

Implementation von Feedbackmechanismen in Onlinekursen am Beispiel Masterstudy LMS

Bachelorarbeit von

Elias Kia

an der Fakultät für Informatik
Institut für Informationssicherheit und Verlässlichkeit (KASTEL)

Erstgutachter:	Prof. Dr. Anne Koziolk
Zweitgutachter:	Prof. Dr. Ralf Reussner
Betreuender Mitarbeiter:	M.Ed. Kai Marquardt
Zweiter betreuender Mitarbeiter:	Dr. Lucia Happe

15. Juli 2022 – 15. November 2022

Karlsruher Institut für Technologie
Fakultät für Informatik
Postfach 6980
76128 Karlsruhe

Zusammenfassung

Das Erhalten von Feedback ist ein wichtiger Teil des Lernprozesses. Doch wie steht es um den Einsatz von Feedback in Onlinekursen, die über Weboberflächen bearbeitet werden? In dieser Arbeit wurden zwei Ansätze zur Vermittlung von Feedback betrachtet, welche Einsatz in Onlinekursen finden. Diese sind erstens ein Fortschrittsbalken zur Anzeige des Kursfortschritts und zweitens ein Overlay mit durch den Kursersteller einstellbaren Nachrichten. Beide Ansätze wurden für eine genauere Untersuchung ausgewählt und theoretisch analysiert. Diese Auswahl wurde getroffen, da es bisher kaum wissenschaftliche Arbeiten gibt, die diese Art des Feedbacks in einem Kontext von Onlinekursen isoliert betrachten.

Um die Auswirkungen dieser Feedbackmechanismen auszuwerten, wurden sie für eine bereits bestehende Webseite mit Onlinekursen implementiert. Die vorgegebene Webseite „Rockstartit“ wurde mit WordPress und dem Plugin „MasterStudy LMS“ entwickelt. Aufbauend darauf wurde die Implementation der Feedbackmechanismen vorgenommen. Im Rahmen der Implementierung wurde festgestellt, dass das Plugin „MasterStudy LMS“ nur sehr eingeschränkt für Erweiterungen dieser Art geeignet war. Für zukünftige Arbeiten ist zu überlegen, ob die Verwendung einer weniger einschränkenden Plattform bzw. eines eignen Plugins langfristig eine gute Entscheidung wäre.

Abschließend wurden in einer exemplarischen Studie die Erkenntnisse aus verwandten Arbeiten nach der Implementierung praktisch überprüft. Durch den Vergleich einer Testgruppe mit Fortschrittsbalken und einer Kontrollgruppe ohne Fortschrittsbalken im Kurs konnten über ein Quasi-Experiment Unterschiede in der Evaluation deutlich gemacht werden. So hilft der Fortschrittsbalken im Durchschnitt mit der Orientierung und um den Überblick über den Kurs zu behalten. Auch mit der Motivation den Kurs zu bearbeiten oder ein Erfolgsgefühl zu vermitteln, könnte ein Fortschrittsbalken helfen. In keinem Fall wurde dieser als negativ oder störend empfunden. Weitere umfassendere Studien könnten diese Ergebnisse bestätigen. Zur Evaluation des Overlays wurde ein Ansatz vorgeschlagen, der in zukünftigen Arbeiten umgesetzt und möglicherweise in einer größer angelegten Studie analysiert werden kann.

Inhaltsverzeichnis

Zusammenfassung	i
1 Einleitung	1
1.1 Motivation	1
1.2 Forschungsfragen	2
2 Grundlagen	3
2.1 Wirkung und Absicht von Feedback	3
2.2 Festlegung was gutes Feedback ausmacht	4
2.3 Methoden zur Vermittlung von Feedback	8
2.4 Gamification	8
2.5 Lernmanagement-Systeme	9
2.6 WordPress Grundlagen	10
2.6.1 Posts	10
2.6.2 WordPress Plugins	11
2.6.3 Hooks	12
2.6.4 Rockstartit	13
2.6.5 MasterStudy LMS	14
3 Verwandte Arbeiten	15
3.1 Arbeiten zu Fortschrittsbalken	15
3.2 Arbeiten zur Effektivität von Kommentaren als Feedback	16
3.3 System zur Feedbackausgabe an Gruppen	17
3.4 System zur Ausgabe von adaptivem Feedback für Programmieraufgaben	18
4 Anforderungen an die Implementation	19
4.1 Begründung für die Auswahl	19
4.1.1 Fortschrittsbalken	19
4.1.2 Feedback bei Vervollständigung	20
4.2 Anforderungen	21
4.2.1 Fortschrittsbalken	21
4.2.2 Feedback bei Vervollständigung	21
5 Implementation	23
5.1 Design	23
5.2 Besonderheiten	26
5.2.1 Einbindung von CSS und JS Dateien	26
5.2.2 Anpassung des <i>Complete</i> Knopfes	27

5.2.3	Gleiches Anzeigen des Fortschrittbalkens zwischen Chrome und Firefox	28
5.2.4	Korrekte Anzeige von H5P Elementen	30
5.3	Testszzenarien	31
5.3.1	Szenario 1	31
5.3.2	Szenario 2	31
5.3.3	Szenario 3	32
5.4	Limitationen	32
5.5	Ergebnisse	33
5.5.1	Fortschrittsbalken	33
5.5.2	Overlay	33
5.6	Modularer Ansatz	35
5.7	Anpassungen nach Codereview	37
5.7.1	Klärung von Variablennamen	37
5.7.2	Überflüssige Kommentare	38
5.7.3	Entfernen nutzloser expliziter Variablen	38
5.7.4	CSS Werte im JS	38
6	Evaluation des Fortschrittbalkens	41
6.1	Design	41
6.2	Motivation und Hypothese	41
6.3	Instrument - Konzeption und Begründung	42
6.4	Ergebnisse	44
6.5	Diskussion	46
7	Limitationen und zukünftige Arbeiten	49
7.1	Limitationen	49
7.2	Zukünftige Arbeiten	50
8	Fazit	51
	Literatur	53

Abbildungsverzeichnis

2.1	Zusammenfassung der Struktur von Feedback nach Hattie und Timperley, übersetzt entnommen aus [17]	7
2.2	Die Funktion <i>function_1</i> aus Skript zwei wird zum <i>Action Hook hook_1</i> registriert, der dann in Skript eins ausgelöst und <i>function_1</i> damit aufgerufen wird.	13
2.3	Ein Kurs mit vier Lektionen. Für WordPress sind sowohl Kurs als auch Lektion <i>Posts</i>	14
5.1	Diagramm, welches das Plugin „a-rockstartit-masterstudy-integration“, sowie dessen Interaktion mit „MasterStudy LMS“ darstellen soll.	24
5.2	Zeitlicher Verlauf beim Auslösen der für das Plugin relevanten <i>Hooks</i>	24
5.3	Ablauf beim Generieren des Overlays mit dem Feedback Kommentar.	25
5.4	Ablauf beim Generieren des Fortschrittsbalkens.	26
5.5	Die Navigationsleiste einer unvollständigen Lektion mit Fortschrittsbalken.	33
5.6	Die Navigationsleiste einer vervollständigten Lektion mit Fortschrittsbalken.	33
5.7	Eine Lektion mit sichtbarem Overlay, in dem Text zu sehen ist.	34
5.8	Eine Lektion mit sichtbarem Overlay, in dem ein H5P Element zu sehen ist.	34
5.9	Die neu hinzugefügten Einstellungen einer Lektion.	35
5.10	Modulare Aufteilung des Plugins.	36
6.1	Übersicht zu den Ergebnissen.	45

Tabellenverzeichnis

2.1	Feedback Methoden und mögliche Umsetzungen in Online-Umgebungen	8
2.2	Typische Gamification Elemente nach Nah et al. [21]	9
6.1	Informationen über die Probanden	44
6.2	Durchschnittswerte und Standardabweichungen zu den Aussagen	44

1 Einleitung

1.1 Motivation

In der Lehre stellt Feedback einen wichtigen Teil des Lernprozesses dar. In einer traditionellen Klassenumgebung einer Schule muss eine Lehrkraft, so gut es ihr je nach Größe der Gruppe möglich ist, allen Schülerinnen und Schülern ein möglichst angemessenes und individuelles Feedback zu ihrer Leistung geben. Bereits hier ist es schon schwierig, unter den gegebenen Zeitvorgaben dieser Aufgabe wahrhaftig gerecht zu werden. Noch schlimmer steht es um dieses Problem bei Hochschulen, wo noch mehr Studierenden unter noch strengeren Zeitvorgaben Feedback gegeben werden soll.

In den letzten Jahren sind viele Online-Plattformen entwickelt worden, die versuchen bei diesem Problem Abhilfe zu schaffen. Dabei sind viele unterschiedliche Ansätze entstanden, um bei Aufgaben wie beispielsweise der Verwaltung von Lernmaterial oder auch der Ausgabe von Feedback Unterstützung zu leisten oder diese ganz zu übernehmen. Wenn ein System vollständig automatisiertes Feedback geben kann, stellt dies eine große Entlastung für Lehrkräfte, sowie einen Gewinn für den Lernprozess dar. Zusätzlich wird dadurch auch eine bessere Skalierbarkeit des Kurses ermöglicht, da alle Schülerinnen und Schüler nun automatisiert Feedback erhalten können. Wie gut dies dann funktionieren kann, hängt selbstverständlich von der tatsächlichen Implementierung ab. Weiter gibt es zwar viele Studien dazu, was als Feedback gesehen werden kann, was gutes Feedback ausmacht und auch inwiefern dieses im Unterricht effektiv sein kann. Jedoch sind einige Studien (siehe unter anderem [10, 12, 27, 11]), gerade im Bereich von Online-Lehrplattformen ungenau, wenn es darum geht, die Effektivität einzelner Arten von Vermittlung von Feedback zu beurteilen. Typischerweise werden mehrere Ansätze für Feedback auf einmal angewandt und hinterher nur das Gesamtbild ausgewertet. Selbstverständlich sind für eine möglichst optimale Lernerfahrung auch mehrere Methoden auf einmal in Betracht zu ziehen, aber so bleibt es schwierig beurteilen zu können, welche Methoden hilfreich sind und welche nicht. Ein großer Teil der Arbeiten, die dazu im Lehrkontext gefunden werden konnten, prüft höchstens, welche Art des Feedbacks aus einer Auswahl am besten ankam. Es fehlen also Arbeiten darüber, ob eine spezifische Art Feedback zu vermitteln, für sich allein genommen einen positiven Einfluss auf die Bearbeitung eines Onlinekurses hat.

Eine einfache Art, Schülerinnen und Schülern Feedback zu geben, die in diesem Kontext noch nahezu unerforscht scheint, ist der Einsatz von Fortschrittsbalken. Typischerweise sind Fortschrittsbalken im Kontext eines Ladevorgangs vorzufinden. In dieser Arbeit soll der Fortschrittsbalken dazu eingesetzt werden, um den Fortschritt im Kurs anzuzeigen und damit Motivation zu erzeugen. Es scheint auch vielversprechend, gezielter Forschung in diesem Bereich nachzugehen, da Arbeiten, die unterschiedliches Feedback in Onlinekursen untersucht haben, darüber berichten, wie Fortschrittsbalken als gute und sinnvolle

Elemente wahrgenommen wurden (siehe [10, 24]). Dementsprechend ist die Absicht dieser Arbeit, erste Schritte in diese Richtung der Forschung zu gehen. In einer kleinen Studie soll geprüft werden, ob es Hinweise darauf gibt, dass ein Fortschrittsbalken zu einer Verbesserung der Motivation bei der Bearbeitung eines Onlinekurses führt. Dafür muss ebenfalls eine Online-Plattform gewählt werden, auf der solche Kurse vorhanden sind. Eine solche Online-Plattform, namens „Rockstartit“, wurde für Bildungs- und Forschungszwecke bereits erstellt, und wird nach wie vor weiterentwickelt (siehe Abschnitt 2.6.4 für eine genauere Beschreibung). Momentan fehlen noch gewisse Möglichkeiten, in den Lerneinheiten Feedback an die Schülerinnen und Schüler zu geben. Durch das Erweitern dieser Plattform können diese ersten Schritte erfolgen. Idealerweise kann die hier entstandene Arbeit als Grundlage für das Hinzufügen von weiterem Feedback dienen.

1.2 Forschungsfragen

Für diese Arbeit wurden zwei Ziele angestrebt. Erstens wurde der von der Fachliteratur empfohlene Fortschrittsbalken, sowie ein Overlay mit einstellbarem Feedback zur Analyse im Rahmen dieser Arbeit gewählt und implementiert. Die konkrete Auswahl der Feedbackmechanismen für diese Arbeit ist in 4.1 begründet. Zweitens wurde Einsatz von Fortschrittsbalken darauf untersucht, ob dieser den Lernprozess positiv beeinflusst. Der Kernbestandteil dieser Arbeit bestand aus der Auswahl und der Implementation der Mechanismen, sowie der Evaluation des Fortschrittsbalkens. Die Forschungsfragen, die im Rahmen dieser Arbeit bearbeitet wurden, lauten also:

RQ1 Wie kann eine Umsetzung von Online Feedback in einem Kurs aussehen?

RQ2 Können die ausgewählten Feedbackmechanismen den Lernprozess positiv beeinflussen?

RQ2a Können Fortschrittsbalken in Onlinekursen Motivation, den Eindruck etwas zu erreichen und den Kursüberblick positiv beeinflussen?

RQ1 bezieht sich, neben dem Betrachten anderer Ansätze, auch insbesondere auf die Möglichkeiten zur Implementierung unter den gegebenen Rahmenbedingungen. Um RQ2 nachzugehen, wurde mit dem entwickelten Fortschrittsbalken eine Studie über dessen Wirkung durchgeführt. Die Ergebnisse stellen eine Hilfe für die Ermittlung zukünftiger Arbeiten dar. Dies gilt speziell für den Einsatz von Fortschrittsbalken in Online-Lernplattformen, da es in diesem Bereich nur wenig explizite Forschung gibt.

2 Grundlagen

Zunächst werden einige grundlegende Begriffe sowie Konzepte geklärt, die für das Verständnis der weiteren Arbeit wichtig sind.

2.1 Wirkung und Absicht von Feedback

Um die Bedeutung von Feedback genauer erläutern zu können, muss geklärt werden, was Feedback ist, und wo es vorkommt. Hattie und Timperley beschreiben Feedback als Konsequenz der eigenen Leistung [17, p. 81]. In einer Lernumgebung wird jegliche Information über die eigene Leistung bzw. das eigene Verständnis zum Feedback. Für eine lernende Person sind jegliche Formen der Konfrontation mit dem eigenen Wissen, sei es durch Kontrollen oder Gesprächen mit anderen, auch eine Anregung zur Reflexion. Die Lehrkraft muss hier entscheiden, wie mit diesem Phänomen in Hinsicht der Unterrichtsgestaltung umgegangen werden soll. Daraus ergeben sich zwei Fragen. Erstens, hat Feedback tatsächlich einen Einfluss auf den Lernprozess? Zweitens, welches Ziel wird durch Einbinden von Feedback angestrebt? Also in welchen Aspekten erhoffe ich mir eine Verbesserung des Lernprozesses durch Verwendung von Feedback?

In Artikeln wie [17] und [16] wurden Untersuchungen hinsichtlich der ersten Frage zusammengetragen. Feedback hat ganz eindeutig einen Effekt auf den Lernprozess. Je nach Art des Feedbacks variiert der Effekt, nicht nur darin, wie stark dieses ist, sondern auch, ob dieses positiv oder negativ ist. Die sich daraus ergebende Frage über die Natur von sinnvollem Feedback wird im nächsten Unterabschnitt diskutiert. Beim Aufbau eines Kurses, der aktiv versucht, Feedback einzubinden, müssen sich auch über die Ziele, die damit erreicht werden sollen, Gedanken gemacht werden. Dies gilt unabhängig davon, ob es sich um eine traditionellere Unterrichtsumgebung, wie ein Klassenraum, oder eine Online-Lerneinheit handelt. Wenn versucht werden soll, eine generelle Antwort auf die zweite Frage zu geben, müssen zwei Aspekte beachtet werden. Zum einen, was die anzustrebenden Ziele sind, zum anderen, ob die beteiligten Gruppen von Lernenden und Lehrenden sich darüber einig sind, was die Ziele sind.

Eine von Dawson et al. durchgeführte Befragung von Studierenden und Lehrkräften legt nahe, dass Übereinstimmung über den Hauptzweck von Feedback herrscht [7, p. 28]. Nämlich eine Verbesserung im Lernprozess zu erzielen. Bei der Frage nach einer genaueren Angabe waren die häufigsten Antworten eine Verbesserung in der Arbeitsweise und im Verständnis. In Hinsicht der Wichtigkeit dieses Ziels besteht auch Einigkeit zu anderen Artikeln ([17, p. 86], [22, p. 200],[2, p. 1f]). Ein weiteres Ziel, das auch genannt wird, ist das Ausdrücken von Affekten, also die Lernenden zu motivieren, zu ermutigen oder ihnen sonstige positive Gefühle zuzusprechen. Auch wenn in der von Dawson et al. durchgeführten Befragung der Ausdruck von Affekten deutlich weniger häufig angegeben

wurde, wird nochmal betont, dass diese durchaus eine wichtige Stellung im Feedback haben, wenn auch eine zweitrangige [7, p. 29]. Sonstige, wenn auch weniger häufig angegebene Ziele waren, dass Feedback eine Rechtfertigung der Bewertung darstellen soll und dass Feedback dazu da ist, die Stärken und Schwächen einer Person darzulegen.

Auch interessant zu betrachten ist, dass in einer Arbeit von Cavalcanti et al. die Ergebnisse von Forschungen zu automatisiert generiertem Feedback in Online-Lernumgebungen betrachtet wurden [2, p. 7]. Das Ziel, das am häufigsten genannt wurde, ist Feedback zu verwenden, um in einem spezifischen Thema Hilfestellung zu leisten. Unter anderem, um mitzuteilen, wo Fehler liegen, Tipps zu geben, wie diese behoben werden können und Lob bei richtigen Antworten auszugeben. Das nächst häufig genannte Ziel ist das der Selbstorganisation. Genauer gesagt Umgebungen zu schaffen, die zum Lernen motivieren, wie personalisiertes Feedback, Gamification (siehe Abschnitt 2.4) und Dashboards. Dazu kommt eine Analyse, um den Leistungsgrad zu ermitteln und entsprechend helfen zu können, in der Hoffnung Abbruchraten zu reduzieren. Deutlich weniger oft genannte Ziele sind Feedback zu verwenden, um die Lehrkraft zu unterstützen, und um dabei zu helfen, Plagiate zu reduzieren.

2.2 Festlegung was gutes Feedback ausmacht

In einem idealen Szenario nimmt eine Online-Lernumgebung der Lehrkraft viel Arbeit bei der Vergabe von Feedback ab. Um das Potenzial von Online-Lernumgebungen vollständig ausnutzen zu können, müssen diese möglichst autonom in der Lage dazu sein, Lernenden wichtige Hinweise zur Verbesserung der Selbstregulierung zu geben. Die Lernenden sollen diese Systeme nutzen, um sich eigenständig neue Inhalte aneignen zu können. Nicol und Macfarlane-Dick haben sich mit Feedback, das Selbstregulierung fördern soll, auseinandergesetzt und sieben Prinzipien, was gutes Feedback ausmacht, aufgestellt [22]. In diesem Abschnitt sollen diese Prinzipien kurz vorgestellt, sowie ein Vergleich mit dem Ansatz den Hattie und Timperley [17] präsentiert werden. Zunächst folgt eine Zusammenfassung der sieben Prinzipien nach Nicol und Macfarlane-Dick:

1. Feedback soll klarstellen, was eine gute Leistung ist. Um die erwarteten Leistungen erreichen und den eigenen Fortschritt einschätzen zu können, müssen die Erwartungen über diese so weit es geht abgeglichen werden. Einer der angegebenen Lösungsansätze, der sich gut in eine Online-Lernumgebung integrieren ließe, wäre eine genaue Anforderungsangabe, die in unterschiedlichen Leistungsstufen unterteilt ist [22, p. 206].
2. Feedback fördert Selbstreflexion über das Lernen. Damit dies gewährleistet werden kann, müssen die Lernenden häufig eine Selbsteinschätzung durchführen. Dafür ist relevant, dass ein Verständnis zwischen der erwarteten und der eigenen Leistung vorherrscht. Einer der Umsetzungsvorschläge, der mit einer Online-Lernumgebung vereinbar wäre, ist am Ende einer Aufgabe über die erreichten und kommenden Meilensteine zu reflektieren [22, p. 207f.].
3. Feedback soll qualitative Informationen über das Lernverhalten liefern. Das heißt den Lernenden die Möglichkeit geben, sich auf eine Fehlersuche über ihre eigene

Leistung zu begeben und diese zu verbessern. Vorschläge, die sich in einer Online-Lernumgebung einarbeiten ließen, sind eine gezielte Minimierung von Feedback, damit dieses besser wahrgenommen wird, wiederholbare Testfragen und eine Priorisierung von Bereichen, in denen man sich verbessern könnte [22, p. 209f.].

