



Universität Regensburg

Teach me Code

Abschlussprojekt im Fach Usability Engineering am Institut für Information und
Medien, Sprache und Kultur (I:IMSK)

Vorgelegt von: Lucas Müller, Iuliia Borisenko, Paula Schuhmann
E-Mail (Universität): Lucas.Mueller@stud.uni-regensburg.de, Paula.Schuhmann@stud.uni-regensburg.de, Iuliia.Borisenk@stud.uni-regensburg.de
Abgegeben am: 21.03.2022

Inhaltsverzeichnis

1 Einleitung	5
2 Anforderungserhebung	6
2.1 Interview	6
2.1.1 Entwicklung der Interview-Fragen	6
2.1.2 Feature Requests	11
2.2 Wettbewerbsanalyse	19
2.3 Fokusgruppe	36
3 Anforderungsspezifizierung	45
3.1 Personas	45
3.2 User Stories	54
3.3 Use-Cases	55
3.4 Hierarchische Taskanalyse	56
3.5 Anforderungsliste	59
3.5.1 Nicht-funktionale Anforderungen	59
3.5.2 Funktionale Anforderungen	64
4 Konzept	72
4.1 Apparat	72
4.1.1 Spielertype und Gamification-Elemente	72
4.2 Entscheidung für eine mobile Applikation	79
4.2.1 Entwicklung mobiler Application	80
4.2.2 Interaktion auf der Lernplattform	80
4.3 Studiendesign	81
4.4 Studie	82
4.5 Teilnehmer	83
4.6 Prozedur	84
5 Ergebnisse	85
5.1 NASA-TLX Questionnaire	85
5.2 Kommentare von der Interaktion	86
5.3 Auswertung	87
6 Summative Evaluation	89
6.1 Projektplan und Management	89
6.2 Interviews, Fokusgruppe	89
6.3 Wettbewerbsanalyse	90
6.4 Personas, Use-Cases, User Stories, HTA	91
6.5 Design	93
6.6 Erstellung der Prototypen	93
6.7 Ergebnisse	94

7 Zusammenfassung	96
Literaturverzeichnis	99

Abbildungsverzeichnis

1	Screenshot der Bewertung von "Mimo iOS"-Anwendung	24
2	Beschreibung der App "Learn Python iOS" in AppStore	25
3	Beschreibung der App "Python Recipes iOS, MacOS" in AppStore . .	26
4	Beschreibung der App "Python for beginners iOS" in AppStore . . .	27
5	Startseite der App "Codecademy GO"	28
6	Rezensionen der App "Codecademy GO"	30
7	Screenshot der Bewertung von "SoloLearn iOS"	31
8	Clustering von Patterns	51
9	Persona 1	52
10	Persona 2	53
11	Persona 3	54
12	Use-Cases	102
13	Hierarchische Taskanalyse	103
14	Deskriptive Statistik	103
15	NASA-TLX-Kategorie für gamifizierte Prototype	104
16	NASA-TLX-Kategorie für nicht gamifizierte Prototype	104
17	Mittelwerte von NASA-TLX Kategorien für nicht-gamifizierte und gamifizierte Prototypen	105

1 Einleitung

Folgende Arbeit wurde im Rahmen des Kurses des "Usability Engineering" verfasst. Ziel dieser Arbeit war es, die uns vermittelten theoretischen und praktischen Inhalte zur Gestaltung der User Experience anhand eines Projektes zu verwirklichen. Unser Projekt ist die Konzipierung, Entwurf und Ausgestaltung eines App-Prototypen zum Erlernen der Programmiersprache Python. Dem User-Centered-Design Prozess folgend wurden zunächst Interviews geführt. Ziel dieser Interviews ist es, Wünsche, Anforderungen als auch Erfahrungen mit Apps dieser Art zu erfassen und diese dann in die Anforderungsspezifizierung einfließen zu lassen. Daraufhin folgte die Wettbewerbsanalyse. In dieser werden Lernapps verschiedenster Anbieter hinsichtlich ihrer Funktionalität und Features analysiert. Die Funktionalitäten der Apps wurden in Kategorien unterteilt, anhand diesen wurden die untersuchten Applikation miteinander verglichen. Mittels der Auswertung der Interviews wird eine Fokusgruppe festgelegt an die sich die App als Nutzergruppe festgelegt.

