Fenster geöffnet. So kann das WBT den Raum einnehmen, den es braucht. Masterly Mate bleibt damit unabhängig vom Stil des Autorenwerkzeugs. Für die erste produktive Version fehlt es noch an einem geeigneten RTE, da dessen Implementierung den Rahmen der vorliegenden Arbeit sprengen würde. Daher erfolgt die Registrierung des Ergebnisses zunächst durch eine manuelle Eingabe des Nutzers.

## 8.5. Lizenzierung

Nach den in Abschnitt 3.1.7 beschriebenen freien Lizenzen und dem Einbringen der Idee in das Konzept (siehe Abschnitt 4.2) wurde für die Zwecke von Masterly Mate auf die AGPL<sup>1</sup> zurückgegriffen. Da diese mit der GPL<sup>2</sup> kompatibel ist, wird eine eventuelle Verbreitung der Software im Sinne von OpenSource möglich. Darüber hinaus kann das Projekt unter anderen kompatiblen Lizenzen verbreitet werden (Free Software Foundation, 2007).

Mit der Nutzung der AGPL entsteht die Pflicht, den Quelltext der Anwendung direkt als Download anzubieten. Dazu wird im Interface Masterly Mate ein Link auf die GIT-Ressource im Footer angeboten. Zusätzlich wurde, wie bei allen Lizenzen nötig, in jeder Datei ein Lizenztext vorrangestellt.

## 8.6. Dokumentation

mit rake doc:app wurde eine API-Dokumentation erstellt, die im Footer referenziert wurde

Ueberrimmt Julian?

---

<sup>1</sup> Affero GNU General Public License

<sup>2</sup> GNU General Public License

## 8.7. Gestaltung

Das Design von Masterly Mate ist in der ersten Version zunächst einmal von Funktionalität geprägt. Jedoch finden schon einige der weiter oben aufgezählten gestalterischen Grundkonzepte hier Eingang. So ist es der Erwartungskonformität geschuldet, dass Navigation und Anzeige sich in optisch voneinander getrennten Bereichen befinden. Ebenfalls von vielen anderen Seiten bekannt, ist das Konzept, die Änderung der Sprache oben rechts und damit abseits von allen anderen Kontrollen zu platzieren. Die Navigation ist von der Anzeige durch das Gesetz der Nähe und einen Sichtbaren Trennstrich getrennt. Um eine bessere Übersichtlichkeit auf den WBTs zu gewährleisten, werden diese außerdem in einem neuen Tab geöffnet. Da Masterly Mate als gamifizierte Anwendung nicht vollkommen auf ein Mindestmaß an ansprechender Optik verzichten kann, ist es in einem warmen Orange gehalten. Die wenigen anderen Farben sind so gewählt, dass sie nicht zu sehr in Kontrast zur Hauptfarbe stehen.

## 8.8. Authentifizierung

Ueberrimmt Julian?

## 8.9. Themen

Der Model Topic, welches die Themen abbildet verfügt über die Attribute name und parent\_name. Zweites dient zur Identifizierung des Überthemas. Die Wahl dieser Form der Referenz zu verwenden ist der Datenbank geschuldet. Auf diese Weise liegt der Fremdschlüssel jeweils auf der n-Seite der 1:n Beziehung.

# **Teil IV.**

## **Reflexion**

## 9. Zusammenfassung der Ergebnisse

Es hat sich gezeigt, dass RoR sehr mächtig ist. Innerhalb kürzester Zeit entstehen Webapplikationen mit umfangreichen Funktionen. Für Masterly Mate reichten vorhandene gems aus und es musste kein proprietärer Ruby-Code geschrieben werden.

**Aufgetretene Probleme** Sehr problematisch war hingegen der Umgang mit SCORM. Diese, recht komplexe, Spezifikation konnte längst nicht in vollem Umfang implementiert werden. Es steht auch zur Diskussion, ob das gewählte scorm-gem die gewünschten Ziele in Kommunikation mit einer SCORM-RTE fehlerfrei erfüllt. Derzeit entstehen, wie bereits in Abschnitt 8.4 erwähnt, Fehler bei der Validierung von PIFs.

Es war für die erste produktive Version angedacht, eine Kommunikation zwischen LMS und WBT bereitzustellen. Nach Recherchen hat sich gezeigt, dass dieses Vorhaben den Umfang eines eigenen Projektes beansprucht. Alternativ ist die Verwendung des SCORM-Players, wie in (Mitter, 2005) und (Knall, 2005) zu überdenken.

ist das alles?