4. Feedback regt zum Gespräch über das Lernen mit Gleichgesinnten und Lehrkräften an. Um sicherstellen zu können, dass das Feedback tatsächlich verstanden wurde, sollte es in Form eines Dialogs stattfinden. Die Vorschläge, dass sich Lernende Peer-Reviews geben und dass sie Feedback Kommentare als nützlich oder nicht nützlich erklären können, wären in Online-Lernumgebungen umsetzbar [22, p. 210].
5. Feedback fördert die Motivation und das Selbstbewusstsein. Wenn Lehrkräfte Feedback geben, wirkt sich das auf die Ziele der Lernenden, sowie ihre Einstellung gegenüber diesen aus. Kurse sollten möglichst viele kleine Aufgaben enthalten, von denen nicht viel abhängt. Einer der gegebenen Vorschläge ist, automatisierte Tests mit Feedback zu verwenden, was sich in einer Online-Lernumgebung gut umsetzen ließe [22, p. 212].
6. Feedback hilft, die Lücke zwischen dem aktuellen und dem gewünschten Lernstand zu schließen. Auch wenn ein Feedback inhaltlich wertvoll sein kann, besteht die Möglichkeit, dass dieses trotzdem nicht richtig verstanden wird. Es muss eine Art Möglichkeit geben, zur Aufgabe zurückkehren zu können, damit sichergestellt werden kann, dass diese Inhalte korrekt vermittelt wurden. Zumindest müssten Aufgaben so strukturiert sein, dass Feedback in der vorherigen Aufgabe, für die Bearbeitung der folgenden Aufgabe hilft. Wenn es die Situation erlaubt, sind Simulationen, die möglichst sofort Feedback liefern, ebenfalls hilfreich. All diese Vorschläge müssten auch in einer Online-Lernumgebung umsetzbar sein [22, p. 213f.].
7. Feedback gibt einer Lehrkraft Informationen, die benutzt werden können, um die Lehrkonzepte anzupassen. Damit eine Lehrkraft so gut wie möglich helfen kann, muss diese auch ein gutes Verständnis des Lernverhaltens der Lernenden haben. Klassische Ansätze sind Durchführung von Umfragen oder Beobachtungen. In einer Online-Lernumgebung könnte auch das Erheben unterschiedlicher Daten implementiert werden, z. B. wie viel Zeit in einem Abschnitt verbracht wurde oder wie viele Fehler bei einer bestimmten Aufgabe gemacht wurden. Auch könnten, von einer geeigneten Implementation, viele der gebrachten Vorschläge umgesetzt werden. Lernende können bei einer Abgabe sagen, zu welchen Punkten sie Feedback erhalten möchten oder sie können angeben, welche der Aufgabenteile sie als besonders schwer empfanden [22, p. 214f.].

Diese sieben Prinzipien wurden mit dem Zweck aufgestellt, die Selbstregulierung Studierender zu verbessern. Feedback soll die Aufgabe haben, den aktuellen Zustand der Studierenden in Relation zu ihren Zielen zu setzen. Je effektiver der Prozess der Selbstregulierung laufen kann, desto besser kann das Feedback aufgenommen werden. In dem von Hattie und Timperley präsentierten Modell wird versucht, Feedback etwas allgemeiner zu untersuchen [17]. Ausgangspunkt ist die Annahme, dass Feedback auf die Absicht

fokussiert ist, die Lücke zwischen dem aktuellen und dem gewünschten Wissensstand zu schließen. Dies kann selbstständig von Seite der Schülerinnen und Schüler geschehen, durch Verbesserung des Arbeitsaufwands oder der Effektivität. Jedoch gibt es auch schlechte Ansätze, um die Lücke, zumindest nach der eigenen Wahrnehmung, zu schließen. Beispielsweise können die Ziele aufgegeben oder der eigene Standard erheblich verringert werden. Es kann auch vorkommen, dass die Ziele ineinander verschwimmen und somit nur manche Ziele verfolgt, während andere ignoriert werden [17, p. 86f.]. Leider wird auch häufig erwartet, dass die Aufgabe Feedback zu vermitteln einzig und allein die Verantwortung der Lehrkraft sei, allerdings funktioniert Feedback am besten, wenn sich beide Seiten darum bemühen [17, p. 88]. Selbstverständlich können und sollen die Lehrkräfte hier Unterstützung leisten, mitunter durch Spezifizierung der Ziele. Bis hierhin wird, ähnlich wie bei Nicol und Macfarlane-Dick, einer angemessenen Zielsetzung eine große Bedeutung zugeschrieben. Von hier aus versuchen Hattie und Timperley die Natur von Feedback zu betrachten, um zu bestimmen, was gutes Feedback ausmacht. Neben ihrer Erklärung haben Hattie und Timperley ihren Ansatz auch als Grafik zusammengefasst, welche für diese Arbeit übersetzt wurde und als Abbildung 2.1 zu sehen ist. Nach ihrem Vorschlag kann jedes Feedback auf die folgende Art und Weise untersucht werden. Feedback kann nur auf bereits vermitteltes Wissen aufbauen und muss, um effektiv sein zu können, in der Lage sein, diese drei Fragen beantworten zu können:

- Was ist mein Ziel? Also eine klare Darstellung der vorhandenen Ziele.
- Wie gut bin ich auf dem Weg? Genauer, eine Veranschaulichung des Fortschritts auf dem Weg zum Ziel.
- Wo muss ich als Nächstes hin? Also alle Aktivitäten, die für einen besseren Fortschritt notwendig sind.

Dabei ist auch nicht gedacht, dass die Fragen isoliert voneinander zu betrachten sind, sondern auch wie diese ineinander führen können. Danach wird auf den Fokus des Feedbacks eingegangen. Genauer gesagt, wird Feedback in eine der folgenden vier Ebenen eingeordnet:

- Die Aufgaben Ebene: Kann das Feedback klarstellen, wie gut die Aufgabe verstanden wurde? Hier werden Anweisungen zum Erhalt von richtigen, mehr oder anderen Informationen vermittelt.
- Die Prozess Ebene: Ist wegen des Feedbacks der Hauptprozess vermittelt worden, der für das Verständnis der Aufgabe notwendig ist? Hier geht es darum, der Lernprozess oder den Ansatz zum Vervollständigen der Aufgaben zu verbessern.
- Die Selbstregulations Ebene: Fördert das Feedback das selbstständige Beobachten und das Regulieren der eigenen Handlungen? Hauptziel ist es, selbstständige Evaluation zu fördern und Zuversicht aufzubauen, selbständig weitere Aufgaben anzugehen.
- Die persönliche Ebene: Regt das Feedback zu Reflexion an und fördert es Affekte des Lernenden über sich selbst? Dieses Feedback richtet sich persönlich an Schülerinnen und Schüler und wendet sich an Affekte, nicht an eine Leistung in Bezug zur Aufgabe.

Wichtig ist hier, dass nicht alle Ebenen gleich effektiv sind. Besonders bei der persönlichen Ebene ist es zweifelhaft, ob diese überhaupt einen Effekt hat. Sie ist auf jeden Fall von den Schülerinnen und Schülern in irgendeiner Form gewünscht, tatsächlich scheint es aber keine starken Hinweise auf eine Beeinflussung des Lernprozesses zu geben. Die anderen drei Ebenen haben, abhängig von Kontexten, die Möglichkeit positive Effekte hervorzubringen und gehören entsprechend auch in dieser Hinsicht beachtet. Gerade der Selbstregulations Ebene wird viel Bedeutung zugeschrieben, da diese auch zu einer intrinsisch motivierten Verfolgung der Ziele führen kann.

Auch wenn sich die dahinter liegenden Absichten unterscheiden, bestehen bei beiden Ansätzen Ähnlichkeiten über das, was für Feedback wichtig ist. Bei beiden Arbeiten wird großen Wert auf die Förderung von Selbstorganisation gelegt. Während Hattie und Timperley versuchen eine allgemeine Struktur von Feedback aufzustellen, stellen Nicol und Macfarlane-Dick direkte Richtlinien auf. Bei beiden Arbeiten wird auch die Limitation aufgeführt, dass dies nur grobe Ansätze sind, die als einen Startpunkt gesehen werden sollen. Da aber in beiden ein Wert liegt, können auch beide bei der Gestaltung von Feedback beachtet und miteinbezogen werden.

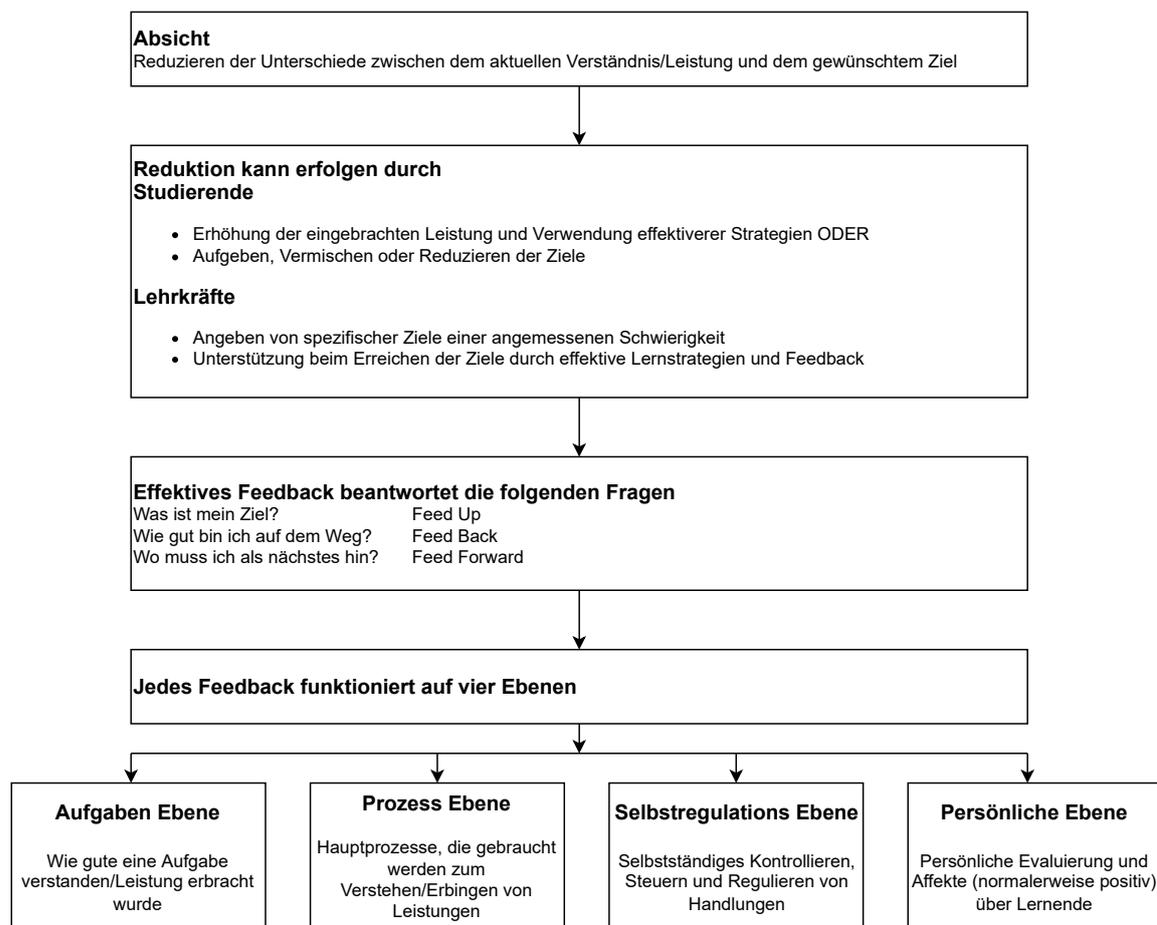


Abbildung 2.1: Zusammenfassung der Struktur von Feedback nach Hattie und Timperley, übersetzt entnommen aus [17]

2.3 Methoden zur Vermittlung von Feedback

Nachdem sowohl die Ziele von Feedback, als auch die vorgeschlagenen Grundprinzipien von gutem Feedback präsentiert wurden, bleibt noch immer die Frage über die konkrete Umsetzung offen. Die folgende Tabelle 2.1 soll eine kleine Übersicht darstellen, welche basierend auf folgender Literatur([7],[2],[5]) zusammengestellt wurde. Auch wenn manche der genannten Methoden übereinstimmen, unterscheidet sich die Motivation in der Literatur, die nun vorab genannt wird. Calvancanti et al. haben Literatur über die automatische Ausgabe von Feedback untersucht und zusammengetragen, welche Techniken für die automatisierte Feedback Generierung verwendet werden [2, p. 7]. Costello und Crane haben Möglichkeiten von digitalem Feedback für lernende Personen untersucht [5, p. 217ff]. Dawson et al. haben Lehrkräfte und Studierende zu mehreren Aspekten von Feedback befragt, unter anderem auch, was die bevorzugten Methoden sind, Feedback zu vermitteln [7, p. 30].

Feedback Methode	Erwähnt in	Mögliche Onlineumsetzung
Automatische Bewertung, Vergleich mit gewünschter Lösung	[2],[5]	Anzeigen des gewünschten Ergebnisses zum Abgleich
Peer Feedback	[5],[7]	Hochladen und Teilen von Dateien
Schriftliche Kommentare	[5],[7]	Kommunikationssysteme
Mündliche Kommentare	[5], [7]	Hochladen von Audio- oder Videodateien
Automatisierte Kommentare	[5]	Festlegen eines Systems
Online-Umgebung (Kommunikation, e-Portfolio)	[2], [5]	Einrichten notwendiger Webinfrastruktur
Tests und Quiz	[5]	Festlegen von System
Ermittlung wichtiger Themen	[2]	Einbinden von Natural Language Processing
Sticker	[5]	Errungenschaftssysteme

Tabelle 2.1: Feedback Methoden und mögliche Umsetzungen in Online-Umgebungen

2.4 Gamification

Gamification ist ein Term, der die Verwendung von Elementen aus Spielen in einem nicht-Spiel Kontext beschreibt [8]. In den letzten Jahren wird zunehmend versucht, Gamification in einem Lehrkontext einzubinden. Auch wenn die Forschung über die Auswirkung von Gamification im Lehrbereich noch in ihren Anfängen ist, liegt die Vermutung nahe, dass die Verwendung von Gamification einen positiven Effekt hat ([26, p. 106f], [9, p. 83]). Neben der Steigerung von Motivation kann der Einsatz von Gamification Elementen in einem Kurs den Stress reduzieren, der durch Interaktion mit Technik, wie beispielsweise Schulrechnern, verursacht werden kann [11]. Wenn man sich die in 2.1 präsentierte

Definition von Feedback anschaut, liegt es nahe Gamification Elemente auch als Methoden für Feedback einzuordnen. Denn alle Elemente basieren darauf, eine Konsequenz der eigenen Leistung zu sein und sind in der Lage, die Unterschiede zwischen dem aktuellen Verständnis und dem gewünschten Ziel zu veranschaulichen. Nah et al. haben in einer Zusammenfassung mehrerer Studien typische Gamification Elemente aufgelistet [7], die in der Übersichtstabelle 2.2 abgebildet sind.

Gamification Element	Zweck
(Erfahrungs-)Punkte	Zeigen den Fortschritt an und können als Währung verwendet werden
Level	Kapseln einzelne Aufgaben voneinander ab und ermöglichen eine Schwierigkeitsangabe
Abzeichen	Sollen eine Belohnung für besondere, sowie Motivation für zukünftige Leistungen darstellen
Ranglisten	Wettbewerbselement zur Motivation
Belohnungen	Je nach Umsetzung des Systems als zusätzliche extrinsische Motivation
Fortschrittsbalken	Erzeugen Motivation in dem sie darüber informieren, wie nah man an einem Ziel ist
Rahmenhandlung	Durch Erzeugen eines Kontextes für die Aufgabe soll das Interesse an der Bearbeitung erhöht werden
Regelmäßige Rückmeldung	Verbessert die Einbindung des Spielers in das Element mit dem es interagiert

Tabelle 2.2: Typische Gamification Elemente nach Nah et al. [21]

2.5 Lernmanagement-Systeme

In diesem Abschnitt werden Funktionen und Merkmale von Lernmanagement-Systemen(LMS) eingeführt. Piña hat in einem Artikel Merkmale, Vorteile und Beispiele von LMS zusammengefasst, die hier kurz aufgegriffen werden [25]. Ein LMS ist eine Plattform, die dazu dient, Kurse sowie Interaktionen in diesen zu verwalten. Außerdem dient sie auch als Plattform für die Übermittlung von Inhalten. Lehrkräfte können Texte, Video und Audiodateien hochladen, sowie externe Quellen verlinken, die zum Lernen verwendet werden können. Zusätzlich kann ein LMS als eine Kommunikationsplattform verwendet werden. Über Foren wird eine Interaktion zwischen Teilnehmenden und Lehrenden oder auch zwischen Teilnehmern unter sich ermöglicht. Zudem werden oft auch private Nachrichtensysteme, ähnlich wie mit E-Mails oder Text Chats, angeboten. Zusätzlich gibt es typischerweise auch Elemente zur Leistungseinschätzung mit der Möglichkeit Feedback zu geben. Mögliche Beispiele sind das Einreichen von Hausaufgaben mit Abgabefristen oder automatisierte Tests und Umfragen. Schließlich gibt es noch Werkzeuge zur Kursverwaltung, also die Möglichkeit Teilnehmer zu gruppieren und festzulegen, ob diese bestanden haben, sowie eine generelle Festlegung der Zugriffsrechte einzelner Teilnehmer. Typischerweise haben

viele LMS auch eine Option Plugins zu erstellen und herunterzuladen, damit eine Erweiterbarkeit und Zuschnitt auf den eigenen Bedarf gewährleistet werden kann. Beispiele für Open-Source LMS wären Moodle ¹ und Illias ².

2.6 WordPress Grundlagen

WordPress ist ein Content-Management-System, welches erlaubt, eigene Inhalte als Webseite zu veröffentlichen. Das System ist Open Source und verwendet PHP als Programmiersprache. Für das Erstellen von Webseiten sind aber auch HTML, CSS und Javascript integriert. Außerdem gehört zu jeder Webseite auch eine Datenbank. Sollte der Wunsch nach zusätzlichen Funktionalitäten bestehen, wird jedoch erwartet, dass der Code von WordPress, der sogenannte WordPress Core, unverändert gelassen wird. Jegliche Änderungen werden bei einem Update überschrieben und gehen verloren. Für das Hinzufügen weiterer Funktionalität wird ein Plugin-System verwendet (siehe 2.6.2). Diese können über *Hooks* (2.6.3) sich in das System einhaken und Funktionalität hinzufügen. Viele Plugins sind online über eine integrierte Ablage verfügbar und können direkt in die eigene WordPress Seite eingebunden werden. Aus genau den gleichen Gründen wie für den WordPress Core, sollen nicht selbst erstellte Plugins, unmodifiziert bleiben. Analog lassen sich aber auch Plugins für Plugins erstellen. Wichtig zu beachten ist, dass Funktionen und *Hooks* auf einer Seite global verfügbar sind und daher auch einen eindeutigen Bezeichner brauchen, um Konflikte zu vermeiden.

2.6.1 Posts

Posts sind Einträge auf der eigenen Webseite, die mit einem Zeitraum assoziiert sind. Sie können in Kategorien eingeteilt werden, haben einen assoziierten Autor und werden oft für soziale Interaktion verwendet. Eine weitere Kategorie von Einträgen sind *Pages*, welche all diese Dinge nicht haben und statische Aspekte der Seite enthalten. Beispiele hierfür sind die Startseite, eine „Über uns“ Seite oder eine Kontaktseite. Außerdem ist noch wichtig zu wissen, dass WordPress Daten über *Posts* speichert. Mit der im Listing 2.1 abgebildete Methode können diese im Code aufgerufen und verwaltet werden.

```
1 get_post( int|WP_Post|null $post = null, string $output = OBJECT,  
2 string $filter = 'raw' )
```

Listing 2.1: Methode zum Aufrufen der Daten eines *Posts*

Hier kann eine *Post id* oder ein *Post Objekt* eingegeben werden und es werden die dazugehörigen Daten ausgegeben. Überdies können noch ein Typ für die Ausgabe, sowie einige mögliche Filteroptionen angegeben werden. Neben den normalen Daten gibt es auch noch Metadaten zu jedem Post. Diese enthalten eher administrative Informationen über den Post. In Listing 2.2 ist die Funktion, mit der die Metadaten zu einem Post aufgerufen werden, zu sehen.

¹<https://moodle.org/>

²<https://www.ilias.de/>

```
1 get_post_meta( int $post_id, string $key = '', bool $single = false )
```

Listing 2.2: Methode zum Aufrufen der Metadaten eines *Posts*

Hier muss die Id des *Posts* angegeben werden. Wenn nicht alle Metadaten gewünscht sind, kann auch ein Key spezifiziert werden, damit nur die dazugehörigen Werte ausgegeben werden. Außerdem kann auch spezifiziert werden, ob nur der erste, zum Key gehörende Wert ausgegeben werden soll. Der Zugriff auf die Metadaten ist auch deshalb wichtig, weil über diese vorab getroffene Einstellungen aufgerufen werden können.

2.6.2 WordPress Plugins

Ein Plugin stellt in WordPress eine Erweiterung für den vorhanden Code der Webseite dar. Dieser kann an spezifischen Stellen eingreifen und Modifikationen ausführen. Diese Modifikationen können an WordPress direkt, aber auch an wiederum andere Plugins gerichtet sein. Idealerweise können diese ohne Probleme nach Bedarf aktiviert und deaktiviert, sowie mit anderen Plugins kombiniert werden. Es ist aber wichtig zu beachten, dass WordPress alle Datei- und Methodennamen global betrachtet. Wenn diese also nicht eindeutig genug sind, kann es bei einer gleichzeitigen Aktivierung zu Fehlern oder im Fall von *Hooks* zu unerwünschten Seiteneffekten kommen. WordPress stellt eine große Ablage mit Plugins zur Verfügung, auf die Plugins hoch- und heruntergeladen werden können. Für einen spezifischen Bedarf können aber auch eigene Plugins erstellt werden.