In der Anforderungsspezifizierung wurde die Nutzergruppe konkretisiert, mögliche Interaktionen Anwendungsfälle sowie Schlüsseltasks konzipiert. Im Anschluss werden die Anforderungen an die App konkretisiert.

Mithilfe von Axure, einer Software zur Erstellung von Prototypen für Webanwendung und mobiler Applikationen, werden gemäß der Anforderungen, gamifizierte und nicht-gamifizierte Versionen des Prototypen entwickelt. Abschließend wird ein Usability-Testing durchgeführt. Die Analyse der Testergebnisse wird ergeben, welche von den zwei konkurrierenden Ansätzen, den attraktivieren Lernprozess versprechen kann.

2 Anforderungserhebung

2.1 Interview

2.1.1 Entwicklung der Interview-Fragen

Unser Produkt richtet sich an diejenigen, die keine oder geringe Vorerfahrungen in der Programmiersprache Python haben, aber diese Sprache gerne erlernen wollen. Es wird davon ausgegangen, dass die KundInnen von unserem Produkt sowie die NutzerInnen gleich gesetzt werden können. Potenzielle KundInnen haben in der Regel eine akademischen Hintergrund und sind vielleicht im Umgang mit Programmiersprachen bereits vertraut, aber noch nicht mit der Programmiersprache Python. Das kann bedeuten, dass die KundInnen schon die NutzerInnen von einem ähnlichen Produkt waren, welches ihnen behilflich war, eine andere Programmiersprache zu erlernen. Es kann aber auch sein, dass die KundInnen noch keine Produkte in dieser Richtung benutzt haben. Somit können nicht nur die Fragen über Ziele und Anliegen gestellt werden, sondern auch über das Verhalten des Produktes (Goodwin, 2009, S. 114).

Die erste Phase des Interviews soll sich nicht auf spezifischen Fragen wie der Funktionalität oder dem Design konzentrieren, sondern die Bedürfnisse und Herausforderungen einer bestimmten Situation, eines bestimmten Kontextes klarer machen. Sobald diese Bedürfnisse gut erfasst worden, kann zur Spezifizierung der Anforderungen übergegangen werden. (Lazar, 2017, S. 190).

Die erste Frage soll eine einführende Frage sein, und den beruflichen und persönlichen Hintergrund der potenziellen NutzerInnen erfragen. (Goodwin, 2009, S. 115).

- Könnten Sie uns bitte etwas über sich selbst kurz erzählen-sind sie Studierende/r? In welchem Bereich sind Sie tätig?
- Und haben Sie schon Vorerfahrungen in den Programmiersprachen?

Die InterviewteilnehmerInnen werden gefragt, in welchen Situationen und in welchem Lernumfeld der Lernprozess normalerweise stattfindet - am Laptop oder am Handy. Ist das Erlernen der Programmiersprache für die Unterhaltung gedacht

oder wird die Programmiersprache für berufliche Ziele erlernt. Diese Fragen sollen uns die Informationen liefern, warum und wann man die Programmiersprache Python lernen will (Lazar, 2017, S. 190).

- Nutzen Sie Laptop oder lieber Handy beim Lernprozess?
- Würden Sie Python lernen für die Unterhaltungsziele oder für beruflichen Einsatz?
- Wie möchten Sie lernen - audiell, visuell, motorisch?

Da wir keine langfristigen Beobachtungen durchführen können und damit in unserem Projekt genaue Informationen gesammelt werden können, mit welchen Wahrnehmungssystemen welche Elemente erlernt werden, sollen wir bei solchen self-reporting Fragen danach um präzisere Antworten bitten (Goodwin, 2009, S. 55). Das bedeutet, immer wenn die Befragten die Information über sich selbst geben, oder sich an irgendwas erinnern oder über ihr eigenes gewöhnliches Verhalten beschreiben, soll die Information entweder überprüft werden, oder sie sollen ihre Angaben präzisieren (Goodwin, 2009, S. 136).

Bei der Frage, wie man sich selbst den Lernprozess und individuelle Wahrnehmungssysteme vorstellt, wird um die Details gebeten, indem der Zusammenhang zwischen den Wahrnehmungssystemen und entsprechenden Lernkomponenten herausgefunden wird.

- Und welche Elemente entsprechen Ihrer Meinung nach den visuellen, audiellen und motorischen Wahrnehmungstypen?