# 10. Fazit und Ausblick

## 10.1. Ideen für weitere Versionen

- Nutzer kann eine Liste aller WBTs sehen, die er absolviert hat
- Forum
- Durchführung von Refactorings für DRY KISS
- Profilbild
- SCORM RTE implementieren
- Mailerfunktionalität
- Investierte Zeit mit in das Konzept für Belohnungen einbringen
- Am Ende jedes WBT eine Wertung abgeben (zu schwer / zu leicht) -> tendiert eine Wertung zu stark in eine Richtung, wird das WBT dem am nächsten passenden Rang zugeordnet
- Adaptierung von Lerninhalten, wie in (Knall, 2005)?
- SCORM-Player (Mitter, 2005)
- noch Überraschungen mit einbringen? (Korte, 2009)

### 10.1.1. Gamification für Newbies

- Quick-Start Guide (Video-)Instruction, (Video-)Tutorial
- Statistical evaluation (Ranking)
- Unlock big equipments for the selected design
- Possibility of using a Open-ID



- Newsletter-Feature
- Self-assessment regarding to school grades

### **10.1.2. Gamification für Regulars**

- Collections of Achievements according to the current progression

### **10.1.3. Gamification für Enthusiasten**

- Levels
- Dynamic difficulty / i.e. riddles

### **10.1.4. Weitere Möglichkeiten für Gamification**

- Design Selection (Tamagochi, Avatar)
- Assistance possible
- Progress bar
- status message
- discussion room
- class room

wie schaut das Gamification-Konzept von MM aus? Irgendwie ist Fortschrittsbalken und Statistik und Möglichkeit als Tutor zu gering bis Meister. Gut ist der Anreiz, ab Meister WBTs einbringen und editieren zu können. Wie war das mit dem Avatar oder der UI, die immer besser gestaltet werden kann?

## 10.2. Anschließende Vorhaben

sehen, was wird

- wirklich nur in Ausnahmefällen eine DE? (siehe 3.1.3)
- (un)populär
- neue Verwendungszwecke
- nur eine Installation oder mehrere

kann mithilfe von Masterly Mate ein idealer Lehrer gefunden werden?

# Abbildungsverzeichnis

2.1. Wortwolke über die „Masterly Mate“-Idee . . . . .	3
3.1. Umstrukturierung von Synapsen . . . . .	8
3.2. Komponenten eines SCORM-Paketes . . . . .	17
3.3. Schema der SCORM-Manifest Datei . . . . .	19
3.4. SCORM-API Anbindung . . . . .	20
5.1. Verteilung der Punkte . . . . .	33

# Tabellenverzeichnis

3.1. Ruby on Rails Komponenten . . . . .	25
3.2. Komponenten der MVC-Architektur und deren Eigenschaften . . . . .	27
B.1. Rechte und Pflichten der verschiedenen Rollen . . . . .	x

# Listings

# Literaturverzeichnis

- ADL (2009), Sharable Content Object Reference Model (R) 2004 4th Edition Run-Time Environment Version 1.0, Technical report, ADL, USA. <http://www.moschorus.com/centre/?b0c2e7>.
- ADL (2011), SCORM Users Guide for Programmers, Technical report, ADL, USA. [http://www.adlnet.gov/wp-content/uploads/2011/12/SCORM\\_Users\\_Guide\\_for\\_Programmers.pdf](http://www.adlnet.gov/wp-content/uploads/2011/12/SCORM_Users_Guide_for_Programmers.pdf).
- Baumgartner, P., Häfele, H. und Maier-Häfele, K. (2002), *E-Learning Praxishandbuch : Auswahl von Lernplattformen; Marktübersicht, Funktionen, Fachbegriffe*, StudienVerl., Innsbruck.
- Breitlauch, L. (2013), 'Linda Breitlauch über Serious Games – Quo Vadis 2013', <http://video.golem.de/games/10714/linda-breitlauch-ueber-serious-games-quo-vadis-2013.html>. Video.
- Csikszentmihalyi, M. und Schiefele, U. (1993), 'Die Qualität des Erlebens und der Prozeß des Lernens', *Zeitschrift für Pädagogik*.
- Dreyfus, S. E. und Dreyfus, H. L. (1980), A Five-Stage Model Of The Mental Activities Involved In Directed Skill Acquisition, Technical report, University of California, Berkley. <http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA084551&Location=U2&doc=GetTRDoc.pdf>.
- Fiedling, R. (2000), Architectural Styles and the Design of Network-based Software Architectures, Dissertation, University of California, Irvine. [http://www.ics.uci.edu/~fielding/pubs/dissertation/fielding\\_dissertation.pdf](http://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf).
- Free Software Foundation (2007), 'GNU AFFERO GENERAL PUBLIC LICENSE', <http://www.gnu.org/licenses/agpl-3.0.de.html>.