Das Wichtigste beim Erstellen eines Plugins ist eine grundlegende PHP-Datei, die den gleichen Namen wie das Verzeichnis haben soll, in dem alle Plugin Dateien sein werden. Hier müssen auch mindestens der Plugin Name und eine Beschreibung innerhalb eines Kommentars definiert werden, wie im Listing 2.3 gesehen werden kann. Diese Informationen werden dann in den Plugin Einstellungen angezeigt. Es können aber noch weitere Details, wie beispielsweise Autoren, definiert werden.

```
1 /**
2  * Plugin Name: MasterStudy LMS - Rockstartit Masterstudy integration
3  * Description: Add additional features to masterstudy for use in the Rockstartit project
4  * Author: Elias Kia
5  * Version: 1.0
6  */
```

Listing 2.3: Headerkommentar mit Informationen über das Plugin

Sollte es sich um ein kleines Plugin handeln können hier auch gleich alle weiteren Funktionalitäten hinzugefügt werden, andernfalls werden über diese Datei alle weiteren Dateien eingebunden. Dazu muss zunächst der Pfad zu dem Plugin definiert werden. Danach wird dieser beim Einbinden der weiteren Dateien verwendet. In Listing 2.4 ist abgebildet, wie dies aussehen kann.

```
1 // Plugin path
2 define( 'ROCKSTARTIT_MASTERSTUDY_DIR', plugin_dir_path( __FILE__ ) );
3
4 // Load necessary php files
5 require_once ROCKSTARTIT_MASTERSTUDY_DIR . '
    rockstartit_masterstudy_integration_fetch_post_data.php';
```

Listing 2.4: Angabe des Pfads und Einbinden weiterer Dateien

2.6.3 Hooks

Hooks stellen eine Möglichkeit dar, zu einem Plugin zusätzliche Funktionalität hinzuzufügen, ohne das Plugin selbst modifizieren zu müssen. Eigene Funktionen werden in diesem Zusammenhang *Callables* genannt, und können zu bereits vorhandenen *Hooks* registriert werden. Sie werden dann beim Auslösen eines *Hooks* aufgerufen. Es können auch Parameter im *Hook* festgelegt werden, die von den registrierten Funktionen genutzt werden können. Die zwei Arten von *Hooks*, die es gibt, sind *Actions* und *Filters*. Beide realisieren eine Rückruffunktion. Sie werden jedoch für unterschiedliche Zwecke verwendet.

Actions rufen registrierte Funktionen auf, die Code ausführen, ohne danach etwas an den *Action Hook* zurückzugeben. Sie stellen einen Zeitpunkt dar, an dem der ausgeführte Code für etwas anderes unterbrochen werden kann. Durch Aufruf der Funktion, die in Listing 2.5 zu sehen ist, kann eine Funktion registriert werden.

```
1 add_action( string $hook_name, callable $callback, int $priority = 10,
2 int $accepted_args = 1 )
```

Listing 2.5: Registrieren einer Funktion zu einem *Action Hook*

Der Name des *Action Hooks*, sowie der Funktion müssen angegeben werden. Sowohl Priorität (kleinere Prioritäten werden zuerst ausgeführt), als auch Anzahl übergebener Argumente, sind optionale Parameter. Um in einem eigenen Plugin einen *Action Hook* auszulösen, wird die in Listing 2.6 abgebildete Funktion aufgerufen.

```
1 do_action( string $hook_name, mixed $arg )
```

Listing 2.6: Auslösen eines *Action Hooks*

Nur der Name des *Action Hooks* muss angegeben werden. Optional können noch weitere Argumente angegeben werden, die an registrierte Funktionen weitergegeben werden können. Ein exemplarischer Ablauf für das Registrieren und auslösen eines *Action Hooks* kann in Abbildung 2.2 gesehen werden.

Filter sollen hingegen eine Gelegenheit darstellen, bei der an Daten, die gleich verwendet werden sollen, vorab noch einmal Modifikationen durchgeführt werden können. *Filter* sollten keine Seiteneffekte haben und nur den übergebenen Wert bearbeiten und anschließend an den aufrufenden *Filter Hook* zurückgeben. Zur Registrierung zu einem *Filter* wird die in Listing 2.7 dargestellte Funktion aufgerufen, die so wie das Registrieren von *Actions* funktioniert.

```
1 add_filter( string $hook_name, callable $callback, int $priority = 10,
2 int $accepted_args = 1 )
```

Listing 2.7: Registrieren einer Funktion zu einem *Filter Hook*

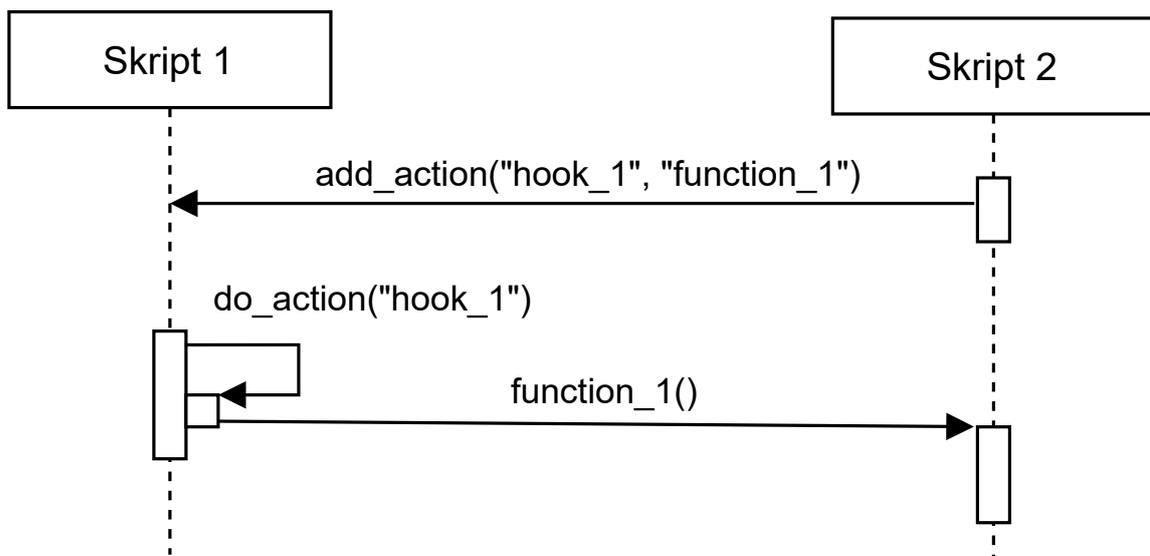


Abbildung 2.2: Die Funktion *function_1* aus Skript zwei wird zum *Action Hook hook_1* registriert, der dann in Skript eins ausgelöst und *function_1* damit aufgerufen wird.

Das Auslösen von einem eigenen *Filter Hook* funktioniert auch ähnlich wie bei *Actions*. Als Unterschied muss aber noch der zu filternde Wert angegeben werden. Dies ist in Listing 2.8 zu sehen.

```
1 apply_filters( string $hook_name, mixed $value, mixed $args )
```

Listing 2.8: Auslösen eines *Filter Hooks*

Insgesamt sind *Hooks* eine einfache und übersichtliche Möglichkeit, um die Funktionalität von vorhandenen Plugins zu erweitern, ohne diese direkt modifizieren zu müssen. Sie sind außerdem von WordPress für genau diesen Zweck entworfen worden, da vorhandener Code nicht direkt modifiziert werden soll. Der eine große Nachteil ist, dass damit eine vollkommene Abhängigkeit zu den Entwicklern des jeweiligen Plugins entsteht. Wenn diese keine passenden *Hooks* zur Verfügung stellen, kann keine Funktionalität hinzugefügt werden, ohne den Code direkt zu modifizieren.

2.6.4 Rockstartit

Die Webseite „Rockstartit“, welche über WordPress aufgebaut wurde, bietet interdisziplinäre Kurse für Schülerinnen und Schüler in den Klassen sieben bis neun an. Die Kurse auf der Webseite sollen im Rahmen von Unterrichtsstunden an der Schule als Unterstützung für die Schülerinnen und Schüler, sowie zur Entlastung der Lehrkräfte verwendet werden können. Es werden über andere aktuelle Themen, wie beispielsweise unsere Umwelt oder Klimaschutz, Zugänge zur Informatik geboten. Ziel ist es also zu zeigen, dass Informatik eine große Menge an Anknüpfungspunkten zu anderen Bereichen hat und damit den Lernprozess zu unterstützen. Zu einem Überthema gibt es dann mehrere Lehreinheiten, an denen gezeigt werden kann, wie viele Einsatzgebiete die Informatik hat. Da die Webseite

über WordPress läuft, werden Plugins benötigt, um weitere Funktionalitäten hinzufügen zu können.

2.6.5 MasterStudy LMS

„MasterStudy LMS“ ist ein von StylemixThemes entwickeltes Plugin für WordPress, welches LMS Funktionalitäten hinzufügt. In erster Linie erlaubt das Plugin einem Kurse zu erstellen, an denen angemeldete Nutzer teilnehmen können. Ein Kurs stellt für WordPress einen *Post* dar und setzt sich wiederum aus Lektionen zusammen, die für WordPress ebenfalls als *Post* dargestellt werden. Um die beiden intern voneinander unterscheiden können, werden Kurse über eine sogenannte *Postid* angesprochen und Lektionen über eine *Itemid*. Die Namen dienen nur zur eindeutigen Unterscheidung im Plugin. Für WordPress sind beides *Posts*. Ein Kurs ist also ein *Post*, welcher wiederum eine Liste von anderen *Posts* enthält. Abbildung 2.3 stellt dies grafisch dar. Nur über das Plugin wird beides als ein Kurs mit einer Liste von Lektionen interpretiert. Ein Nutzer kann dann über den Kurs durch die darin definierte Liste von Lektionen gehen. Diese Entkopplung ermöglicht, dass die gleichen Lektionen auch in unterschiedlichen Kursen verwendet werden können. Außerdem kann für eine Lektion eine Probeeinstellung aktiviert werden, dann kann diese betrachtet werden, ohne dass man angemeldet sein muss. Ist diese Einstellung aktiv, wird jedoch die Navigationsleiste am Ende der Lektion deaktiviert, was eine Navigation umständlich gestaltet. Zusätzlich kann auch eingestellt werden, dass für einen Kurs bezahlt werden muss. Weitere Einstellungen, beispielsweise wie die Bezahlung ablaufen soll, sind ebenfalls vorhanden, aber im Rahmen dieser Arbeit nicht von Relevanz.

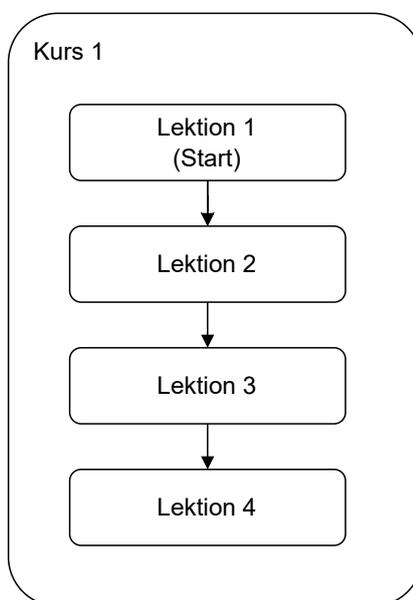


Abbildung 2.3: Ein Kurs mit vier Lektionen. Für WordPress sind sowohl Kurs als auch Lektion *Posts*.

3 Verwandte Arbeiten

Nachdem in Kapitel 2 die Grundlagen präsentiert wurden, sollen im Folgenden zunächst Arbeiten über Feedback betrachtet werden. Einführend werden Arbeiten, welche die auch hier verwendeten Elemente untersucht haben, präsentiert. Im Anschluss werden andere Implementierungen von Feedback-Systemen gezeigt. Es werden zunächst Ziel und Vorgehensweise kurz zusammengefasst. Danach werden die Unterschiede zu dieser Arbeit betrachtet.

3.1 Arbeiten zu Fortschrittsbalken

Ein Fortschrittsbalken gibt in irgendeiner Form Informationen über den Zustand eines laufenden Prozesses ab ([13, p. 7], [19, p. 1], [23, p. 316]). In manchen Arbeiten wird dieser daher auch als Schnittstelle zwischen Mensch und Maschine bezeichnet ([19, p. 1], [23, p. 316]). Ein häufig genannter Anwendungszweck für Fortschrittsbalken ist, dass dieser durch Einblick in den Verlauf eines Prozesses, die Wahrnehmung beeinflussen soll ([3, p. 417], [13, p. 7], [14, p. 863], [15, p. 115], [23, p. 316]). Typischerweise ist hiermit gemeint einen Nutzer zu beruhigen bzw. diesem die Ängste darüber zu nehmen, dass ein System feststecken würde, indem der Fortschritt dargestellt wird. Viele Untersuchungen finden in einem der typischen Kontexte eines Ladevorgangs statt. Es wurde untersucht, ob unterschiedliche Merkmale eines Fortschrittsbalkens selbst Einfluss auf die Wahrnehmung über die Geschwindigkeit ausüben kann. Unter anderem wurden Farbe [14], Länge und Dicke [23], sowie Form ([19], [23]) untersucht. Zwar gibt es Hinweise darauf, dass ringförmige Fortschrittsbalken als schneller wahrgenommen werden, es konnten aber noch keine starken Zusammenhänge zwischen äußeren Merkmalen von Fortschrittsbalken und der Wahrnehmung über die Geschwindigkeit gezeigt werden.

Eine weitere Art von Untersuchung bestand darin, die Wahrnehmung der Geschwindigkeit im Zusammenhang mit dem Füllverhalten zu betrachten ([3], [13], [15]). In allen hier genannten Arbeiten wurden Fortschrittsbalken untersucht, die sich linear, beschleunigend und entschleunigend füllen. Mit beschleunigend ist gemeint, dass der Balken sich zunächst langsam füllt und dann immer schneller wird, während entschleunigend das gegenteilige Verhalten ist. Ein sich linear füllender Fortschrittsbalken hat eine konstante Zunahme. Über die Wahrnehmung der Zeit konnte nicht wirklich etwas gezeigt werden, aber es wurden unterschiedliche Präferenzen über das Füllverhalten der Fortschrittsbalken genannt. Zwischen den zwei Studien, die sich mit der aktiven Betrachtung von sich füllenden Balken auseinandergesetzt haben, erzielte in einer der beschleunigende [15, p. 117] und in der anderen der entschleunigende [13, p. 11f] Balken eine bessere Wirkung. Conrad et al. haben das Füllverhalten im Rahmen der Bearbeitung von Online Umfragen untersucht [3, p. 425f]. Hier schienen beschleunigende Balken besonders schlecht anzukommen und zu

erhöhten Abbruchraten zu führen. Beim Rest konnten keine großen Unterschiede zum gänzlichen Auslassen eines Fortschrittsbalkens aufgezeigt werden. Es gab jedoch Hinweise darauf, dass bei einem entschleunigenden Balken, die Toleranz für komplizierte Aufgaben anstieg.

Allerdings ist der hier untersuchte Kontext nicht zutreffend auf den dieser Arbeit. In anderen Arbeiten, die unter anderem Fortschrittsbalken im Rahmen der Anwendung in Unterrichtseinheiten untersucht haben, schienen diese gut anzukommen. Bei Ding et al. wurden neben Abzeichen, Avataren und der Möglichkeit Forumsbeiträgen einen „Daumen nach oben“ zu geben, Fortschrittsbalken als zweitbeliebtestes Element genannt. Sie halfen dabei den eigenen Fortschritt zu beobachten, sowie das Gefühl zu vermitteln bereits etwas erreicht zu haben [10, p. 137]. Olsson et al. kamen zu ähnlichen Ergebnissen über die positive Reaktion zu Fortschrittsbalken [24, p. 451f]. Als Vorteil wurde auch genannt, dass nicht nur Schülerinnen und Schülern geholfen wird den Überblick zu behalten, sondern auch Lehrkräften. Sanchez et al. beschreiben auch, dass Fortschrittsbalken helfen einen Überblick zu schaffen und ein Gefühl für die Nähe zum Ziel zu vermitteln [27, p. 5f]. Grazia-Marquez und Bauer konnten hingegen keinen Einfluss von Fortschrittsbalken festhalten [12, p. 410f]. Allerdings haben alle vorliegenden Arbeiten gezeigt, dass Fortschrittsbalken keine negativen Auswirkungen auf Lernende hatten. Möglicherweise blieben diese unbeachtet, hatten aber auf manche einen positiven Einfluss.

3.2 Arbeiten zur Effektivität von Kommentaren als Feedback

Wie bereits im Abschnitt 2.1 beschrieben, kann nach Definitionen von Feedback, wie denen von Hattie und Timperley, Feedback als ein ziemlich umfassender Begriff verstanden werden. Typischerweise wird Feedback jedoch als irgendeine Form eines Kommentars präsentiert. Auch bei dem Aufstellen der vier Ebenen von Feedback scheinen Hattie und Timperley Kommentare als Feedback im Sinn gehabt zu haben [17, p. 84 ff]. Da die Unterschiede in der Effektivität der vier Ebenen bereits in Abschnitt 2.2 präsentiert wurden, soll im Folgenden ein weiterer Ansatz in der Forschung betrachtet werden.

Bei der Betrachtung der Effektivität von Feedback kommt es oft zur Einschränkung auf den Einsatz nach einer Art von Test. Mehrere Arbeiten kategorisieren dann das Feedback in Korrektheit der Lösung (KR), Anzeigen der korrekten Lösung (KCR) und elaboriertem Feedback (EF) ([1, p. 155], [6, p. 590], [29, p. 482], [30, p. 834]). EF kann dabei jegliche Art von Kommentar darstellen, der ausführlicher ist als KR und KCR. Grundsätzlich gelangen Arbeiten in diesem Bereich auch zum Schluss, dass EF besser ist als KR und KCR, aber es gibt ein paar Feinheiten auf die es sich lohnt zu achten. Vasilyeva et al. haben untersucht, dass die Effektivität von EF möglicherweise abhängig vom Lerntyp ist [30, p. 836ff]. Außerdem scheint Motivation auch eine Rolle zu spielen. Attali und Van der Kleij haben zusammenhängende Aufgaben untersucht und Hinweise darauf gefunden, dass EF nur Wirkung zeigte, wenn die erste Aufgabe falsch beantwortet wurde [1, p. 165]. Dies passt auch zu den Folgerungen von Maier et al., dass Feedback nur Wirkung gezeigt hat, wenn die Schülerinnen und Schüler dieses als hilfreich empfanden [20, p. 95]. Maier et al. halten obendrein fest, dass zu detaillierte Kommentare auch oft nicht gelesen wurden, aber

das EF hilfreich darin war, tiefere Zusammenhänge zu erläutern. Dies stimmt ebenfalls mit den Ergebnissen von Van der Kleij et. al überein [29, p. 501].

Weiter wurde von manchen auch der richtige Zeitpunkt für Feedback untersucht ([1, p. 166], [18], [29, p. 502]). Genauer, ob es besser wäre Feedback sofort nach dem Lösen einer Aufgabe zu geben oder mit einer Verzögerung. Die genaue Definition einer Verzögerung unterscheidet sich zwischen den Arbeiten. Allgemein kann aber darunter verstanden werden, dass das Feedback nicht unmittelbar auf die Vergabe einer Antwort folgt. Van der Kleij et al. konnten keinen Zusammenhang zwischen der erzielten Leistung und dem Zeitpunkt des Feedbacks feststellen [29, p. 502]. In einer anderen Arbeit sind Attali und Van der Kleij, aber zum Schluss gekommen, dass bei einer richtigen Antwort ein Feedback mit einer Verzögerung und bei einer falschen Antwort ein sofortiges Anzeigen des Feedbacks eine bessere Wirkung hat [1, p. 166]. Auch wenn Lehrkräfte scheinbar eher zu Feedback mit Verzögerung tendieren, haben Lefevre und Cox untersucht, was Studierende bevorzugen und sind zum Ergebnis gekommen, dass diese eigentlich sofortiges Feedback wollen [18, p. 1364]. Einen ähnlichen Meinungsunterschied zwischen Studierenden und Lehrkräften konnten auch Dawson et al. in einer Befragung feststellen [7, p. 30]. Auch wenn der richtige Zeitpunkt für Feedback umstrittener ist, als die Anwendung von EF, sollte beides berücksichtigt werden.

3.3 System zur Feedbackausgabe an Gruppen

In ihrer Arbeit [4] beschreiben Constapel et al. ihr System namens IKARion. Dieses soll Hilfe bei Gruppenarbeiten in Online unterstützten Lernumgebungen bieten. Das System arbeitet über drei Hauptkomponenten. Als Erstes gibt es die Schnittstelle zum Lernmanagement-System, in Form eines Plugins. In der Arbeit wird konkret Moodle ¹ als genutztes Lernmanagement-System erwähnt. Die Kursmitglieder sollen auch nur über Moodle mit dem System interagieren. Wenn das System feststellt, dass die Kursmitglieder Feedback benötigen, bekommen sie dieses in Form von Nachrichten über Moodle. Zusammen mit einer zusätzlichen Erweiterung, die eine Umgebung für Gruppenarbeit festlegt, können die Daten von Moodle gesammelt und an die zweite Komponente, ein Learning Analytics Backend (LAB), weitergegeben werden. Dieses verwaltet und verarbeitet die gesammelten Daten. Lehrkräfte können auch über das LAB Statistiken über die Aktivitäten der Gruppen erhalten. Diese werden auch an die dritte Komponente, dem Regelbasierten Intervention System (RIS), weitergegeben. Hier werden über Regeln Situationen festgelegt, in denen das System eine Nachricht mit Feedback an eine Gruppe versenden soll. In der Arbeit werden als Maßeinheiten die Anzahl (sinnvoll) geschriebener Wörter, die Verteilung der Beteiligung einzelner Mitglieder und die vergangene Zeit seit der letzten Kommunikation genannt. Es wird jedoch auch beschrieben, dass Personen mit entsprechender Berechtigung diese Regeln anpassen können. Sollten diese Fälle auftreten, kann das System automatisch eine Nachricht mit passendem Feedback an die Gruppe versenden.