Um Ziele unseres Produktes zu verdeutlichen, sollen wir Fragen zu den Funktionalitäten stellen, die von unserem Produkt erwartet werden, z.B. Fragen nach Funktionalitäten, die in anderen Systemen noch nicht existieren (Goodwin, 2009, S. 116). Hier werden explorative Fragen gestellt, die die Kommunikation mit unserem Produkt beschreiben könnten (Lazar, 2017, S. 194).

- Was sind Ihrer Meinung nach wichtige Schritte für das Lernen einer Programmiersprache?

- Wie lernen Sie die Programmiersprachen? oder wie würden Sie sich den Lernprozess vorstellen?
- Wie würden Sie sich den Ablauf einer Übung in der App vorstellen?
- Was erwarten Sie von einer mobilen Anwendung oder vom Websystem für den erfolgreichen Lernprozess für ProgrammieranfängerInnen? Welche Faktoren sind Ihnen bei der Produktauswahl am wichtigsten?

Folgende Fragen sollen schon auf das Ziel näher eingehen-Anforderungen und Frustrationen, Ziele und Zweifel der NutzerInnen beleuchten (Lazar, 2017, S. 193). Sobald wir das Diskussionfeld enger machen, wird es schwierig wieder zu dem breiteren Feld zu kommen (Goodwin, 2009, S. 116).

- Haben Sie schon mal eine Programmiersprache über eine Online Plattform erlernt?
- Was sind heute die größten Probleme oder Ineffizienzen in dem Lernprozess mithilfe von mobilen Applications?
- Haben Sie schon schlechte Erfahrungen oder Probleme bei der Interaktion mit den Produkten gehabt, welche für den Lernprozess der Programmiersprachen entwickelt sind?
- Können Sie uns jetzt solche Systeme zeigen oder nennen, die Sie heute nutzen für das Lernen der Programmiersprachen oder bei denen Sie auf Probleme gestoßen haben?
- Wollten Sie schon irgendwas auf der Web/Appanwendung tun, was aber nicht möglich war? Und was waren das für Schwierigkeiten?
- Auf welche anderen Probleme können Sie beim Lernen auf der Lernplattform stoßen?

Wichtig ist, dass wir keine direkten Fragen stellen, die uns zu unserem Zieldesign führen würden (Lazar, 2017, S. 193). Beispielsweise, die Frage, ob unsere NutzerInnen gerne Lernvideos auf unserer mobilen Lernplattform anschauen möchten, wird mit sicherlich positiven beantwortet werden. Das erklärt aber noch nicht, wie groß die Liste der erwünschten Funktionalitäten bei den Befragten NutzerInnen ist,

und in welchem Bereich der Liste diesen Frage gehören (Goodwin, 2009, S. 123). Zudem gehören diese Fragen über bestimmte Funktionalitäten zu den close-ended-Fragen und sollen lieber in open-ended-Fragen umgewandelt werden. Es ist empfehlenswert, solche Fragen über gewisse Funktionalitäten am Ende des Interviews zu stellen (Goodwin, 2009, S. 123). Statt solcher präzisen Fragen, können breitere Fragen formuliert werden wie z.B. "If you could describe the perfect system for solving your problem, what would it look like?" (Lazar, 2017, S. 193).

Obwohl die Autoren (Lazar, 2017) der Ansicht sind, dass es trotzdem gut ist, bestimmte Fragen über die Interaktion und Funktionalitäten zu stellen, wird in unserem Projekt auf wenige davon eingegangen, damit wir die NutzerInnen nicht dazu zwingen, Lösungen für das Design zu generieren (Goodwin, 2009, S. 124).

- Welche audiovisuellen Elemente sollen berücksichtigt werden?
- Wie würden Sie sich die Eingabe der Daten vorstellen? Wie notwendig finden Sie die Tastatur-Nutzung?
- Soll der Produkt Ihnen an bestimmte Funktionalitäten erinnern können oder über bestimmte Aktivitäten benachrichtigen? Welche Funktionalitäten können das sein?
- Soll der Produkt Unterhaltungselemente beinhalten? Oder soll er designerisch "nüchtern" sein und nur eine Menge an Funktionalitäten ermöglichen?