- Gruben, M. (2012), Analyse und Vergleich von Autorensystemen für ein WBT zu Vorlesungsinhalten. Studienarbeit.
- Gruben, M., Merkle, B. und Babics, J. (2012), Web Based Training and the Dreyfus-Model. Präsentation.
- Horton, W. K. (2000), *Designing Web based training : how to teach anyone anything anywhere anytime*, Wiley, New York.
- Jacobs, P. und Preuße, M. (2010), *Kompaktwissen AEVO in vier Handlungsfeldern*, 1. Aufl. edn, Bildungsverl. EINS - Stam, Troisdorf.
- Keller, H. B. (2000), *Maschinelle Intelligenz : Grundlagen, Lernverfahren, Bausteine intelligenter Systeme*, Computational intelligence, Vieweg, Braunschweig.
- Knall, T. (2005), Automatische Adaptierung von SCORM-basierenden Lerninhalten, Master's thesis, Technische Universität Graz, Institut für Informationssysteme und Computer Medien (IICM), A-8010 Graz.
- Korte, M. (2009), 'Wie das Lernen gelingt', *Geo-Wissen* **44**, 28–31.
- Koster, R. (c2005), *A theory of fun for game design*, Paraglyph Press, Scottsdale, AZ.
- Kreutzer, T. (2011), *Open Content Lizenzen : ein Leitfaden für die Praxis*, Deutsche Unesco-Kommission, Bonn.
- Kröger, H. und Reisky, A. (2004), *Blended Learning - Erfolgsfaktor Wissen*, Wissen und Bildung im Internet ; 6, Bertelsmann, Bielefeld.
- Mitter, M. (2005), Implementierung eines SCORM-basierenden Zugangs zu Web-basierendem Unterricht, Master's thesis, Technische Universität Graz, Institut für Informationssysteme und Computer Medien (IICM), A-8010 Graz.
- Niegemann, H. M., ed. (2004), *Kompendium E-Learning*, X.media.press, Springer, Berlin.
- RailsGuides (2013), *Getting Started with Rails*. [http://guides.rubyonrails.org/getting\\_started.html#what-is-rails](http://guides.rubyonrails.org/getting_started.html#what-is-rails).
- Raymer, R. (2011), 'Gamification: Using Game Mechanics to Enhance eLearning', *eLearn Magazine* .
- Schleifer, H.-J. (2003), e-Learning an der LVG – Neue Formen des Unterrichts an der LVG Heidelberg, Technical report, LVG Heidelberg.

- Spitzer, M. (2012), *Digitale Demenz : wie wir uns und unsere Kinder um den Verstand bringen*, Droemer, München.
- Stampfl, N. S. (2012), *Die verspielte Gesellschaft : Gamification oder Leben im Zeitalter des Computerspiels*, Telepolis, 1. Aufl. edn, Heise, Hannover.
- Thomé, D. (1989), Kriterien zur Bewertung von Lernsoftware : mit einer exemplarischen Beurteilung von Deutsch-Lernprogrammen, PhD thesis, Universität Heidelberg, Heidelberg.
- Wendt, M. (2003), *Praxisbuch CBT und WBT : konzipieren, entwickeln, gestalten*, Hanser, München.



# Anhang

# A. Abbildungen

## **B. Tabellen**

Tabelle B.1.: Rechte und Pflichten der verschiedenen Rollen

	<b>Admi- nistrator</b>	<b>Lernender</b>	<b>Lehrender (Tutor)</b>	<b>meisterlich- er Tutor</b>
WBTs lesen	✓	✓	✓	✓
Punkte aus Quiz in WBTs ziehen	✗	✓ (bis Exper- te)	✗	✗
WBTs erstellen & löschen	✓	✗	✗	✓ (nur eigene)
WBTs bearbeiten	✓	✗	✗	✓
Im Rang steigen	✗	✓ (bis Exper- te)	✓ (bis Meister)	✗
Im Rang fallen	✗	✓ (bei nicht bestehen oder igno- rieren eines jährlichen Tests)	✓ (bei zu vie- len negativen Bewertungen)	✓ (bei zu vie- len negativen Bewertun- gen)
Lernende unter- weisen	✗	✗	✓ (maximal 2 fachliche Rän- ge unter dem eigenen)	✓ (jeder un- terhalb des ei- genen fachli- chen Rangs)
von Lernenden bewertet werden	✗	✗	✓	✓
Forum moderie- ren	✓	✗	✗	✗
zum Forum bei- tragen	✓	✓	✓	✓
Themen bearbei- ten	✓	✗	✗	✗

# C. GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

`<http://fsf.org/>`

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

## Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

## 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “**Document**”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “**you**”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “**Modified Version**” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “**Secondary Section**” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “**Invariant Sections**” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “**Cover Texts**” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “**Transparent**” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to

thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “**Opaque**”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “**Title Page**” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The “**publisher**” means any person or entity that distributes copies of the Document to the public.

A section “**Entitled XYZ**” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “**Acknowledgements**”, “**Dedications**”, “**Endorsements**”, or “**History**”.) To “**Preserve the Title**” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

## 2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license

notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

### 3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.



It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

## 4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.

- I. Preserve the section Entitled “History”, Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled “History” in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the “History” section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled “Acknowledgements” or “Dedications”, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## 5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

## 6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

## **7. AGGREGATION WITH INDEPENDENT WORKS**

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

## **8. TRANSLATION**

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and

the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

## **9. TERMINATION**

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

## **10. FUTURE REVISIONS OF THIS LICENSE**

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

## **11. RELICENSING**

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

## **ADDENDUM: How to use this License for your documents**

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with ... Texts.” line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.