Die hier genannte Arbeit über IKARion unterscheidet sich in drei Aspekten von dieser Arbeit. Erstens sind die Ziele unterschiedlich. Das IKARion System, sowie das Feedback, welches es ausgibt, richtet sich an Gruppen. In dieser Arbeit soll es aber nicht um Feedback

¹<https://moodle.org/>

in Rahmen von Gruppendynamiken gehen. Das hier verwendete Feedback ist auf Interaktion mit dem Inhalt beschränkt und soll helfen, mögliche Wissenslücken zu schließen und die Teilnehmer zu motivieren. Zweitens wird IKARion als Plugin in Kombination mit dem Lernmanagement-System Moodle eingesetzt und soll dementsprechend Feedback über einen Zeitraum im Kontext eines größeren Kurses liefern. Rockstartit, für welches das Feedback hier entwickelt wird, läuft auf einer WordPress Basis und soll vorerst nicht die Rolle eines LMS übernehmen. Da Lehrkräfte sich entscheiden können, nur bestimmte Lehreinheiten in ihrem Unterricht zu verwenden, sollte jedes Feedback für sich auch immer nur zu einzelnen Lerneinheiten oder gar nur Teile davon gehören. Als einen momentan noch vorliegenden Unterschied könnte die Abwesenheit von automatisch intelligent generiertem Feedback genannt werden. Dies ist nicht im Rahmen dieser Arbeit vorgesehen, könnte aber noch Ziel für zukünftige Arbeiten werden.

3.4 System zur Ausgabe von adaptivem Feedback für Programmieraufgaben

Silva et al. präsentieren in ihrer Arbeit [28] ein von ihnen entwickeltes System, das Studierenden beim Lernen von Programmiersprachen helfen soll, in diesem konkreten Fall C/C++. Was den hier präsentierten Ansatz besonders macht, ist, dass das System schon während dem Programmieren in der Lage sein soll Hilfe leisten zu können. Die Hoffnung ist, damit eine verbesserte Lernerfahrung zu erschaffen, im Vergleich zu einer Situation, in der eine Hausaufgabe abgegeben und nur am Ende eine Korrektur erhalten wird. Um dies zu erreichen, stellen Lehrkräfte neben den Aufgaben noch zwei weitere Dinge zur Verfügung. Eine Musterlösung und Hinweise, die entweder in einem Text, Video oder Flussdiagramm Format vorliegen. Dann müssen noch alle Hinweise mit Abschnitten der Musterlösung verknüpft werden. Die Studierenden können dann über eine vorgegebene Schnittstelle programmieren und bei Bedarf Hilfe fordern. Das System versucht dann über einen Abgleich mit der Musterlösung ein der Situation angemessenes Feedback zu geben. Über Knöpfe können die Studierenden angeben, in welchem der drei Formate sie das Feedback erhalten möchten. Sie können zusätzlich einen Bereich im Code angeben, zu dem sie Hinweise möchten und das System versucht etwas möglichst Passendes zu finden. Der Code muss dabei auch nicht syntaktisch korrekt sein, das System versucht in jedem Fall nach einem zum Ansatz passenden Hinweis zu suchen. Falls kein Codeabschnitt angegeben wird, versucht das System generelle Hinweise über den gesamten Code auszugeben.

Hier können drei Unterschiede zu Rockstartit mit den hier geplanten Feedback Ergänzungen genannt werden. Zunächst scheint das System auf spezifische Kurse über Programmieren ausgelegt zu sein. Rockstartit soll interdisziplinäre Onlinekurse anbieten, ohne einen konkreten Bezug zum Programmieren haben zu müssen. Ein weiterer Unterschied ist, dass in Rockstartit die Ermittlung und Ausgabe von einem zu der Situation passenden Feedback nicht geplant ist. Über die Lehreinheit werden unterschiedliche Feedbackmethoden angewandt werden, aber Teilnehmer sollen nicht zu einer Situation Auswahlmöglichkeiten haben, was noch als letzter Unterschied genannt werden kann.

4 Anforderungen an die Implementation

Im Rahmen dieser Arbeit sollen zwei Aspekte untersucht werden. Einerseits soll überprüft werden, ob das Einfügen ausgewählter Feedbackmethoden in Rockstarit Kursen einen Einfluss auf die Bearbeitung hat. Andererseits wird dabei gleichzeitig die Erweiterbarkeit der verwendeten Strukturen und Plugins in WordPress untersucht. Insbesondere die Implementation anknüpfend an die zugrunde liegende Basis des MasterStudy LMS Plugins wird untersucht und die Ergebnisse in Kapitel 5 präsentiert. Zunächst soll im Folgenden die Auswahl, aus pädagogischer Sicht, aber auch ein wenig aus der Sicht der Implementierung, begründet werden und danach die Anforderungen an gewählten Feedbackmethoden gestellt werden.

4.1 Begründung für die Auswahl

Für diese Arbeit wurde beschlossen als Feedbackmethoden, einerseits einen Fortschrittsbalken und andererseits eine Abschlussnachricht bei Vervollständigung einer Lektion zu implementieren. Es folgt nun eine Begründung für die Auswahl der jeweiligen Methoden.

4.1.1 Fortschrittsbalken

Auf Grundlage der in 3.1 präsentierten positiven Rückmeldung zu Fortschrittsbalken im Kontext von Online-Lernkursen wurde beschlossen, auch die Einheiten von Rockstartit, um einen Fortschrittsbalken zu erweitern. Aufgrund der bereits vorhandenen Arbeiten wird erwartet, dass der Fortschrittsbalken zumindest nicht negativ auffällt. Idealerweise sollte der Fortschrittsbalken auch einen positiven Einfluss auf den Lernerfolg haben. Sei es, weil dieser den Überblick über den Kursumfang erleichtert, das Gefühl vermittelt, etwas erreicht zu haben oder Motivation erzeugt, indem gezeigt wird, wie nah man sich am Ziel befindet. Diese Vermutungen werden in Kapitel 6 untersucht.

Außerdem kann ein Fortschrittsbalken auch durchaus in das Schema von gutem Feedback nach Hattie und Timperley, wie in 2.2 vorgestellt, eingearbeitet werden. Ein Fortschrittsbalken lässt sich zumindest dem Ziel von Feedback, dem Reduzieren der Unterschiede zwischen der aktuellen Leistung und dem Ziel, insoweit unterordnen, als es im Kontext von Lernumgebungen eben dies visualisieren soll. Außerdem ist ein Fortschrittsbalken auch in der Lage, die drei wichtigen Fragen zu beantworten:

1. Der Fortschrittsbalken selbst soll eine Darstellung davon sein, welche Ziele es gibt.
2. Der Füllstand des Fortschrittsbalkens visualisiert, wie weit man auf dem Weg zum Ziel ist.

3. Sofern alle wichtigen Aktivitäten dafür verantwortlich sind den Fortschrittsbalken weiter zu füllen, wird somit auch klar, was als Nächstes zu tun ist.

Dadurch, dass die Schülerinnen und Schüler ihren Fortschritt beobachten und evaluieren können, hilft ein Fortschrittsbalken auf der Selbstregulations Ebene. Man könnte auch argumentieren, dass möglicherweise daraus resultierende Erfolgserlebnisse, auch zusätzlich auf der persönlichen Ebene arbeiten. Auf die gleiche Weise könnte man auch begründen, warum ein Fortschrittsbalken die sieben Prinzipien von gutem Feedback nach Nicol und Macfarlane-Dick erfüllt. Ein spannender Aspekt, der nicht besonders gut durch das Modell von Hattie und Timperley abgedeckt ist, ist der siebte Punkt. Nämlich, dass Lehrkräfte ebenso durch Blick auf den Fortschrittsbalken ein Gefühl dafür bekommen können, wie aufwendig der Kurs ist. Ein Vorteil, der auch, wie in 3.1 erwähnt, von Olsson et al. [24, p. 451] genannt wird.

Auch von der Seite der Implementierung ist der Fortschrittsbalken interessant in der Hinsicht, dass dieser Daten über den Kurs braucht und anschließend an die richtige Stelle eingefügt werden soll. Es kann hieran also der Aufwand und mögliche Probleme davon untersucht werden, wie es ist im Kontext des MasterStudy LMS Plugins, an die Daten über den Kurs zu gelangen. Zusätzlich kann man hieran auch untersuchen, wie aufwendig es ist, die Anzeige des Fortschrittsbalkens in die Lektion einzufügen und ob es Probleme gibt, die dabei auftreten.

4.1.2 Feedback bei Vervollständigung

In den Abschnitten 2.1, 2.2 und 3.2 wurde die Nützlichkeit von Feedback, insbesondere in Form eines Kommentars, diskutiert. Deshalb soll auch in den Rockstartit Kursen die Möglichkeit entwickelt werden, Feedback einarbeiten zu können. In Bezug zu dem MasterStudy LMS Plugin, soll also die Option zur Verfügung gestellt werden, eine abschließende Nachricht für jede Lektion einstellen zu können. Da eine Lektion in vielen Fällen ein Thema umfasst, wird dies zumindest als ein guter Zeitpunkt vermutet, bei dem man argumentieren könnte, dass dieser in mancher Definition von Feedback mit Verzögerung fällt. Weiter muss die Nachricht für das Feedback als Einstellung selbst von den Erstellern des Kurses festgelegt werden. Die einzigen Nachrichten, die, unabhängig vom Inhalt, bereits als Standard ausgegeben werden könnten, würden unter Hattie und Timperley in die persönliche Ebene fallen, die nicht effektiv zu sein scheint. Daher ist es notwendig, den Kursentwicklern den Inhalt des abschließenden Kommentars zu überlassen, damit dieser nach Bedarf die anderen Feedbackebenen umfassen kann.

Auch würde dieses Feedback nicht zwangsläufig aufgabenbezogen sein, da dieses einfach nur als Abschluss der Lektion angezeigt werden soll. Eine Konfrontation mit dem eigenen Wissensstand, bzw. dem, was man aus der Lektion mitgenommen hat, ist dennoch in jedem Fall möglich. Je nach Absicht der Lehrkraft könnte eine Zusammenfassung, eine weiterführende Fragestellung oder der Verweis auf vertiefendes Material angezeigt werden. Weiter soll, auch wenn dies schon als EF eingeordnet werden kann, trotzdem die Möglichkeit bestehen auch kleine Tests zum Prüfen des eigenen Kenntnisstandes einzufügen. Dies soll durch die Option H5P Elemente einfügen zu können möglich gemacht werden.

Aus Sicht der Implementierung können hier gleich mehrere Dinge untersucht werden. Erstens, wie einfach es ist, einer Lektion zusätzliche Elemente, sowie Verhalten hinzuzufügen. Zweitens, den Aufwand, neue Einstellungen hinzuzufügen und diese bei Bedarf aufzurufen. Zuletzt, wie gut das Einbinden dritter Plugins funktionieren kann.

4.2 Anforderungen

Im Folgenden werden die Anforderungen an die Feedbackmethoden gelistet. Bei Anforderungen an das System, bei denen es möglich ist, werden diese unter Anforderungen aus einer pädagogischen Sicht untergeordnet.

4.2.1 Fortschrittsbalken

Wie im Abschnitt 3.1 gezeigt, scheinen Form, Größe und Farbe, keinen signifikanten Einfluss auf die Wirkung des Fortschrittsbalkens zu haben. Man kann also die Gestaltung so wählen, dass diese möglichst in das restliche Design des MasterStudy LMS Plugins hinein passt. Zum Füllverhalten im Kontext von Lernumgebungen konnten hier keine passenden Studien gefunden werden, dies soll in dieser Arbeit aber auch nicht weiter untersucht werden. Die Leiste wird sich also an der maximalen Anzahl an Lektionen orientieren. Daraus ergeben sich die folgenden Anforderungen:

- A1 Es soll Schülerinnen und Schülern ermöglicht werden ihren Fortschritt visuell beobachten zu können
 - A1.1 Zu einer Lektion wird am Ende ein Fortschrittsbalken eingefügt
 - A1.2 Der Fortschritt wird über das Verhältnis der aktuellen Lektion gegenüber der maximalen Anzahl an Lektionen bestimmt
 - A1.3 Über dem Fortschrittsbalken wird das in A1.2 beschriebene Verhältnis als Prozentzahl angezeigt
 - A1.4 Der Status der Lektion, also ob diese bereits vervollständigt wurde oder nicht, beeinflusst den Fortschritt nicht
- A2 Schülerinnen und Schüler sollen eine visuell konsistente Erfahrung erhalten
 - A2.1 Der Fortschrittsbalken soll zwischen Chrome (Version 107) und Firefox (Version 106) möglichst gleich aussehen
 - A2.2 Die Enden des Fortschrittsbalkens sollen abgerundet sein
 - A2.3 Die Hintergrundfläche um den Fortschrittsbalken soll, passend zum Rest des Footers, nach der Vervollständigung grün statt weiß angezeigt werden

4.2.2 Feedback bei Vervollständigung

Im Abschnitt 4.1.2 wurde erklärt, warum es als sinnvoll eingestuft wurde, den Kurserstellern selbst zu überlassen, Feedback zu einer Lektion einzufügen. Zusammen mit der Präsentation des neu eingefügten Feedbacks ergeben sich folgende Anforderungen:

- A1 Es soll ermöglicht werden Feedback zu einer Lektion einfügen zu können
 - A1.1 Auf der 'Edit Lesson' Seite wird die Einstellung 'Completion feedback' hinzugefügt
 - A1.2 Über die in A1.1 spezifizierte Einstellung kann ein Text mithilfe eines Texteditors eingefügt werden
- A2 Es soll ermöglicht werden den Text für erhöhte Lesbarkeit zu formatieren
 - A2.1 Im Editor können nach HTML Elementen gerichtete Paragrafen verwendet werden
 - A2.2 Im Editor können die nach HTML Elementen gerichteten Header 1-6 verwendet werden
 - A2.3 Im Editor kann fett gedruckter Text eingefügt werden
 - A2.4 Im Editor können Aufzählungspunkte eingefügt werden
- A3 Am Ende einer Lektion soll die Möglichkeit bestehen, dass Schülerinnen und Schüler mit einer Aufgabe ihr Wissen überprüfen können
 - A3.1 Auf der 'Edit Lesson' Seite wird die Einstellung 'H5P shortcode' hinzugefügt
 - A3.2 Über die in A3.1 spezifizierte Einstellung kann ein shortcode Verweis auf ein H5P Element eingegeben werden, das im Feedback zu der Lektion erscheinen soll
- A4 Schülerinnen und Schüler sollen klar erkennen können, dass ihnen gerade Feedback gegeben wird
 - A4.1 Zu einer Lektion, zu der Feedback eingestellt wurde, wird ein Overlay Element hinzugefügt
 - A4.2 Wenn kein Feedback zu einer Lektion eingestellt wurde, wird kein zusätzliches Verhalten hinzugefügt
 - A4.3 Wenn die Lektion geladen wird ist das Overlay Element zunächst nicht sichtbar
 - A4.4 Die Navigationsleiste am Ende der Lektion wird um die Funktionalität ergänzt, dass durch Anklicken des 'Complete' Elements, dass in A4.1 spezifizierte Overlay erscheint
 - A4.5 Der Status der Lektion, also ob diese bereits vervollständigt wurde oder nicht, ist nicht relevant für die in A4.4 beschriebene Anforderung
 - A4.6 Der Text welcher in der in A1 spezifizierten Einstellung eingegeben wurde, wird auf dem Overlay angezeigt
 - A4.7 Beim Überschreiten der maximal auf einmal anzeigbaren Menge an Text kann mithilfe einer Scrollbar dieser angezeigt werden
 - A4.8 Das Overlay hat einen Knopf um dieses wieder zu schließen
 - A4.9 Das Overlay hat einen Knopf um zur nächsten Lektion zu gelangen

5 Implementation

In diesem Kapitel soll spezifisch die Implementation des im Rahmen der Arbeit entwickelten Plugins untersucht werden. Genauer sollen zuerst die Gesamtstruktur und im Anschluss ein paar Besonderheiten, die während der Entwicklung aufgetaucht sind, betrachtet werden. Im Anschluss werden noch ein paar Testszenarien beschrieben. Abschließend werden noch einige Limitationen hinsichtlich der Implementierung vorgestellt.

5.1 Design

Im Abschnitt 2.6 wurde bereits beschrieben, dass es nicht empfehlenswert ist den WordPress Core oder ein Plugin, welches nicht von einem selbst entwickelt wurde, direkt zu modifizieren. Der Grund ist, dass bei jedem Update die eigenen Änderungen verworfen werden. Stattdessen, sollte ein weiteres Plugin verwendet werden, um zusätzliche Funktionalität hinzuzufügen. Durch die Verwendung von geeigneten *Hooks*, kann man an der richtigen Stelle, die gewünschten Modifikationen vornehmen. Zusätzlich stehen in einem neuen Plugin, Klassen und Methoden eines anderen Plugins zu Verfügung, da für WordPress diese global verfügbar sind. Entsprechend ist es bei dem Entwurf wichtig, für alles einen eindeutigen Bezeichner zu haben. Anhand der folgenden Abbildung 5.1 sollen die Stellen, an denen die zwei Plugins miteinander interagieren, dargestellt werden. Zur Übersicht wurden Methoden, die hier keine besonders große Relevanz darstellen, ausgelassen. Außerdem sind manche Dateien nur Skript-Dateien, die keine Klassen definieren. Dies wurde hier durch Angeben der Sprache über dem Dateinamen gekennzeichnet. Die benötigten Javascript und CSS Dateien, werden über das Plugin für WordPress eingebunden, wo diese ebenfalls global zur Verfügung stehen. Alle Pfeile, die mit „calls“ gekennzeichnet sind, sollen das Auslösen von einem *Hook* darstellen. Da durch die *Hooks* auch ein zeitlicher Verlauf entsteht, wurden zur weiteren Verdeutlichung Sequenzdiagramme erstellt. Abbildung 5.2 beschreibt den Start der Interaktion zwischen den zwei Plugins: Im Plugin „MasterStudy LMS“ wird, über das Skript *stm-lms-lesson* der *Hook* *stm_lms_lesson_started* beim Laden einer Lektion aufgerufen. Durch den *Hook* werden ebenfalls die ids des Kurses und der Lektion, so wie in Abschnitt 2.6.5 als *post_id* und *item_id*, übergeben. Im Plugin „a-rockstartit-masterstudy-integration“ wird im Skript *rockstartit_masterstudy_integration_fetch_post_data* eine Methode über diesen *Hook* aufgerufen. Diese erstellt ein Objekt der Klasse *Rockstartit_masterstudy_integration_helper* dem sie sowohl *post_id* als auch *item_id* übergibt. Dieses Objekt wird benutzt, um Informationen über die aktuelle Lektion zu erhalten und entsprechend über einen weiteren *Hook* an Methoden weitergegeben, die es gleich brauchen. Der im Rahmen dieses Plugin neu definierte *Hook* *rockstartit_masterstudy_post_data_recieved*, ruft Methoden auf, die dann die in Abschnitt 4 vorgestellten Arten von Feedback umsetzen. Damit ist ein zeitlicher Verlauf sichergestellt

5 Implementation

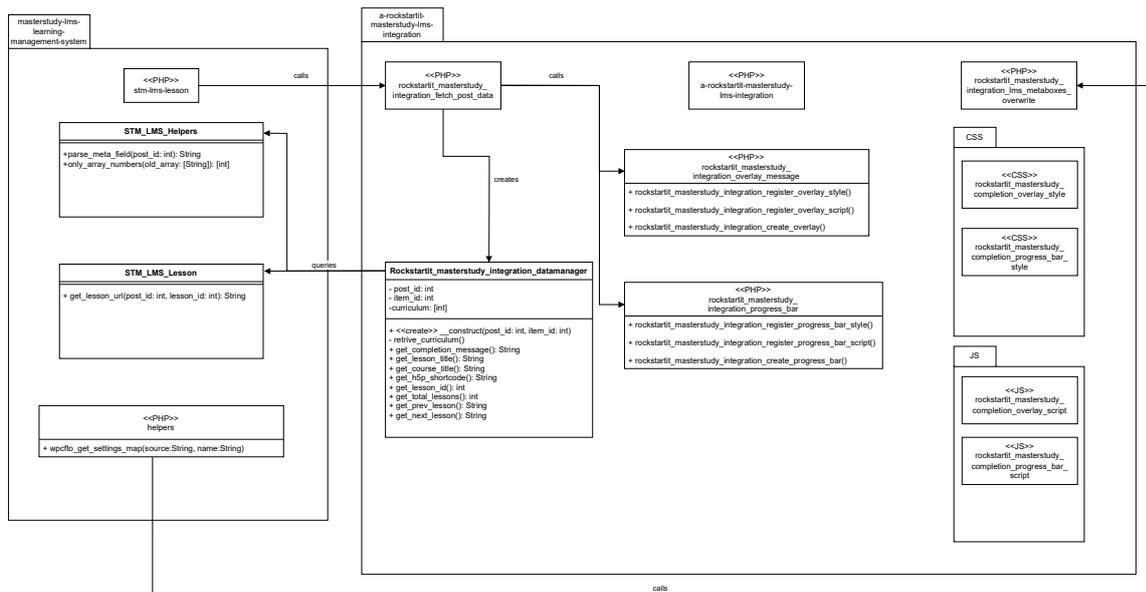


Abbildung 5.1: Diagramm, welches das Plugin „a-rockstartit-masterstudy-integration“, sowie dessen Interaktion mit „MasterStudy LMS“ darstellen soll.