Da wir in unserem Interview herausfinden wollen, ob Gamification-Elemente den Lernprozess stimulieren, sollen präzisere Fragen gestellt werden, die dem Verständnis helfen sollen, welcher Spielertyp die NutzerInnen sein könnten und welche Spielemente ausgewählt werden sollen. Die Fragen werden basierend auf den Taxonomien von Bartle (1996), Nacke et al. (2011) und Marczewski (2015) und den dementsprechend zugewiesenen Spielmechaniken entwickelt (Xu, 2011; Webb, 2013; Tondello et al., 2016; Schöbel et al., 2020).

Die Fragen werden basierend auf der Studie von Schneider et al. (2017) entwickelt, um die Spielertypen und gewünschte Unterhaltungselemente zu herauszufinden. Die Elemente werden in Interviews abgefragt, und je nach den 4 möglichen Spielertypen kategorisiert - Achievers, Killers, Explorers und Socializers, die die in-

terviewte Person sein kann. Es ist natürlich nicht ausgeschlossen, dass man nicht nur einem Spielertyp zugewiesen wird. Sonst wäre dies ein kritischer Punkt an der benutzten Bartle's Taxonomie (Barata et al., 2014, S. 18). Aus diesem Grund wurden andere Taxonomien entwickelt wie die Taxonomie von Marczewski (2015) oder das BrainHex Modell von Barata et al. (2014), welche die Überlappungen der Spielereigenschaften und Motivationsfaktoren zulassen.

Alle Taxonomien haben den Anspruch auf Gültigkeit. In unserer Studie wird davon ausgegangen, dass mehrere Spielertypen sich aus den drei analysierten Taxonomien miteinander nicht nur überlappen, sondern oft den gleichen Spielertyp haben, wie z.B. Socializers aus der Bartle's Taxonomie, dem BrainHex Modell und Marczewski's Taxonomie. Oder Achievers aus der Marczewski's, BrainHex und Bartle's Taxonomien. Oder Free Spirits aus der Marczewski's Taxonomie, Seeker aus dem BrainHex Modell und Explorer aus der Bartle's Taxonomie.

Somit konzentrieren wir uns in der Studie nicht darauf, dass nicht Spielertypen herausgefunden werden sollen, sondern dass die möglichen Gamification-Elemente, die nach den Spielertypen kategorisiert werden, nach ihrer Sinnhaftigkeit und den geäußerten Bedürfnissen in den Interviews abgefragt werden.

- Würden Sie Ihren Lernvorgang, Erfolg oder Unklarheiten beim Lernen mit KommilitonInnen, FreundInnen, ExpertInnen besprechen? Wie würden Sie das machen? Würden Sie miteinander telefonieren, miteinander chatten oder einander Ihre Fortschritte auf eine gewisse Weise mitteilen?
- Welche sozialen und kommunikativen Funktionalitäten würden Sie in einer Anwendung für das Python-Lernen bevorzugen?
- Wie finden Sie Belohnungen für das Lernen? Oder wie würden Sie ein System finden, dass Ihnen Abzeichen zuweist oder mit bestimmten Rewards, neuen Funktionalitäten, Orden, Auszeichnungen, Leistungsfähigkeiten für erledigte Levels, Aufgaben, Quizen belohnt? Welche Belohnungen finden Sie am sinnvollsten?
- Welche explorativen Erkundungsfunktionalitäten würden Sie gerne in der Anwendung treffen? Würden Sie sich eine Anwendung wünschen, die kein Ende im Lernprozess oder stundelange Exploration anbietet? Würden Sie gerne

nach Bugs suchen?

- Wie fühlen Sie sich in den Wettbewerben? Wie finden Sie die Funktionalitäten in der Anwendung, die eine Konkurrenz mit anderen voraussetzen? Unter welchen Bedingungen könnte bei Ihnen Aggressivität entstehen?
- Wie wollen Sie Ihren Lernfortschritt sehen oder kontrollieren können? Wie fühlen Sie sich, wenn Sie in ein neues Level oder neue Leistungen erreichen?

Bei der Entwicklung von Fragen wird berücksichtigt, ob die Befragten stark einem Erinnerungsprozess oder Beschreibung eigener Aktivitäten unterworfen werden (Goodwin, 2009, S. 118). Wenn dies der Fall ist, dann wird versucht, den self-reporting Fehler zu minimieren, in dem um Präzisierung gebeten wird, damit die erhaltenen Informationen dem tatsächlichen Verhalten entspricht.