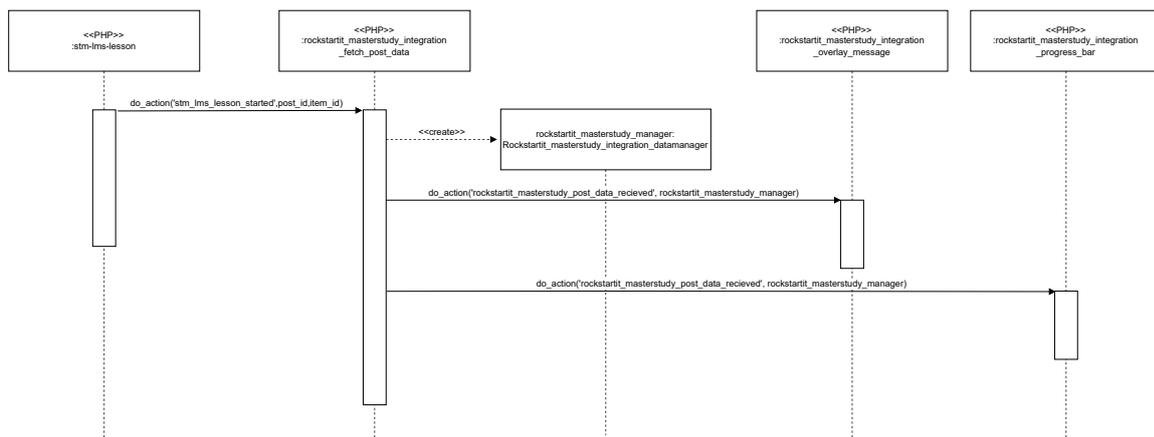


Abbildung 5.2: Zeitlicher Verlauf beim Auslösen der für das Plugin relevanten Hooks.

und über den *Hook* könnten in Zukunft zusätzliche Erweiterungen erstellt werden. Abbildung 5.3 zeigt den Ablauf für das Erstellen des Overlays. Wie bereits in Abbildung 5.2 beschrieben, erstellt das Skript `rockstartit_masterstudy_integration_fetch_post_data` ein Objekt der Klasse `Rockstartit_masterstudy_integration_helper` und übergibt dieses beim Aufruf des Hooks `rockstartit_masterstudy_post_data_recieved`. Beim Aufruf des Hooks wird im Skript `rockstartit_masterstudy_integration_overlay_message` das Overlay erstellt. Dafür fragt dieses auch das im Hook übergebene Objekt nach zu dieser Lektion gespeicherten Daten ab, die für das Overlay gebraucht werden. Zum Schluss werden noch die Knöpfe über den Aufruf einer Javascript Methode erstellt und an das Overlay gehängt. Im Weiteren gibt es noch das Erstellen des Fortschrittsbalkens, dargestellt in Abbildung 5.4. Die zugrunde liegende Vorgehensweise ist die gleiche wie mit dem Overlay. Über den

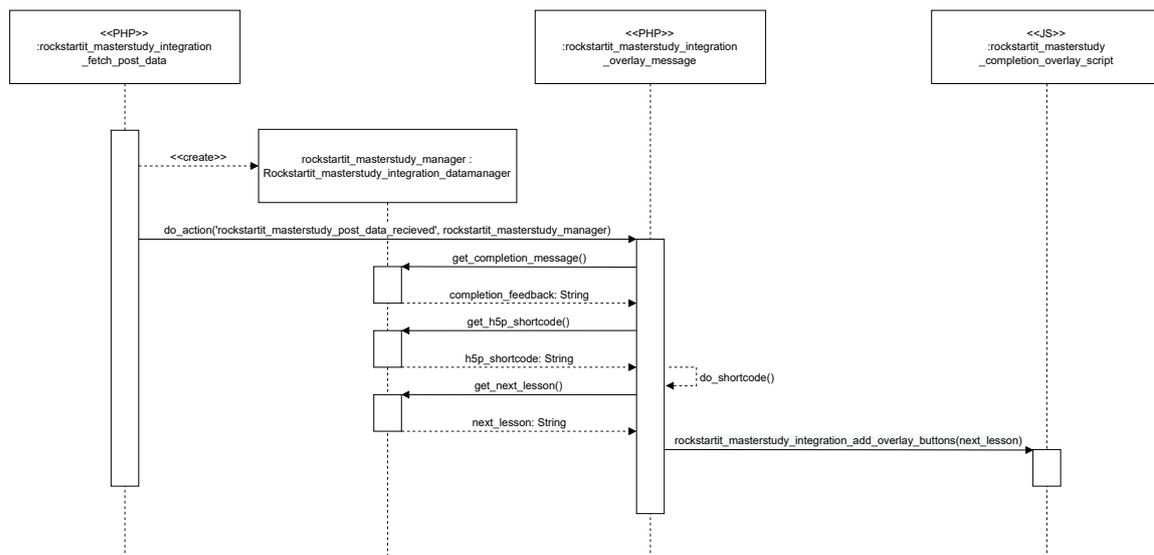


Abbildung 5.3: Ablauf beim Generieren des Overlays mit dem Feedback Kommentar.

Aufruf des *Hooks* fängt das Skript *rockstartit_masterstudy_integration_progress_bar* an, den Fortschrittsbalken zu erstellen. Dabei wird das über den *Hook* übergebene Objekt benutzt, um benötigte Informationen über die Lektion zu erhalten. Da es zeitlich hier so nicht ganz eingeordnet werden kann, sei noch erwähnt, dass der Fortschrittsbalken nach dem Laden der Seite über jQuery an das Ende nach ganz unten bewegt wird.

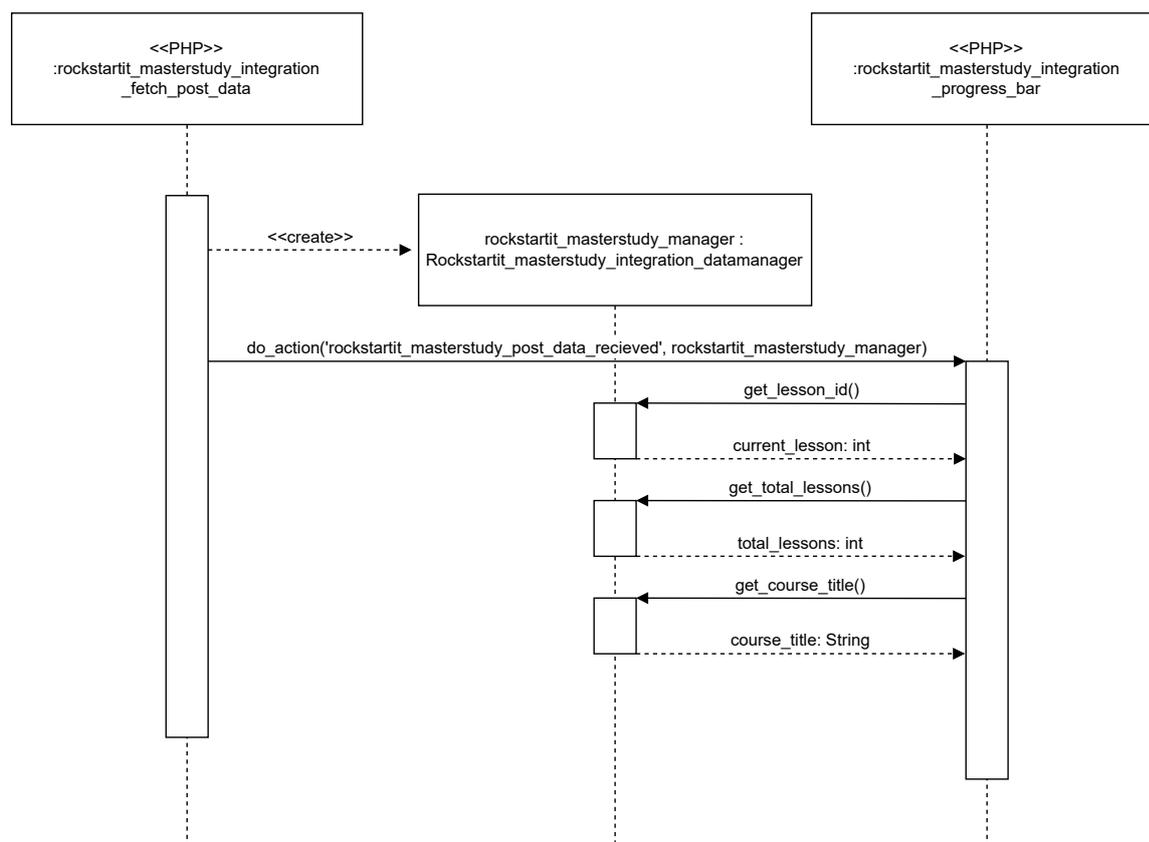


Abbildung 5.4: Ablauf beim Generieren des Fortschrittsbalkens.

5.2 Besonderheiten

Im Verlauf der Entwicklung sind einige Probleme aufgetaucht, dessen Lösung es wert sein könnte, nochmal genauer betrachtet zu werden. In diesem Abschnitt werden diese Probleme sowie der gewählte Lösungsansatz vorgestellt.

5.2.1 Einbindung von CSS und JS Dateien

Typischerweise werden in WordPress CSS und JS Dateien eingebunden, in dem auf diese durch bestimmte Methodenaufrufe verwiesen wird. Danach stehen sie der WordPress Seite global zur Verfügung. Über den Hook *wp_enqueue_scripts* können dann die Methoden aufgerufen werden, die für die Einbindung verantwortlich sind. Der folgende Codeausschnitt in Listing 5.1 aus dem Plugin soll dies veranschaulichen.

```

1 // Register style sheet.
2 add_action( 'wp_enqueue_scripts', '
    rockstartit_masterstudy_integration_register_overlay_style' );
3 /**
4  * Registers the stylesheet needed for the overlay.
5  */
6 function rockstartit_masterstudy_integration_register_overlay_style() {
7     wp_register_style( 'rockstartit_masterstudy_completion_overlay_css', plugins_url( '
    assets/css/rockstartit_masterstudy_completion_overlay_style.css', __FILE__ ) );
8     wp_enqueue_style( 'rockstartit_masterstudy_completion_overlay_css' );
9 }
10
11 // Register javascript file.
12 add_action( 'wp_enqueue_scripts', '
    rockstartit_masterstudy_integration_register_overlay_script' );
13 /**
14  * Registers the script needed for the overlay.
15  */
16 function rockstartit_masterstudy_integration_register_overlay_script() {
17     wp_register_script( 'rockstartit_masterstudy_completion_overlay_js', plugins_url( '
    assets/js/rockstartit_masterstudy_completion_overlay_script.js', __FILE__ ), array( '
    jquery' ) );
18     wp_enqueue_script( 'rockstartit_masterstudy_completion_overlay_js' );
19 }

```

Listing 5.1: Registrieren von JS und CSS Dateien

In Hinblick auf die benötigten Parameter sind sich die Methoden zum Einbinden der CSS und JS Dateien ähnlich. Es gibt eine Methode, um die Datei mit einem eindeutigen *Handler* zu registrieren und dann eine weitere, um diese dann über den definierten *Handler* einzubinden. Zusätzlich können beim Registrieren auch Abhängigkeiten angegeben werden, was man hier bei der JS Datei mit jQuery sehen kann.

Bei der Entwicklung sind hier zwei Probleme aufgetaucht, die beide mit der Angabe des Pfades bei der Registrierung zu tun haben. Erstens muss man aufpassen, dass alle Dateien auch wirklich eindeutige Namen haben, eine unterschiedliche Dateiendung reicht nicht. Zweitens ist bei der Methode zur Pfadangabe *plugins_url* der zweite Parameter, der den Pfad zum Plugin angibt, zwar optional, aber auf diesen zu verzichten ist nicht zu empfehlen. Beides kann nämlich inkonsistentes Verhalten, über Systeme hinweg auslösen und dafür sorgen, dass die benötigten CSS und JS Dateien nicht geladen werden. Nach den entsprechenden Änderungen scheinen diese Probleme nicht mehr aufzutreten.

5.2.2 Anpassung des *Complete* Knopfes

Im „MasterStudy LMS“ Plugin wurde die Javascript Logik unter anderem mit jQuery verfasst. So wurde auch bei dem *Complete* Knopf am Ende der Lektion, Interaktion über jQuery festgelegt. Unter anderem werden beim Klicken auch die CSS Klassen angepasst, um andere Farben anzeigen zu können. Allerdings hat dies auch dafür gesorgt, dass übliche Methoden, welche alle Elemente einer Klasse holen, nicht mehr funktionieren. Damit diese funktionieren, mussten im neu entwickelten Plugin die Anpassungen ebenfalls über jQuery erfolgen. Um eine logisch klare Trennung zu erzeugen, wurde kein zusätzliches

Verhalten am vorhandenen Knopf hinzugefügt. Stattdessen wird ein neues div-Element um den Knopf erzeugt. Wenn dieses geklickt wird, ist es dafür verantwortlich das Overlay sichtbar zu machen, sofern eines erzeugt wurde. Außerdem ist noch nennenswert, dass ein einfaches div-Element um den *Complete* Knopf zu groß erzeugt werden würde. Dies liegt daran, dass der *Complete* Knopf Teil eines *Flexbox* Layouts ist. Für die richtige Größe muss das umgebende div-Element ebenfalls als *Flexbox* eingestellt werden. Der notwendige JS und CSS Code für den gewählten Ansatz, kann in Listing 5.2 und Listing 5.3 gesehen werden.

```
1 //Add onclick logic to existing complete button. Scince it has been modified in jQuery by
   masterstudy, it appears that it can only be modified by jQuery
2 jQuery(document).ready(function( $ ) {
3
4   /*If an additional flexbox is not added, the area that can be clicked to make the overlay
   appear will be bigger than the completion button.
5   * This makes them mostly the same size
6   */
7   $(".stm-lms-lesson_navigation_complete").wrap("<div id='
   rockstartit_masterstudy_integration_complete_container'></div>");
8
9   $(document).on('click', '#rockstartit_masterstudy_integration_complete_container',
   function(){
10    if ($("#rockstartit_masterstudy_integration_overlay").length) { //Checks if the overlay
   exists to catch errors.
11    rockstartit_masterstudy_integration_overlay_on(); //make Overlay visible
12    }
13  });
14 });
```

Listing 5.2: JS Code für den *Complete* Knopf

```
1 /* Used to make the area that can be clicked to make the
2    overlay appear, to be around the same size as the
3    complete button */
4 #rockstartit_masterstudy_integration_complete_container{
5   display: flex;
6 }
```

Listing 5.3: CSS Code für den *Complete* Knopf

5.2.3 Gleiches Anzeigen des Fortschrittbalkens zwischen Chrome und Firefox

Ein weiteres Problem, das im Verlauf der Entwicklung auftauchte, war, dass die Fortschrittsbalken in Chrome und Firefox unterschiedlich angezeigt wurden. Dies liegt daran, dass die zugrunde liegenden Engines unterschiedliche Voreinstellungen haben. Diese können aber über CSS explizit zurückgesetzt werden. Der dafür relevante Ausschnitt aus dem Plugin ist in Listing 5.4 zu sehen.

```

1 /*Style for the progressbar element.
2 Resets default browser styles to achive
3 a unanimous look*/
4 #rockstartit_masterstudy_integration_lesson_pogress{
5     -webkit-appearance: none;
6     -moz-appearance: none;
7     appearance: none;
8 }

```

Listing 5.4: Zurücksetzen von CSS Voreinstellungen

Das Attribut *-webkit-appearance* spricht dabei explizit Chrome (sowie auf der Chromium Engine laufende Browser, wie beispielsweise Edge) und *-moz-appearance* Firefox an. Das Attribute *appearance* spricht hingegen alle Browser an und tritt sozusagen extra nochmal in Kraft, falls die ersten zwei Attribute es nicht tun sollten. Um nun noch ein gleiches Aussehen zwischen den Browsern zu erzeugen, muss dieses explizit eingestellt werden. Für Chrome gibt es zwei unterschiedliche Einstellungen. Einmal eine, die das Aussehen für den äußeren Teil des Fortschrittsbalkens bestimmt und eine, die das Aussehen für den Inhalt bestimmt, wie in Listing 5.5 gesehen werden kann.

```

1 /*Needed for chromium based browsers to make the border of the
2 progress bar appear correctly*/
3 #rockstartit_masterstudy_integration_lesson_pogress::-webkit-progress-bar{
4     border-radius: 30px;
5     background-color: white;
6 }
7 /*Needed for chromium based browsers to make the content of the
8 progress bar appear correctly*/
9 #rockstartit_masterstudy_integration_lesson_pogress::-webkit-progress-value{
10     border-radius: 30px;
11     background-color: #007cba;
12 }

```

Listing 5.5: CSS Einstellungen für Browser in der Chromium Engine

Bei Firefox gibt es hingegen nur eine Einstellung für den Inhalt des Fortschrittsbalkens. Die Einstellungen für den äußeren Teil müssen noch in die allgemeine Version der CSS Regel aufgeschrieben werden. Listing 5.6 zeigt die spezifischen Einstellungen für Browser mit der Firefox Engine.

```

1 /*Needed for firefox based browsers to make the content of the
2 progress bar appear correctly*/
3 #rockstartit_masterstudy_integration_lesson_pogress::-moz-progress-bar{
4     border-radius: 30px;
5     background-color: #007cba;
6 }

```

Listing 5.6: CSS Einstellungen für Browser in der Firefox Engine

Durch Einfügen der gleichen Einstellungen in den Versionen für beide Browser kann sichergestellt werden, dass diese gleich angezeigt werden.

5.2.4 Korrekte Anzeige von H5P Elementen

Ein den Entwicklern von H5P bekanntes Problem ist, dass ein H5P Element keine Größe zugewiesen bekommt, wenn es innerhalb eines HTML Elements platziert wird, das zunächst unsichtbar ist. Genau dies ist aber mit dem Feedback-Overlay der Fall. Wenn das Overlay dann also sichtbar gemacht wird, scheint es so, als wäre das H5P Element nicht vorhanden. In Wahrheit hat dieses aber lediglich keine Größe zugewiesen bekommen. Sobald die Fenstergröße des Browsers geändert wird, werden aber die Berechnungen neu gemacht und das H5P Element bekommt eine Größe zugewiesen. Dadurch wird es sichtbar. Um nicht händisch die Fenstergröße ändern zu müssen, kann man aber auch per Javascript das entsprechende Ereignis aufrufen, das sich um die Neuberechnung kümmert. Dies ist ein bekannter Workaround, der dennoch wichtig zu wissen ist, wenn man mit H5P arbeitet. Die hier angewendete Lösung ruft bei Handlungen, die sowieso im Hintergrund geschehen werden, wie klicken und scrollen, das *Resize* Ereignis auf, das dem H5P Element eine Größe gibt. Die Umsetzung wird in Listing 5.7 gezeigt.

```
1 /*Due to an issue with caused by inserting h5p into hidden elements, the h5p element loads
   with a height of 0.
2 The follwing code calls the resize event on multiple locations to make the h5p element
   appear. */
3 //call resize after 0,5sec
4 setTimeout((event) => {
5   window.dispatchEvent(new Event('resize'));
6 }, 500);
7
8 //call resize event when page is loaded so that accordion h5p elemnts are displayed
   correctly
9 window.addEventListener("load", (event) => {
10  window.dispatchEvent(new Event('resize'));
11 });
12
13 //call resize event when page is scrolled so that accordion h5p elemnts are displayed
   correctly
14 window.addEventListener('scroll', (event) => {
15  window.dispatchEvent(new Event('resize'));
16  console.log("scrolled.. resized");
17 });
18
19 //call the resize event, when an item is clected, to get elements in accordions displayed
   correctly
20 window.addEventListener('click', (event) => {
21  window.dispatchEvent(new Event('resize'));
22  console.log('Clicked... resize event');
23 });
```

Listing 5.7: Workaround für die korrekte Anzeige von H5P Elementen

5.3 Testszzenarien

Aufgrund der einerseits auf das „MasterStudy LMS“ Plugin aufbauenden und andererseits interaktiven Natur des Plugins, gibt es keine richtige Möglichkeit sinnvolle Unit-Tests zu verfassen. Stattdessen werden im Folgenden Testszzenarien beschrieben, die zusammen alle Anforderungen in Anspruch nehmen sollen.

5.3.1 Szenario 1

Eine Lehrkraft möchte einige Lektionen ihres Kurses um Feedback ergänzen. Diese wählt also zunächst eine Lektion und geht in die dazugehörigen Einstellungen. Die Lehrkraft gibt ihr Feedback in den Texteditor, der zur Einstellung „Completion feedback“ gehört. Für eine erhöhte Lesbarkeit schreibt die Lehrkraft zunächst einen Titeltext mit der Schriftgröße H1 und wählt für den Rest die normale Schriftgröße. Um ein paar Begriffe klar darzustellen, bekommen diese in einer Aufzählung alle einen eigenen Stichpunkt. Um klarzustellen, dass einer der Begriffe besonders wichtig ist, wird dieser fett markiert. Die Lehrkraft speichert die Änderungen an die Lektion und bearbeitet im Anschluss eine weitere Lektion. Da sie den Schülern am Ende eine Aufgabe stellen möchte, gibt sie in das Feld „H5P shortcode“ den shortcode zu einem vorher erstellten H5P Quiz ein. Die Lehrkraft schreibt zusätzlich noch einen Einleitungstext in den Editor für die Einstellung „Completion Feedback“. Die Lehrkraft speichert die Änderungen an die Lektion. Es gibt noch eine dritte Lektion, aber die Lehrkraft ist der Meinung hierfür kein Feedback hinzufügen zu müssen und lässt beide Einstellungen leer.

5.3.2 Szenario 2

Ein Schüler soll einen Kurs testen, in dem zu manchen Lektionen Feedback hinzugefügt wurde. Nachdem der Schüler die erste Lektion gelesen hat, klickt dieser auf „Complete“, woraufhin sich ein Overlay über die Lektion sichtbar macht, das vorher nicht zu sehen war. Auf dem Overlay ist eine Box, auf der ein zu dieser Lektion spezifisches Feedback steht. Da das Feedback etwas länger ist, scrollt der Schüler an der Box im Overlay, um dieses vollständig zu lesen. Mit dem „Weiter“ Knopf im Overlay, klickt sich der Schüler zu der nächsten Lektion im Kurs. Der Schüler überfliegt die Lektion und klickt auf „Complete“. Erneut wird ein Overlay sichtbar, in dem nach einem kleinen Einleitungstext ein H5P Element mit einem Multiple-Choice Quiz steht. Der Schüler ist sich der Antwort nicht sicher und drückt den Knopf mit dem „X“, um das Overlay wieder unsichtbar zu machen. Nachdem dieser sich die Lektion gründlich durchgelesen hat, klickt er erneut auf „Complete“ und das Overlay wird erneut sichtbar. Er beantwortet die Fragen und klickt dann auf „Weiter“, um zur nächsten Lektion zu gelangen. Nachdem der Schüler auch diese Lektion gelesen und auf „Complete“ gedrückt hat, erscheint aber kein Overlay, da kein Feedback für diese Lektion eingestellt wurde. Stattdessen bemerkt der Schüler den Fortschrittsbalken mit den abgerundeten Ecken am Ende der Lektion. Dieser zeigt visuell, aber auch als Prozentzahl an, die wievielte Lektion dies in Verhältnis zu der Gesamtzahl der Lektionen ist. Nachdem der Schüler auf „Complete“ gedrückt hat, hat sich der Hintergrund des gesamten Footers,

also auch bei dem Fortschrittsbalken grün gefärbt. Der Inhalt des Fortschrittsbalkens blieb aber der gleiche wie zuvor.

5.3.3 Szenario 3

Eine Schülerin möchte ihrer Freundin den Onlinekurs zeigen, den sie in der Schule gemacht hat. Dabei fällt ihr auf, dass der Rechner der Freundin Firefox als Browser hat. Die Rechner in der Schule verwendeten hingegen Chrome. Allerdings wird der Kurs, und insbesondere der Fortschrittsbalken, genauso wie in der Schule angezeigt.

5.4 Limitationen

Im Laufe der Entwicklung sind einige Probleme aufgetaucht, zu denen keine Lösung gefunden werden konnte. In diesem Abschnitt werden speziell Limitationen betrachtet, die im Zusammenhang mit der Implementierung stehen. Für die meisten kann mit einer ziemlichen Sicherheit gesagt werden, dass diese darauf zurückgeführt werden können, dass die Entwickler von „MasterStudy LMS“ nicht vorgesehen haben, die Entwicklung dritter Plugins zu unterstützen. Das Plugin ist schließlich ein kommerzielles Produkt mit Pro-Versionen, die, für jährliche Gebühren, mehr Features enthalten. Ein weiterer Faktor ist, dass „MasterStudy LMS“ dafür vorgesehen ist, Webseiten zu erstellen, die bezahlte Kurse anbieten. Daher ist auch vorgesehen, dass jeder Nutzer einen eigenen Account hat. Auch wenn man Kurse kostenlos anbieten kann, wird jeglicher Fortschritt an Accounts gebunden und Kurse sind auch nur zugänglich, sofern man angemeldet ist. Dies ist nicht im Interesse des Rockstartit Projekts. Es wäre zu umständlich für einige Unterrichtseinheiten Schülerinnen und Schülern einen Account erstellen zu lassen oder diesen einen zur Verfügung stellen zu müssen. Es gibt auch die Möglichkeit festzulegen, dass eine Lektion eine Vorschau sein soll. Damit kann diese zwar betrachtet werden, ohne dass man angemeldet sein muss, jedoch wird dann keine Navigationsleiste mehr angezeigt, um zu weiteren Lektionen zu gelangen. Die Vermutung liegt also nahe, dass „MasterStudy LMS“ nicht darauf ausgelegt wurde, auf diese Art und Weise Kurse anzubieten.

Ein weiteres Problem ist, dass es auch keine komplett passenden *Hooks* gibt, um an der richtigen Stelle das Verhalten auf die gewünschte Art und Weise anzupassen. Der für das „a-rockstartit-masterstudy-integration“ Plugin verwendete *Hook stm_lms_lesson_started* aus „MasterStudy LMS“, wird nur ausgelöst, wenn die Nutzer angemeldet sind. Bei einer Vorschau der Lektion wird die neue Funktionalität nicht hinzugefügt. Selbst wenn ein passender *Hook* dafür gefunden werden könnte, basieren alle weiteren Funktionen darauf, den Footer zu verändern, der in der Vorschau sowieso nicht angezeigt wird. Um den Footer doch in der Vorschau erscheinen zu lassen, wurden ebenfalls keine passenden *Hooks* gefunden. Selbst mit direkten Modifikationen am „MasterStudy LMS“ Plugin, die wie in Abschnitt 2.6 diskutiert nicht empfehlenswert sind, konnte nicht das gewünschte Verhalten erzielt werden. Dies lag daran, dass jegliche Modifikationen in dieser Hinsicht, den restlichen Programmablauf gestört haben. Es konnten also nicht nur keine passenden *Hooks* gefunden werden, sondern direkte Modifikationen in der Hinsicht gehen gegen das vorhergesehene Design von „MasterStudy LMS“.

Abschließend gibt es noch das Problem, neue Einstellungen hinzuzufügen. Das Einfügen von Einstellungsfeldern wurde in „MasterStudy LMS“ dadurch gelöst, dass eine anonyme Funktion durch einen Filter aufgerufen wird. Diese erstellt dann ein Array mit allen Einstellungen und gibt dieses für die Weiterverarbeitung zurück. Es konnte keine Möglichkeit gefunden werden, die Einstellungen im Anschluss noch zu bearbeiten. Dies ist aber eine Stelle, bei der am ehesten noch vermutet wird, dass eine bessere Lösung gefunden werden kann. Hier besteht aber auch die Möglichkeit eines Workarounds. Durch Kopieren der Funktion, die durch den Filter aufgerufen wird, kann man eine modifizierte Version, mit den neuen Einstellungen aufrufen. Diese erscheinen dann in den Einstellungen. Allerdings scheint dies auch nur zu funktionieren, insofern der Name der Plugins alphabetisch vorher erscheint. Vermutlich hängt dies damit zusammen, in welcher Reihenfolge die Plugins geladen werden. Da es sich hier aber zweifelsohne um eine Schwachstelle im Programm handelt, sollte bei weiteren Arbeiten am Plugin diese Stelle überarbeitet werden.

5.5 Ergebnisse

Im Folgenden werden die Ergebnisse aus der Frontend Seite vorgestellt. Dafür werden zu allen bereits beschrieben und entwickelten Aspekten nochmals Abbildungen gezeigt, um sich einen besseren Eindruck vom Endergebnis machen zu können.

5.5.1 Fortschrittsbalken

Die Abbildungen 5.5 und 5.6 zeigen jeweils, wie die Navigationsleisten einer Lektion aussehen, wenn die Lektion vervollständigt oder unvollständig ist. In jedem Fall passt sich der Fortschrittsbalken dem Stil der Navigationsleiste an.



Abbildung 5.5: Die Navigationsleiste einer unvollständigen Lektion mit Fortschrittsbalken.



Abbildung 5.6: Die Navigationsleiste einer vervollständigten Lektion mit Fortschrittsbalken.

5.5.2 Overlay

Der zweite implementierte Mechanismus ist das Overlay, welches durch die Abbildungen 5.7 und 5.8 dargestellt ist.

5 Implementation

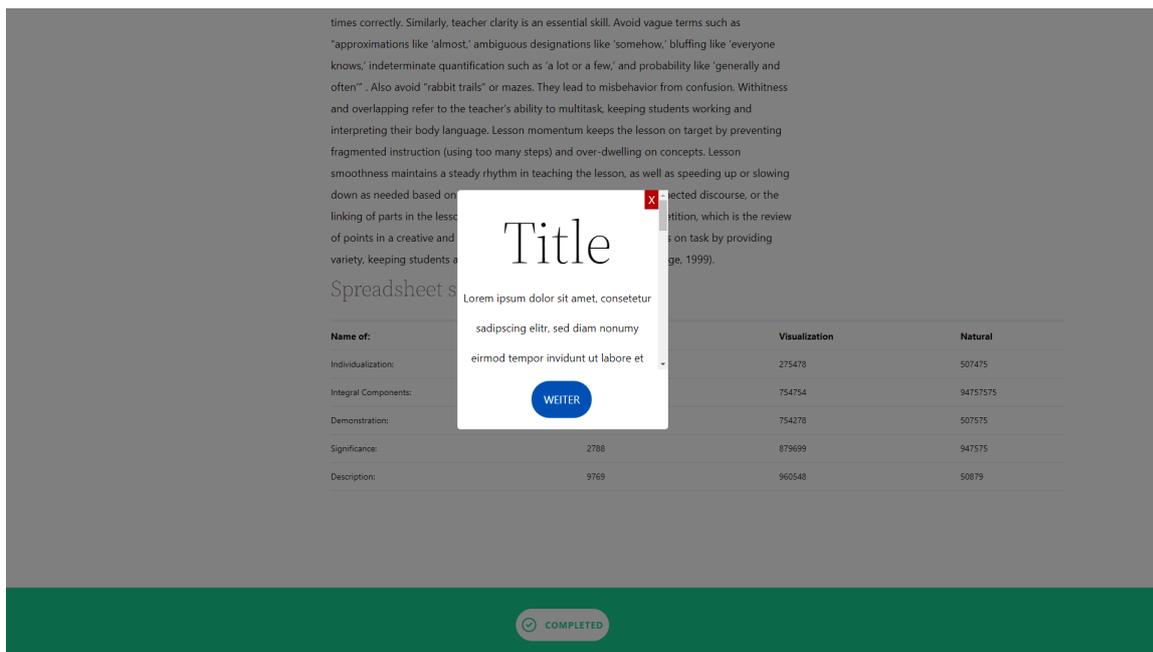


Abbildung 5.7: Eine Lektion mit sichtbarem Overlay, in dem Text zu sehen ist.

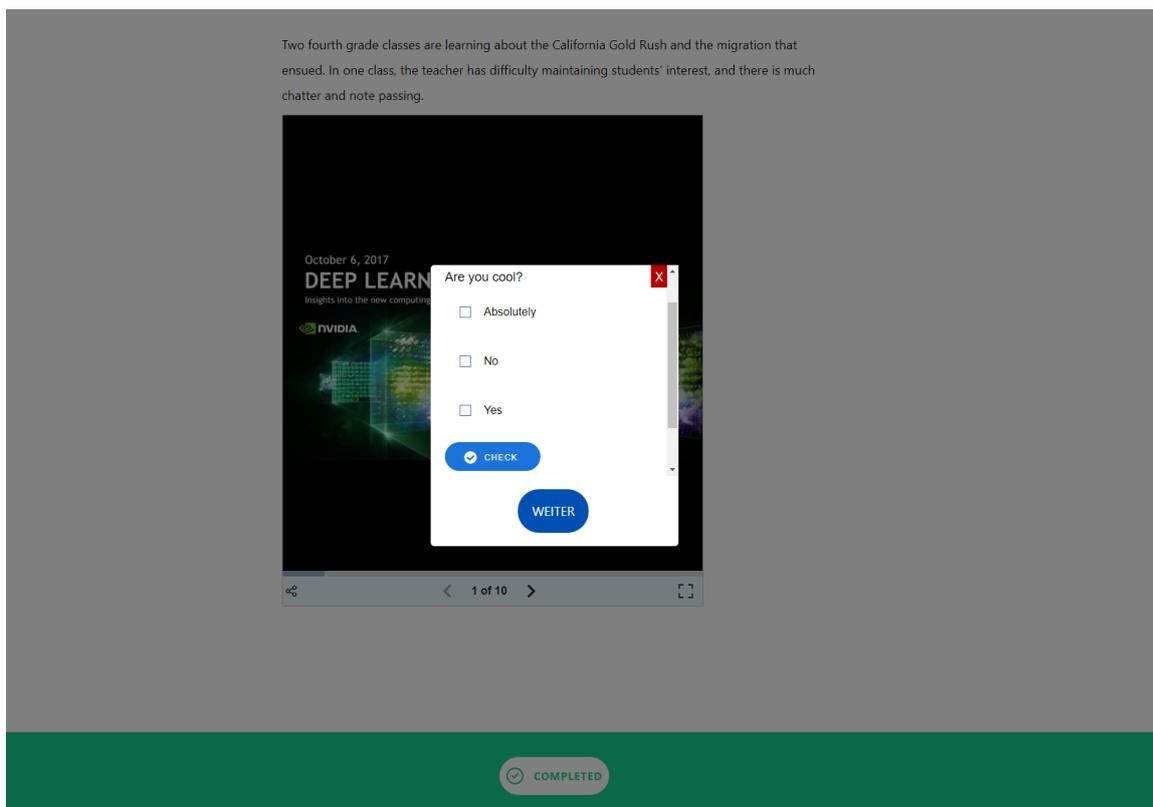


Abbildung 5.8: Eine Lektion mit sichtbarem Overlay, in dem ein H5P Element zu sehen ist.

Neben dem Overlay selbst gibt es noch die hinzugefügten Einstellungen, die in Abbildung 5.9 zu sehen sind.

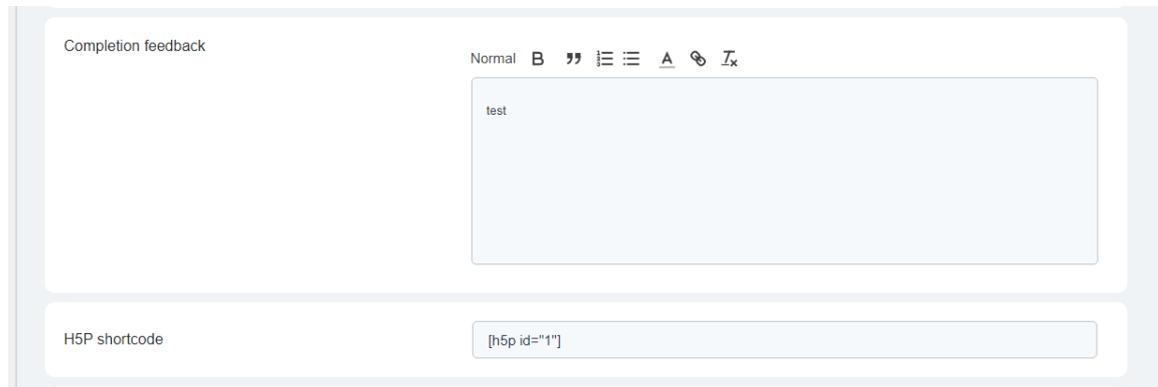


Abbildung 5.9: Die neu hinzugefügten Einstellungen einer Lektion.

5.6 Modularer Ansatz

Gegen Ende der Entwicklung kam noch der Vorschlag auf, das Projekt so aufzuteilen, dass jede neu hinzugefügte Funktionalität für Feedback, ihr eigenes Plugin sein sollte. Dies ermöglicht einen noch modulareren Einsatz, sowie modularere Entwicklung. Falls also der Wunsch besteht, nur den Fortschrittsbalken oder das Overlay zu verwenden, ist dies dadurch möglich. Bei Bedarf kann also nur eines der beiden Plugins verwendet werden. Es können aber auch beide gleichzeitig aktiviert werden, um die volle Funktionalität zu erhalten.

Die Umsetzung war zu großen Teilen nur ein Aufteilen des bereits erstellten Plugins. Die neue Struktur wird in Abbildung 5.10 dargestellt. Es gab nur eine wirklich wichtige Sache zu beachten und das ist die globale Struktur von WordPress. Damit die beiden Plugins gleichzeitig funktionieren konnten, durfte keine Datei und keine Methode zwischen den beiden Plugins den gleichen Namen haben. Andernfalls würde die Aktivierung beider Plugins gleichzeitig nicht funktionieren. Deshalb bestehen auch nahezu alle Änderungen darin, die Namen aller Dateien und Funktionen einzigartig zu machen. Auch die Namen der neu definierten *Hooks* wurden einzigartig gemacht, um mögliche Seiteneffekte zu verhindern. Die einzige weitere Änderung besteht darin, dass bei der Anzeige des Fortschrittsbalkens geprüft wird, ob das Overlay-Plugin aktiviert ist. Sollte nicht damit umgegangen werden, kommt es nämlich zu Fehlern bei der Anzeige, da beide die Navigationsleiste manipulieren. Die Stelle im Code, die dies überprüft, ist in Listing 5.8 zu sehen.

5 Implementation

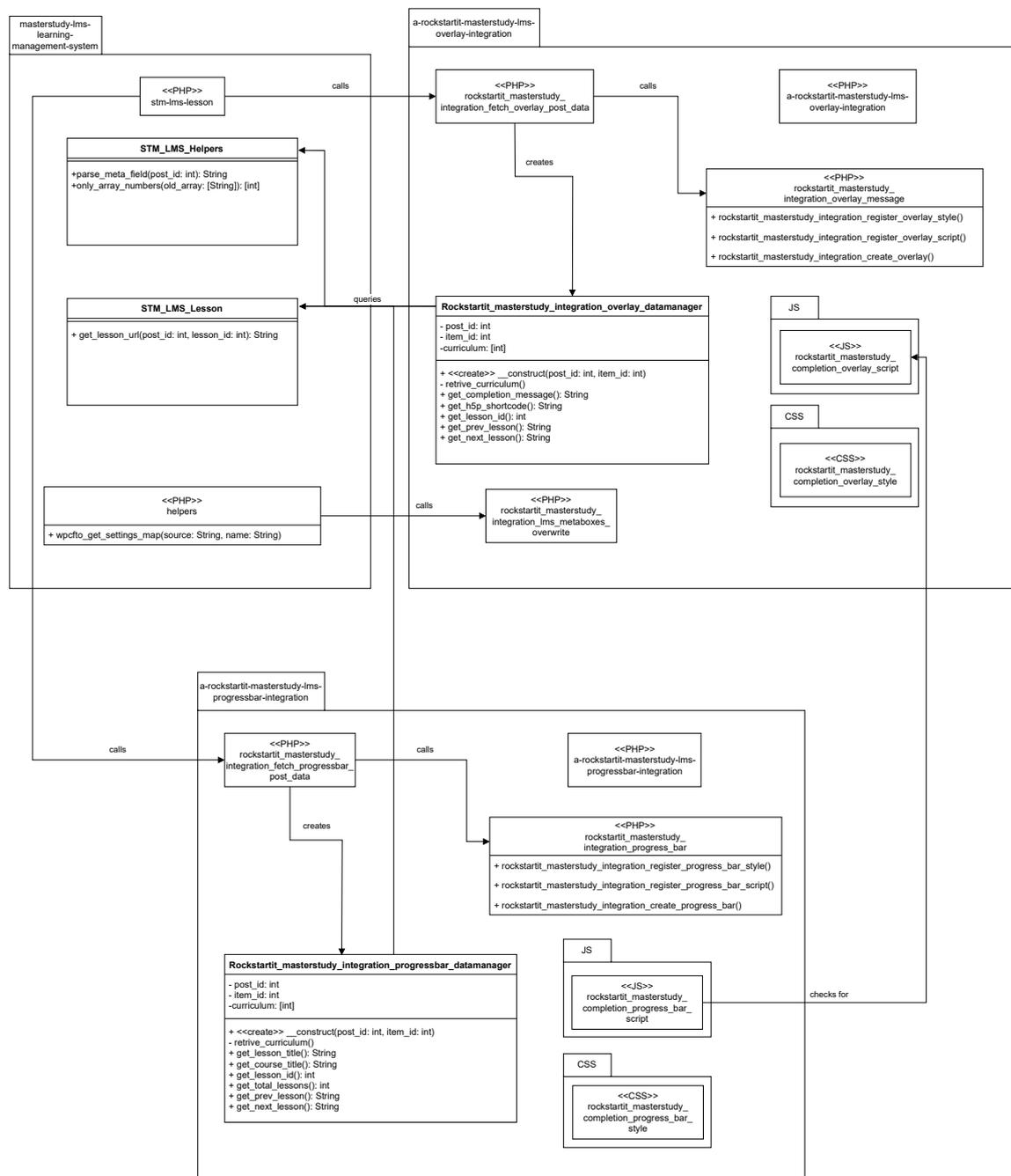


Abbildung 5.10: Modulare Aufteilung des Plugins.

```

1 //Moves progressbar into the footer below the complete button and checks if element from
  overlay plugin is loaded
2 if ($("#rockstartit_masterstudy_integration_complete_container").length){
3   $("#rockstartit_masterstudy_integration_complete_container").parent().wrap("<div id='
  rockstartit_masterstudy_integration_progress_flex'></div>");
4 } else {
5   $(".stm-lms-lesson_navigation_complete").parent().wrap("<div id='
  rockstartit_masterstudy_integration_progress_flex'></div>");
6 }

```

Listing 5.8: Änderung im JS Code für den Fortschrittsbalken

5.7 Anpassungen nach Codereview

Im Rahmen der Arbeit war vorgesehen, dass der vorhandene Code von einer dritten Person reviewed wird. Grundsätzlich war das Ergebnis ein positives. Hier werden aber nochmal einige der Anmerkungen präsentiert.

5.7.1 Klärung von Variablennamen

Um bei der Anwendung der *Hooks* nicht einfach nur magische Zahlen zu verwenden, wurden Variablen eingeführt, welche die Parameter, in denen sie eingesetzt werden, erklären sollen. Im Listing 5.9 ist eine der drei Stellen, in denen dies der Fall ist, als Beispiel aufgeführt.

```

1 $priority = 10;
2 $used_parameters = 2;
3 //Use existing action Hook in masterstudy to retrieve post data
4 add_action('stm_lms_lesson_started', 'rockstartit_masterstudy_integration_retrieve_post_data'
  , $priority, $used_parameters);

```

Listing 5.9: Beispiel für die Variablen

Es wurde aber angemerkt, dass der Einsatz der Variable nicht eindeutig aus dem Code hervorgeht. Zur Verbesserung wurde zum Variablennamen der Name des *Hooks* hinzugefügt sowie ein beschreibender Kommentar ergänzt. Somit sieht diese Stelle nun wie im Listing 5.10 aus.

```

1 $stm_lms_lesson_started_priority = 10; //priority vale for the used hook. set to default
  value of 10
2 $stm_lms_lesson_started_used_parameters = 2; //number of used paramateres, provided by hook.
  First two are used.
3 //Use existing action Hook in masterstudy to retrieve post data
4 add_action('stm_lms_lesson_started', 'rockstartit_masterstudy_integration_retrieve_post_data'
  , $stm_lms_lesson_started_priority, $stm_lms_lesson_started_used_parameters);

```

Listing 5.10: Hinzufügen von Kommentaren zu Variablen

5.7.2 Überflüssige Kommentare

Es ist aufgefallen, dass viele Kommentare zu Methoden in der Klasse *Rockstarit_masterstudy_integration_datamanager* beschreiben, welche anderen Stellen im Code diese Methoden verwenden. Solche Kommentare sind aber überflüssig. Wichtig ist, was die Methode macht, nicht an welchen Stellen diese benutzt wird. Als Beispiel ist hier die Änderung Listing 5.11 von Listing 5.12 zu angeben.

```
1  /**
2   * This method gets the reference to the next lesson, which is used by the overlay in
   the "weiter" button.
3   * @return String The URL of the next lesson as HTML
4   */
5  function get_next_lesson() {
```

Listing 5.11: Version mit überflüssigen Kommentaren

```
1  /**
2   * Gets the URL of the next lesson.
3   * @return String The URL of the next lesson as HTML
4   */
5  function get_next_lesson() {
```

Listing 5.12: Version mit angepassten Kommentaren

5.7.3 Entfernen nutzloser expliziter Variablen

In der Klasse *Rockstarit_masterstudy_integration_datamanager* gab es in einer Methode eine explizite Variable, die nicht notwendig war. Der Code in Listing 5.13 ergab sich aus einer falschen Annahme darüber, dass das Programm andernfalls nicht richtig funktionieren würde.

```
1  function get_h5p_shortcode() {
2      $h5p_shortcode = get_post_meta($this->item_id, 'h5p_shortcode', true);
3      return $h5p_shortcode;
4  }
```

Listing 5.13: Version nutzloser expliziter Variable

Wichtig ist, dass ein entsprechender Entwicklerkommentar gefehlt hat. Auch wenn sich also herausgestellt hat, dass die zusätzliche explizite Variable überflüssig war, sollten solche Entscheidungen für zukünftige Entwickler durch einen Kommentar begründet werden.

5.7.4 CSS Werte im JS

In der JS-Datei, die zum Fortschrittsbalken gehört, wird die von jQuery zur Verfügung gestellte Methode `.css()` verwendet. Über diese wird direkt der CSS Wert eines Elementes verändert. Dieser Code dient zur Färbung des Hintergrunds in Grün, sollte die Lektion abgeschlossen sein und ist in Listing 5.14 abgebildet.

```
1 //Change color if lesson was already completed
2 if ($('#stm_lms_complete_lesson').hasClass('completed')) {
3     $('#rockstartit_masterstudy_integration_lesson_pogress_container').css({"background-
4     color": "#17d292", "color": "#ffffff"});
5 }
6 $(document).on('click', '#rockstartit_masterstudy_integration_complete_container',
7     function(){
8     //Change color if clicked, won't be noticed if color is already this way
9     $('#rockstartit_masterstudy_integration_lesson_pogress_container').css({"background-
10    color": "#17d292", "color": "#ffffff"});
11 });
```

Listing 5.14: Änderung von CSS in jQuery

Das Problem ist hier, dass CSS Werte in einer JS-Datei vorkommen. Damit sind diese nicht mehr klar voneinander getrennt. Nach weiterer Recherche konnte jedoch keine Möglichkeit gefunden werden, diese Werte aus einer externen Datei zu laden. Immerhin sollte, im Falle eines gleichen Wertes wie hier, die Auslagerung in eine Variable stattfinden. Änderungen müssen dann nur einmal vorgenommen werden. Listing 5.15 stellt die geänderte Version dar.

```
1 //it is probably not possible to have the css in a sperate file. At least it can be
2 //factored out into a variable.
3 var completion_style = {"background-color": "#17d292", "color": "#ffffff"};
4 //Change color if lesson was already completed
5 if ($('#stm_lms_complete_lesson').hasClass('completed')) {
6     $('#rockstartit_masterstudy_integration_lesson_pogress_container').css(
7     completion_style);
8 }
9 $(document).on('click', '#rockstartit_masterstudy_integration_complete_container',
10     function(){
11     $('#rockstartit_masterstudy_integration_lesson_pogress_container').css(
12     completion_style);
13 });
```

Listing 5.15: Auslagerung der CSS Werte

6 Evaluation des Fortschrittbalkens

In der Evaluation, die im Rahmen dieser Arbeit durchgeführt wurde, sollte geprüft werden, ob Fortschrittbalken einen positiven Einfluss auf die Bearbeitung des Kurses haben. Eine Evaluation des Overlays ist nicht für diese Arbeit vorgesehen. Dies liegt daran, dass für ein angemessenes Feedback der gesamte Kontext des Kurses miteinbezogen werden muss und dies den Umfang der Arbeit übersteigt. Es wurden aber eine Implementierung, sowie in Abschnitt 7.2 ein Vorschlag für eine Evaluation realisiert, was es möglich macht, in zukünftigen Arbeiten auch der Wirkung des Overlays nachgehen zu können.

6.1 Design

Bei der hier durchgeführten Studie handelt sich um ein über den Verlauf von zwei Wochen durchgeführtes Quasi-Experiment, bei dem alle Probanden die Tests ohne Beobachtung Online durchgeführt haben. Da es im Zeitrahmen dieser Arbeit nicht möglich war, mehrere Gruppen von Schülerinnen und Schülern zu untersuchen, wurden stattdessen Kommilitoninnen und Kommilitonen, sowie Studierende aus anderen Fachbereichen eingeladen. Diese erhielten die Anweisung, einen Kurs zu einem Überthema in „Rockstartit“ zu bearbeiten und anschließend einen dazugehörigen Fragebogen auszufüllen. Abgesehen davon wurden diesen keine weiteren Informationen gegeben, damit sie ein möglichst unbeeinflusstes Ergebnis abgeben konnten.

Alle Probanden wurden zufällig aufgeteilt in eine Kontrollgruppe, die den Kurs ohne einen Fortschrittbalken bearbeiteten und eine Testgruppe, die den Kurs mit einem Fortschrittbalken bearbeiteten. Aus technischen Gründen wurde zuerst der Kontrollgruppe eine Woche Zeit für die Bearbeitung gegeben und im Anschluss erhielt die Testgruppe eine Woche Zeit für die Bearbeitung. Für beide Gruppen wurde der Kurs „Bienenkunde für die KI“ aus dem Überthema „Rettet die Bienen!“ verwendet. In diesem werden die Grundlagen von Künstlicher Intelligenz eingeführt, motiviert durch die automatisierte Datenverarbeitung über Bienen. Der Kurs hat einen Umfang von 14 Seiten und ist damit einer der kleineren zu diesem Überthema. Die Auswahl war entsprechend dadurch begründet, dass bei einem kleineren Kurs die Änderungen am Fortschrittbalken von Seite zu Seite größer sind. Nach Abschluss des Kurses füllten die Probanden einen Fragebogen aus, dessen Fragen im Zusammenhang mit der erwarteten Wirkung des Fortschrittbalkens in Zusammenhang standen.

6.2 Motivation und Hypothese

In Abschnitt 3.1 wurden andere Arbeiten zu Fortschrittbalken betrachtet. Dabei konnte festgestellt werden, dass es im Rahmen von Online-Plattformen, die Kurse für Schüle-

rinnen und Schüler anbieten sollen, keine Studien gibt, die spezifisch die Wirkung von Fortschrittsbalken untersuchen. Für die Arbeiten, die in einem Lehrkontext gesetzt sind, wird der Fortschrittsbalken nicht alleinstehend untersucht. Stattdessen werden, meist im Kontext der Effektivität von Gamification, mehrere Methoden auf einmal getestet und der Fortschrittsbalken ist nur einer unter mehreren Elementen. Deshalb soll diese Evaluation erste Schritte zur Untersuchung von Fortschrittsbalken im Rahmen von durch Onlinekurse unterstützte Lehre, machen. Als Testumgebung eignen sich die Kurse von Rockstartit, da diese sich einerseits an Schüler richten und sie des Weiteren auch eine gute Länge haben. Wenn ein Kurs mit einer angemessenen Anzahl an Seiten, also weder zu groß noch zu klein, gewählt wird, kann der erreichte Fortschritt besser visualisiert werden und erzielt damit möglicherweise auch eine stärkere Wirkung. Dies ist bei dem gewählten Kurs der Fall.

Aus den anderen Arbeiten, in denen, zumindest unter anderem, Fortschrittsbalken untersucht wurden, gehen die folgenden Hypothesen hervor:

- H1 Der Fortschrittsbalken erzeugt Motivation, den Kurs bearbeiten zu wollen.
- H2 Der Fortschrittsbalken vermittelt den Schülerinnen und Schülern einen besseren Überblick über den Umfang des Kurses.
- H3 Der Fortschrittsbalken vermittelt den Eindruck, etwas erreicht zu haben.
- H4 Der Fortschrittsbalken wird nicht negativ auffallen.

In dieser Auflistung sticht die vierte Hypothese ein wenig heraus, da es nicht um einen positiven Einfluss geht. Etwas, was aber auch über andere Studien im Lehrkontext zu sehen war, war, dass die Fortschrittsbalken zumindest nicht negativ auffielen. Das heißt also, dass sie möglicherweise nicht jedem helfen, aber zumindest niemanden behindern sollten. Dies könnte bedeuten, dass Fortschrittsbalken eingesetzt werden können, auch wenn kein positiver Effekt messbar sein sollte.

6.3 Instrument - Konzeption und Begründung

In diesem Abschnitt werden die im Fragebogen verwendeten Fragen vorgestellt, zusammen mit einer Begründung für die Auswahl. Grundsätzlich besteht die Zielgruppe aus Schülerinnen und Schülern, weshalb für eine Lockerung der Atmosphäre die zweite Person als Ansprache in den Fragen verwendet wird. Um ebenfalls mögliche Zusammenhänge mit Geschlecht und Alter überprüfen zu können, wird zunächst jeweils eine Frage dazu gestellt:

1. Dein Geschlecht
2. Dein Alter

Bei ersterem besteht die Möglichkeit zwischen männlich, weiblich oder sonstigem (mit Freitextantwort) zu wählen. Bei dem Alter gibt es einen kurzen Freitextblock, in dem dieses eingegeben werden kann.

Als Nächstes kommen die Fragen, die einen direkten Vergleich zwischen Test- und Kontrollgruppe ermöglichen sollen. Personen, welche die Umfrage ausfüllen, müssen zu Aussagen bewerten, wie sehr diese auf sie zutreffen. Dafür sind die fünf zugehörigen Fragen in einer Matrix angeordnet, bei der zu jeder Aussage eine der Werte „Trifft zu“, „Trifft eher zu“, „Teils-Teils“, „Trifft eher nicht zu“ oder „Trifft nicht zu“ zugeordnet werden kann. Die Überfrage lautet: „Bitte gib zu den folgenden Aussagen an, wie sehr diese auf dich zutreffen.“ In der folgenden Auflistung der dazugehörigen Aussagen wird ebenfalls erklärt, wie diese im Zusammenhang zu den Hypothesen stehen soll. Diese wären:

3. Ich habe mich motiviert gefühlt, den Kurs zu bearbeiten.
Diese Aussage soll die Motivation der Schülerinnen und Schüler prüfen, hauptsächlich um Daten für H1 zu erhalten.
4. Die Bearbeitung des Kurses hat mir Spaß gemacht.
Da Spaß ebenfalls mit der Motivation im Zusammenhang steht, sollte dies auch beachtet werden. Die Ergebnisse können ebenfalls bei der Auswertung von H1 in Betracht gezogen werden. Außerdem könnten daraus ebenfalls Schlüsse über H4 gezogen werden, falls die Bearbeitung mit Fortschrittsbalken weniger Spaß machen sollte.
5. Ich hatte mit jeder abgeschlossenen Seite das Gefühl mich einem Ziel zu nähern.
Hier werden sowohl H3 als auch ein wenig H1 adressiert, da die Aussage einerseits Rückschlüsse über das Gefühl etwas erreicht zu haben gibt, aber auch zur Motivation. Das Gefühl etwas erreicht zu haben kann zudem auch von Motivation zeugen.
6. Nach dem Abschließen einer Seite, hatte ich das Gefühl, etwas erreicht zu haben.
Diese Aussage ist explizit danach gerichtet, Daten für H3 zu erheben
7. Ich hatte einen guten Überblick über den Umfang des Kurses.
Hier soll geprüft werden, ob im Schnitt tatsächlich unterschiedliche Werte, mit und ohne Fortschrittsbalken angegeben werden, um H2 nachzugehen.

Im Anschluss kommen noch zwei Fragen, die nur an die Testgruppe gestellt werden. Die Probanden müssen zwei Aussagen auf einer Skala, mit fünf Abstufungen, von „Hilfreich“ bis „Störend“ einordnen:

6. Ich empfand die Navigationsleiste als...
7. Ich empfand den Fortschrittsbalken als...

Die erste der beiden Aussagen soll nur dazu dienen davon abzulenken, dass es ausschließlich um den Fortschrittsbalken geht. Die zweite Aussage dient nochmal explizit dazu, H4 nachgehen zu können.

Zum Abschluss kommen noch zwei Fragen, die in einem Freitext optional beantwortet werden können, sollten die Probanden weitere Anmerkungen haben:

8. Hättest du noch Vorschläge etwas an dem Fortschrittsbalken zu ändern?
9. Hast du noch sonstige Anmerkungen?

Wichtig ist, dass die erste der beiden Fragen nur an die Testgruppe gestellt wird. Die Kontrollgruppe erhält stattdessen die Frage „Fändest du einen Fortschrittbalken hilfreich?“ Damit kann ausgemessen werden, ob das Vorhandensein eines Fortschrittbalkens ein expliziter Wunsch der Probanden wäre.

6.4 Ergebnisse

Von den 21 Probanden, die teilnahmen, waren 9 in der Kontrollgruppe und 12 in der Testgruppe. Aus der Kontrollgruppe waren 8 Personen männlich und 1 Person weiblich. In der Testgruppe waren 6 Personen männlich und 6 Personen weiblich. Der Altersdurchschnitt betrug ungefähr 21,8 in der Kontrollgruppe, in der Testgruppe hingegen 23,6. Als Zusammenfassung kann dafür auch Tabelle 6.1 betrachtet werden. Die folgende Tabelle

Daten	Kontrollgruppe	Testgruppe	Summe
Teilnehmer gesamt	9	12	21
Anzahl männlich	8	6	14
Anzahl weiblich	1	6	7
Altersdurchschnitt	21,8	23,6	

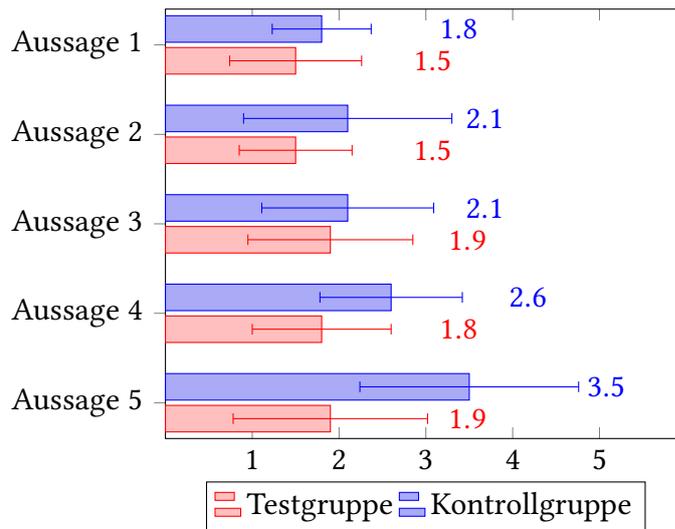
Tabelle 6.1: Informationen über die Probanden

6.2 zeigt die gemeinsam beantworteten Aussagen, sowie den Durchschnittswert(\bar{x}) und die Standardabweichung (SD) in der Kontroll- und in der Testgruppe. Dabei entspricht 1 „trifft zu“, 2 „trifft eher zu“, 3 „teils-teils“, 4 „trifft eher nicht zu“ und 5 „trifft nicht zu“. Die Abbildung 6.1 ist eine zusätzliche visuelle Darstellung der Werte in Tabelle 6.2 als Balkendiagramm. Die Aussagen sind dort ebenfalls in der gleichen Reihenfolge aufgelistet. Dann gab es noch die gruppenspezifischen Fragen. Diese konnten in einem Spektrum be-

Aussage	\bar{x} Kontroll	SD Kontroll	\bar{x} Test	SD Test
Ich habe mich motiviert gefühlt, den Kurs zu bearbeiten.	1,8	0,57	1,5	0,76
Die Bearbeitung des Kurses hat mir Spaß gemacht.	2,1	1,2	1,5	0,65
Ich hatte mit jeder abgeschlossenen Seite das Gefühl mich einem Ziel zu nähern.	2,1	0,99	1,9	0,95
Nach dem Abschließen einer Seite, hatte ich das Gefühl, etwas erreicht zu haben.	2,6	0,82	1,8	0,8
Ich hatte einen guten Überblick über den Umfang des Kurses.	3,5	1,26	1,9	1,12

Tabelle 6.2: Durchschnittswerte und Standardabweichungen zu den Aussagen

antwortet werden, bei denen der Wert 1 „Hilfreich“ und der Wert 5 „Störend“ bedeutet. Die Kontrollgruppe hat ihre Frage „Fändest du einen Fortschrittsbalken hilfreich?“ mit einem Durchschnitt von 1,6 (Standardabweichung: 0,82) beantwortet. Die Testgruppe antwortete im Durchschnitt auf die ihr präsentierte Aussage „Ich empfand den Fortschrittsbalken als...“ mit dem Wert 1,5 (Standardabweichung: 1,04). Es gab auch noch die Aussage „Ich empfand die Navigationsleiste als...“, die lediglich als Störfrage diente, um vom Fortschrittsbalken abzulenken. Der Durchschnittswert der Antworten betrug hier 2,1 (Standardabweichung: 0,69).



Die Aussagen hier sind in der gleichen Reihenfolge wie in Tabelle 6.2.
Ein niedriger Wert heißt, die Aussage trifft eher zu.

Abbildung 6.1: Übersicht zu den Ergebnissen.

6.5 Diskussion

Zunächst sei nochmal erwähnt, dass die Teilnehmerzahl mit 21 Teilnehmern tendenziell eher als klein gesehen werden muss und dass es leichte Abweichungen in der Gruppengröße zugunsten der Testgruppe gab. Beides könnte möglicherweise die Ergebnisse beeinflusst haben. Weiter war die Kontrollgruppe überwiegend männlich und das Alter beider Gruppen nicht im Zielbereich von Schülerinnen und Schülern. Dennoch scheint es so zu sein, als hätte der Fortschrittbalken in allen Aspekten einen positiven Einfluss gehabt. Am kleinsten ist der Unterschied bei allen Aussagen, die in Bezug zur Motivation gestellt wurden. Auch wenn im Schnitt von der Testgruppe eher behauptet wurde, diese Aussagen würden zutreffen, ist es schwierig, bei diesen kleinen Unterschieden H1 als bestätigt zu sehen. Außerdem spielt der Kurs selbst hier eine große Rolle, während der Fortschrittbalken eher als Unterstützung arbeitet. Möglicherweise muss noch sichergestellt werden, dass der Kurs an sich gut ankommt. Dennoch scheint es wertvoll, dem in größeren Studien weiter nachzugehen.

Wie zu erwarten war, konnte jedoch bei der Aussage, die in Verbindung zu H2 stand, ein klarer Unterschied gesehen werden. Da ein Fortschrittbalken ein Element ist, das dazu dient den Fortschritt in einem Prozess auszudrücken, führt dieses auch zu einem besseren Überblick über den Umfang des Kurses. Der Unterschied ist hier auch sichtlich am größten. Mit einem durchschnittlichen Wert von 3,5 liegt die Kontrollgruppe zwischen „Teils-teils“ und „trifft eher nicht zu“, während die Testgruppe mit einem Durchschnittswert von 1,9 auf „trifft eher zu“ liegt. Dennoch wäre es sicherlich hilfreich, H2 in einer größeren Studie nochmal bestätigen zu können.

Weiter sind die Unterschiede bei den Aussagen, die in Verbindung zu H3 gestellt wurden, eher klein. Bei der Aussage „Nach dem Abschließen einer Seite hatte ich das Gefühl etwas erreicht zu haben.“, die am stärksten in Verbindung zu H3 stand, ist noch eher ein Unterschied sichtbar. Die Kontrollgruppe neigte mit einem Durchschnitt von 2,6 eher zu „Teils-teils“, während die Testgruppe mit einem Durchschnitt von 1,8 zu „Trifft eher zu“ neigte. Bei der Aussage „Ich hatte mit jeder abgeschlossenen Seite das Gefühl, mich einem Ziel zu nähern.“, waren die Unterschiede nur sehr klein. Mit Durchschnittswerten von 2,1 bei der Kontrollgruppe und 1,9 bei der Testgruppe lagen beide bei „trifft eher zu“. Der Fortschrittbalken scheint also nur einen geringen Einfluss zu haben, wenn überhaupt, wenn es um das Gefühl geht, sich einem Ziel zu nähern. Allerdings scheint dieser eher einen Unterschied zu machen bei dem Gefühl etwas erreicht zu haben. Möglicherweise ist es auch wichtiger zu sehen, wie sich der Fortschrittbalken füllt und nicht auf was dieser zuläuft. Jedenfalls ist nicht ganz eindeutig H3 als bestätigt zu sehen und es wäre notwendig dem in größeren Studien erneut nachzugehen.

Weiter noch alle Aussagen in Zusammenhang mit H4. Bei der Aussage „Die Bearbeitung des Kurses hat mir Spaß gemacht.“, konnte ein positiver Unterschied gesehen werden. Denn der Durchschnitt der Kontrollgruppe ist 2,1 während der Testgruppe 1,5 ist. Möglicherweise hat ein Fortschrittbalken also nicht nur keinen negativen Einfluss, sondern einen positiven. Dies könnte aus den anderen Faktoren im Zusammenhang mit H1-H3 resultieren. Es gab dann aber noch die direkteren Fragen, die explizit an Kontroll- und Testgruppe gerichtet waren. Die Kontrollgruppe hat auf „Fändest du einen Fortschrittbalken hilfreich?“ im Durchschnitt mit 1,6 also „Hilfreich“ geantwortet. Dies deutet darauf hin,

dass der Fortschrittsbalken sogar vermisst wurde. Ebenso antwortete die Testgruppe auf „Ich empfand den Fortschrittsbalken als...“ mit einem Durchschnitt von 1,5 also ebenfalls „Hilfreich“. Wenn die Auswertung dieser Aussagen, aber auch die zugehörig zu H1-H3 betrachtet werden, kann H4 durchaus als bestätigt gesehen werden. Trotzdem sollte nochmal versucht werden, dies in einer größeren Studie zu bestätigen. Wenn mit genügend Sicherheit ausgeschlossen werden könnte, dass Fortschrittsbalken einen negativen, hindernden oder ablenkenden Einfluss haben, wäre dies möglicherweise bereits Grund genug, sie stets in so einem Kontext zu realisieren.

Abschließend gab es noch ein paar Anmerkungen der Teilnehmer. In der Kontrollgruppe gab es die Anmerkung: „Fortschrittsbalken wäre ganz nett, viel besser wäre aber am Anfang ein kurzer Ausblick, wohin die Reise denn geht.“ Möglicherweise wäre es also eine Überlegung wert, am Anfang eine Art Zusammenfassung zu zeigen, die knapp beschreibt, was getan wird. Weiter gab es noch ein paar Anmerkungen zum Fortschrittsbalken von der Testgruppe. Mehrere gingen darum, noch weitere Informationen hinzuzufügen:

1. „Man könnte die einzelnen Unterkapitel auflisten und markieren wo man sich gerade befindet“
2. „Neben der prozentualen Angabe wäre eine 'n von x Seiten' Angabe interessant“
3. „Ein Vorschlag wäre, dass man bei Fortschrittsbalken nicht nur den abgeschlossen Prozent sieht sondern von Gesamten 100 Prozent. [z.B 80 abgeschlossen und 20 verblieben]“

All diese Vorschläge sollten realisierbar sein und könnten in zukünftigen Iterationen umgesetzt werden. Weiter gab es noch zwei Anmerkungen darüber, dass der Fortschrittsbalken nicht gut aufgefallen wäre:

1. „Weiter oben positionieren - ich habe den Balken gar nicht wahrgenommen und evtl. auch nicht so weit nach unten gescrollt.“
2. „Ich habe ihn mehrmals aus den Augen verloren. Ich hätte ihn größer gemacht, um das Gefühl des Vorankommens zu stärken.“

Möglicherweise könnte der Balken also noch größer gemacht werden. Ein anderer Ansatz, dem aber auch nachgegangen werden könnte, wäre, den Fortschrittsbalken stets am unteren Ende des Bildschirms fixiert sehen zu können. All diese Vorschläge sollten bei zukünftigen Iterationen beachtet werden. Hier könnten auch Evaluationen darüber ausgeführt werden, wie genau der Fortschrittsbalken angezeigt werden soll.

7 Limitationen und zukünftige Arbeiten

Abschließend werden noch Limitationen, sowie mögliche Ansätze für zukünftige Arbeiten diskutiert.

7.1 Limitationen

In Abschnitt 5.4 wurden bereits alle festgestellten Limitationen angesprochen, die im Kontext der Implementation aufgefallen sind. Zusammengefasst gab es zwei große Probleme: Erstens ist „Masterstudy LMS“ ein kommerzielles Produkt mit Pro-Versionen, weshalb der Code nicht für Modifikationen von dritten ausgelegt ist. Neben dem Fehlen von Dokumentation gibt es – und das ist noch viel wichtiger – nur mäßig geeignete *Hooks* zum Hinzufügen neuer Funktionalität. Dies stellt ein großes Problem dar, da direkte Modifikationen nicht unter WordPress vorgesehen sind. Würden dennoch welche durchgeführt werden, gingen diese mit jedem Update des Plugins verloren. Zweitens, ist für die Nutzung der Kurse ein Account notwendig, was, in einem Kontext des Einsatzes in einer Schule, eine größere Einstiegshürde darstellt. Weiter konnte auch keine Möglichkeit gefunden werden, dies auf eine zufriedenstellende Art zu umgehen. Auch wenn eine Implementation für sowohl ein Overlay mit einstellbaren Nachrichten, als auch einen Fortschrittsbalken gefunden wurden, wurde nur für zweiteres eine Evaluation durchgeführt. Dennoch war die Implementation des Overlays ein wichtiger Bestandteil der Arbeit, um abschätzen zu können, wie gut sich mit den gegebenen Frameworks arbeiten lässt.

Weiter konnten keine Studien zu dieser speziellen Art Feedback in einem Overlay zu vermitteln gefunden werden. Weitere Forschungen in diesem Bereich wären jedoch über den zeitlichen Rahmen dieser Arbeit hinausgegangen. Genauso konnten keine Arbeiten gefunden werden, in denen ein Fortschrittsbalken, eingesetzt für Bildungszwecke, als einzelnes Element untersucht wurde. Dies hat die in Kapitel 6 durchgeführte Evaluation motiviert, für die jedoch auch folgende Limitationen festgestellt werden konnten: Zuerst war mit 21 Teilnehmern der Umfang der Teilnehmer noch eher klein. Weiter hat die Altersgruppe nicht ganz zur Zielgruppe von Schülern in den Klassen sieben bis neun gepasst, da für diese Arbeit keine Schulklasse organisiert werden konnte. Zusätzlich war auch der Umfang der Fragen sehr klein und sollte mehr dazu dienen, ein Stimmungsbild wiederzugeben, das als Motivation für zukünftige Forschung dienen kann. Zuletzt gab es auch Kritik an der Art des Fortschrittsbalkens selbst. Der Fortschrittsbalken war, sobald die Seite geladen wurde, ein statisches Element. Es war auch nur eine Prozentzahl des Fortschritts sichtbar, statt einer Anzeige, um die wievielten Seite es sich aus der Gesamtzahl der Lektionen handelt. Visuell gibt es keine detaillierte Indikation, wie beispielsweise kleine Striche auf dem Fortschrittsbalken für jede Lektion. Wie viele Lektionen es insgesamt gibt, hat momentan auch einen Einfluss darauf, wie schnell sich der Fortschrittsbalken füllt, was ab

einer gewissen Größe den Eindruck von langsamem Fortschritt geben kann. Konkret kann dies also auch bedeuten, dass die Ergebnisse mit der Größe des Kurses zusammenhängen könnten. Es gab hier auch keine Maßnahmen um dem entweder durch Animation oder einem abgeänderten, nicht-linearem, Füllverhalten entgegenzuwirken. Zuletzt wurde der Fortschrittsbalken am Ende der Seite nicht immer von allen wahrgenommen.

7.2 Zukünftige Arbeiten

Zuerst sei angemerkt, dass die gegebenen Frameworks von „WordPress“ und insbesondere „MasterStudy LMS“ nur sehr eingeschränkt für Erweiterungen dieser Art geeignet waren. Entsprechend wäre für zukünftige Arbeiten zu überlegen, ob die Verwendung einer weniger einschränkenden Plattform oder das Erstellen eines eigenen Plugins langfristig eine gute Entscheidung wäre. Dies gibt auch mehr Kontrolle über das Anknüpfen anderer Forschungspunkte, wie beispielsweise Analyse des Lernverhaltens. Weiter könnte mit Forschungen zum Einsatz von Overlays, um Feedback mitzuteilen, begonnen werden. Erste Schritte in diesem Gebiet könnten auf eine ähnliche Art und Weise, wie die Evaluation in dieser Arbeit erfolgen. Bei einer kleineren Menge an Testpersonen könnte geprüft werden, ob ein Unterschied festgestellt werden kann, wenn einmal das Overlay eingesetzt wird und einmal nicht. Mögliche Kriterien, die sich hier überprüfen ließen, wären beispielsweise Motivation, Interesse und Verständnis. Diese können auch gut in die Ebenen des Modells von Hattie und Timperley eingearbeitet werden (siehe 2.1). Denn ein größeres Gefühl für das eigene Verständnis erzeugt eine verbesserte Wahrnehmung der zu erreichenden Ziele und mit mehr Motivation und Interesse kann besser auf diese hingearbeitet werden.

Auch zum Einsatz von Fortschrittsbalken gibt es noch weitere Möglichkeiten für Untersuchungen. Zum einen könnte eine ähnliche Evaluation mit Schülerinnen und Schülern durchgeführt werden. Bei einem Vergleich zwischen zwei Klassen würde es sich um eine größere und gezieltere Stichprobe handeln. Damit wäre eine bessere Generalisierbarkeit der Daten gegeben. Zusätzlich ließen sich aber auch nochmals Studien zu Anpassungen an den Eigenschaften des Fortschrittsbalkens durchführen. Passend zu den Anmerkungen könnte geschaut werden, ob es einen Unterschied in der Wahrnehmung macht, wenn der Fortschrittsbalken am unteren Bildschirmrand fixiert ist statt am Ende einer Lektion. Weiter kann auch geprüft werden, ob es sinnvoll ist noch mehr Details hinzuzufügen, wie das Nennen der Seitenzahl, entweder über Text oder visuell durch Striche im Fortschrittsbalken. Möglicherweise wäre auch das Hinzufügen einer Animation sinnvoll. Als mögliches Beispiel könnte beim Laden der Seite angezeigt werden, wie sich der Fortschrittsbalken zum aktuellen Punkt füllt. Abschließend sei auch angemerkt, dass es generell im Bildungskontext, neben dem allgemeinen Einsatz, noch keine oder kaum Studien zu unterschiedlichen Füllverhalten von Fortschrittsbalken zu geben scheint. Möglicherweise könnten sich unterschiedliche Verhalten zum Füllen eines Fortschrittsbalkens, wie in 3.1 beschrieben, auch unterschiedlich auf die Motivation auswirken. Daher könnte es auch sinnvoll sein, in diesem Bereich ebenfalls weiterer Forschung nachzugehen. Generell gilt für alle genannten Modifikationen am Fortschrittsbalken, dass diese in Studien expliziter auf das Erzeugen von Motivation geprüft werden können.

8 Fazit

In dieser Arbeit wurde untersucht, welche Möglichkeiten es gibt, Feedback zu vermitteln, sowie was gutes Feedback ausmacht. Genauer wurden Ansätze zu Vermittlung von Feedback betrachtet, die Einsatz in Onlinekursen gefunden haben. Weiter wurden spezifisch Arbeiten zu zwei Formen von Feedback untersucht. Einerseits das Vermitteln von Feedback durch Kommentare und andererseits in Form eines Fortschrittsbalkens. Gerade bei Fortschrittsbalken konnte festgestellt werden, dass es im Kontext des Einsatzes in Onlinekursen nur wenig bis keine Forschung darüber zu geben scheint, ob dieser Ansatz für sich allein genommen Wirkung zeigt. Auch zur Wirkung von Kommentaren, die spezifisch in Onlinekursen als Overlay erscheinen, konnten keine Arbeiten gefunden werden. Daher wurden beide ausgewählt, um weiter untersucht zu werden.

Dafür wurde beschlossen, das Overlay und den Fortschrittsbalken auf einer Seite, die Onlinekurse anbietet, zu implementieren. Als solche Seite wurde „Rockstartit“ gewählt, die über WordPress zusammen mit dem Plugin „MasterStudy LMS“ entwickelt wurde. Da „MasterStudy LMS“ und dementsprechend auch „Rockstartit“ nicht bereits über diese Ansätze für Feedback verfügt, schien dies für Untersuchungszwecke eine gute Wahl darzustellen. Dabei konnte bereits zu Beginn festgestellt werden, dass es nicht vorgesehen ist, weder WordPress noch fremde Plugins, direkt zu modifizieren. Auch wenn das grundsätzlich möglich ist, werden jegliche Änderungen bei einem Update überschrieben. Stattdessen sollen eigene Plugins entwickelt werden, falls weitere Funktionalität gewünscht ist. Um von einem neuen Plugin auf Abläufe anderer Plugins oder auch von WordPress selbst zugreifen zu können, sind sogenannte *Hooks* entwickelt worden. Diese realisieren eine Art Beobachter-Muster, bei dem Methoden an einen *Hooks* angemeldet werden können. Wird der *Hooks* ausgelöst, werden alle angemeldeten Methoden aufgerufen. Es können sogar Parameter über den *Hooks* mitgegeben werden. Darüber konnten auch der Fortschrittsbalken und das Overlay, bei dem die angezeigten Nachrichten, für jede Seite der Kurse eingestellt werden können, realisiert werden. Allerdings sind zwei Probleme festgestellt worden:

Erstens ist „Masterstudy LMS“ ein kommerzielles Produkt mit Pro-Versionen, weshalb der Code nicht für Modifikationen von Dritten ausgelegt ist. Es fehlen also auch die entsprechenden *Hooks*, um eine sauberere Implementation realisieren zu können. Zweitens ist „MasterStudy LMS“ in einigen Designaspekten, insbesondere dem Zwang zur Erstellung eines Accounts, unfreundlich zum Einsatz in Schulen. Nachhaltige Modifikationen, dies abstellen zu können, sind bisher nicht erfolgt. Zu der Implementation lässt sich also zusammenfassend sagen, dass die Modifikation anderer Plugins stark von den vorhandenen *Hooks* abhängt. Auch wenn die Möglichkeit besteht, die Entwickler von „MasterStudy LMS“ um die Ergänzung weiterer *Hooks* anzufragen, scheint es auf langer Sicht sinnvoller einen eigenen Ansatz anzufertigen, sei es ein Plugin in WordPress oder eine ganz eigene Webseite.

Abschließend wurde mit dem implementierten Fortschrittsbalken noch eine Evaluation durchgeführt. Ziel war es zu untersuchen, ob der Fortschrittsbalken einen positiven Einfluss auf die Motivation, den Überblick oder dem Gefühl etwas zu erreichen, während der Bearbeitung des Kurses hat. Dabei konnte gezeigt werden, dass es einen Unterschied macht, ob ein Fortschrittsbalken eingesetzt wird oder nicht. Zum Overlay wurden im Rahmen dieser Arbeit keine Untersuchungen durchgeführt, aber es wurde ein Ansatz vorgeschlagen. Ähnlich wie für den Fortschrittsbalken, ließen sich durch verwandte Arbeiten Hypothesen über die Wirkung aufstellen, die in einer Evaluation geprüft werden können. Für die Evaluation des Fortschrittsbalkens wurden alle Teilnehmer in zwei Gruppen aufgeteilt. Eine Kontrollgruppe, die einen Kurs ohne Fortschrittsbalken bearbeitet hat und eine Testgruppe, die den gleichen Kurs mit Fortschrittsbalken bearbeitet hat. Im Anschluss der Bearbeitung, sollten beide Gruppen einen kleinen Fragebogen ausfüllen, bei dem zur Vergleichbarkeit nahezu alle Fragen gleich waren. In allen Aspekten konnten, wenn teilweise auch nicht in hinreichender Größe, positive Einflüsse festgestellt werden. Vor allem, wenn es darum geht, einen besseren Überblick über den Umfang des Kurses zu liefern, scheinen Fortschrittsbalken einen positiven Einfluss zu haben. Genauso legt das Ergebnis nahe, dass Fortschrittsbalken zumindest an keiner Stelle einen negativen Einfluss haben. Die zwischen beiden Gruppen unterschiedliche Frage zielte darauf ab, zu erfahren, ob der Fortschrittsbalken nützlich war oder ob ein Fortschrittsbalken gewünscht wurde. In beiden Fällen wurde im Durchschnitt eher zugestimmt. Beim Einfluss auf die Motivation und den Eindruck, etwas erreicht zu haben, konnten keine besonders großen Unterschiede festgestellt werden. Es muss aber noch angemerkt werden, dass die Anzahl der Teilnehmer nicht sehr groß war und vom Alter etwas älter als die gewünschte Zielgruppe war. In Zukunft sollten also größere Studien mit Schülerinnen und Schülern durchgeführt werden. Weiter könnte auch überprüft werden, ob das Ändern gewisser Eigenschaften des Fortschrittsbalkens einen Einfluss hätte. Mögliche Untersuchungspunkte wären noch das Anzeigen von mehr Informationen, ein animierter Fortschrittsbalken oder ein verändertes Verhalten beim Füllen.

Literatur

- [1] Yigal Attali und Fabienne van der Kleij. „Effects of feedback elaboration and feedback timing during computer-based practice in mathematics problem solving“. In: *Computers & Education* 110 (2017), S. 154–169.
- [2] Anderson Pinheiro Cavalcanti u. a. „Automatic feedback in online learning environments: A systematic literature review“. In: *Computers and Education: Artificial Intelligence* 2 (2021), S. 100027.
- [3] Frederick G Conrad u. a. „The impact of progress indicators on task completion“. In: *Interacting with computers* 22.5 (2010), S. 417–427.
- [4] Manfred Constapel u. a. „IKARion: Enhancing a learning platform with intelligent feedback to improve team collaboration and interaction in small groups“. In: *2019 18th International Conference on Information Technology Based Higher Education and Training (ITHET)*. IEEE. 2019, S. 1–10.
- [5] Jane Costello und Daph Crane. „Technologies for learner-centered feedback“. In: *Open Praxis* 5.3 (2013), S. 217–225.
- [6] Erin A Crisp und Curtis J Bonk. „Defining the learner feedback experience“. In: *TechTrends* 62.6 (2018), S. 585–593.
- [7] Phillip Dawson u. a. „What makes for effective feedback: Staff and student perspectives“. In: *Assessment & Evaluation in Higher Education* 44.1 (2019), S. 25–36.
- [8] Sebastian Deterding u. a. „From game design elements to gamefulness: defining "gamification"“. In: *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments*. 2011, S. 9–15.
- [9] Darina Dicheva u. a. „Gamification in education: A systematic mapping study“. In: *Journal of educational technology & society* 18.3 (2015), S. 75–88.
- [10] Lu Ding, ChanMin Kim und Michael Orey. „Studies of student engagement in gamified online discussions“. In: *Computers & Education* 115 (2017), S. 126–142.
- [11] Faridiah Aghadiati Fajri u. a. „Gamification in E-Learning: The Mitigation Role in Technostress.“ In: *International Journal of Evaluation and Research in Education* 10.2 (2021), S. 606–614.
- [12] Caribay Garcia-Marquez und Kristina N Bauer. „An examination and extension of the theory of gamified learning: The moderating role of goal orientation“. In: *Simulation & Gaming* 52.4 (2021), S. 407–434.
- [13] Guillaume Gronier und Alexandre Baudet. „Does Progress Bars' Behavior Influence the User Experience in Human-Computer Interaction?“ In: *Psychol Cogn Sci Open Journal* 5 (2019), S. 6–13.

- [14] Kentaro Hamada u. a. „Color effect on subjective perception of progress bar speed“. In: *2011 Third International Conference on Intelligent Networking and Collaborative Systems*. IEEE. 2011, S. 863–866.
- [15] Chris Harrison u. a. „Rethinking the progress bar“. In: *Proceedings of the 20th annual ACM symposium on User interface software and technology*. 2007, S. 115–118.
- [16] John Hattie. „Influences on student learning“. In: *Inaugural lecture given on August 2.1999 (1999)*, S. 21.
- [17] John Hattie und Helen Timperley. „The power of feedback“. In: *Review of educational research* 77.1 (2007), S. 81–112.
- [18] David Lefevre und Benita Cox. „Delayed instructional feedback may be more effective, but is this contrary to learners’ preferences?“ In: *British Journal of Educational Technology* 48.6 (2017), S. 1357–1367.
- [19] Ying Li u. a. „Shape of progress bar effect on subjective evaluation, duration perception and physiological reaction“. In: *International Journal of Industrial Ergonomics* 81 (2021), S. 103031.
- [20] Uwe Maier, Nicole Wolf und Christoph Randler. „Effects of a computer-assisted formative assessment intervention based on multiple-tier diagnostic items and different feedback types“. In: *Computers & Education* 95 (2016), S. 85–98.
- [21] Fiona Fui-Hoon Nah u. a. „Gamification of education: a review of literature“. In: *International conference on hci in business*. Springer. 2014, S. 401–409.
- [22] David J Nicol und Debra Macfarlane-Dick. „Formative assessment and self-regulated learning: A model and seven principles of good feedback practice“. In: *Studies in higher education* 31.2 (2006), S. 199–218.
- [23] Masakazu Ohtsubo und Kaori Yoshida. „How does shape of progress bar effect on time evaluation“. In: *2014 International Conference on Intelligent Networking and Collaborative Systems*. IEEE. 2014, S. 316–319.
- [24] Marie Olsson, Peter Mozelius und Jonas Collin. „Visualisation and gamification of e-learning and programming education“. In: *Electronic journal of e-learning* 13.6 (2015), pp452–465.
- [25] Anthony A Piña. „An overview of learning management systems“. In: *Virtual Learning Environments: Concepts, methodologies, tools and applications* (2012), S. 33–51.
- [26] Michael Sailer und Lisa Homner. „The gamification of learning: A meta-analysis“. In: *Educational Psychology Review* 32.1 (2020), S. 77–112.
- [27] Diana R Sanchez, Markus Langer und Rupinder Kaur. „Gamification in the classroom: Examining the impact of gamified quizzes on student learning“. In: *Computers & Education* 144 (2020), S. 103666.
- [28] Priscylla Silva, Evandro Costa und Joseana Régis de Araújo. „An adaptive approach to provide feedback for students in programming problem solving“. In: *International Conference on Intelligent Tutoring Systems*. Springer. 2019, S. 14–23.

-
- [29] Fabienne M Van der Kleij, Remco CW Feskens und Theo JHM Eggen. „Effects of feedback in a computer-based learning environment on students’ learning outcomes: A meta-analysis“. In: *Review of educational research* 85.4 (2015), S. 475–511.
- [30] Ekaterina Vasilyeva u. a. „Tailoring feedback in online assessment: influence of learning styles on the feedback preferences and elaborated feedback effectiveness“. In: *2008 Eighth IEEE international conference on advanced learning technologies*. IEEE. 2008, S. 834–838.