Die entwickelten Interview-Fragen gehören dem semistrukturierten Interview, bei dem die Diskussion mit einem bestimmten Set an Fragen beginnt, aber immer noch die Möglichkeit besteht, das Gespräch in bestimmte Richtung zu steuern (Lazar, 2017, S. 198). Wenn die TeilnehmerInnen des Interviews noch keine Erfahrung mit anderen interaktiven Systemen für das Erlernen einer Programmiersprache haben, dann können die Fragen über den Mangel anderer Produkte vermieden werden. Wenn die Befragten auch irgendetwas von ihrem Interesse erwähnen, können mehr Fragen dazu gestellt werden, wie z.B. mehr Detaillierung.

Der Nachteil von unstrukturierten und strukturierten Interview-Arten ist, dass es mehr Herausforderungen bei der Interpretation der Antworten gibt (Lazar, 2017, S. 199).

2.1.2 Feature Requests

Für die erste weibliche interviewte Person, die Medieninformatik B.A. an der Uni Regensburg studiert und die sich schon mit den Programmiersprachen wie Java, JavaScript auskennt, ist es sehr wichtig, die Anwendung mit der Unterstützung der deutsche Sprache zu haben, damit der Lernstoff besser nachvollziebarer ist. Für die Person sind die Elemente, die sie beim Lernen loben oder besser orientieren, sehr wichtig. Solche Aufrufe wie z.B. "Gut gemacht!", "Sehr schön!" helfen ihr besser, den Fortschritt und den Erfolg der Aufgaben zu merken. Kritisch findet die Person

die Situationen, wenn die Punkte abgezogen werden, wenn die Aufgabe falsch gelöst wird. Und wenn es keine mehr Punkte übrig bleiben, dann kann nicht zu der nächsten Aufgabe gekommen werden. Somit sollen diese Verhinderungen wegen Punktabzug vermieden werden. Für die Person ist es wichtig, dass die Anwendung nicht unendlich verwendbar (somit keine Explorer-Gamification-Elemente besitzt (Bartle, 1996)) ist und sogar einen guten Überblick geben soll, wie lange welche thematische Kapitel dauern. Eine der zwei erwähnten Lernplattformen "Sololearn", die die erste befragte Person schon für das Lernen benutzt hat, wird in der Wettbewerbsanalyse berücksichtigt. Auf die Analyse der zweiten Lernplattform "Coding Bat" wird verzichtet.

Eine zusammenfassende Liste mit den wichtigen und sinnvollen Anmerkungen für den Lernprozess und für Design der Lernanwendung der ersten befragten Person liegt vor:

- Punkte, Sterne als Belohnungsmechanismus
- Unterstützung anderer Sprachen auf der Plattform
- Thematische Aufteilung
- Aufteilung nach Schwierigkeitsgrad. Aufsteigender Schwierigkeitsgrad
- Belohnungssystem, das nie enden oder erschöpft werden kann
- kurze und nachvollziehbare Bild- und Video-Elemente sollen beinhaltet werden
- Hilfe-Button, besonders bei Problemen auf der Anwendung
- Systemfeedback wie Ton, wenn z.B. die Aufgabe falsch gemacht wird
- Ein Level verhindert nicht das andere
- Ende soll sein
- Ansporn zum Lernen, Motivation sollen hervorgerufen werden. Vielleicht mit hilfe von Lobaufrufen oder Unterstützungswörtern
- Übersicht der Strukturen und Thematik und direkte Auswahlmöglichkeiten
- Kommunikationsmöglichkeiten, aber nicht mit fremden Leuten
- Angabe, wie lange ein Kapitel dauert

- Tastatur passt ganz gut für aufwändige Eingaben
- Präferenz für den Lernprozess alleine. Sonst Kontakt zu den anderen Personen mithilfe von Sprachaufnahmen oder Treffen an der Uni
- bunte Darstellung allgemein
- verschiedene Art der Aufgaben, wie z.B. Quizen oder Multy-Choice-Fragen
- Wiederholungsmöglichkeiten sollen umgesetzt werden

Die zweite Person vom weiblichen Geschlecht studiert Informatik B.S. an der OTH Regensburg und kennt schon einige Programmiersprachen wie Java und JavaScript. Sie versucht immer gerne den erlernten theoretischen Stoff gleich selbst auszuprobieren, indem sie selbst den Code schreibt oder den Code in der Entwicklungsumgebung ausprobiert. Sie findet benutzerfreundliche, fehlerlose Interfaces für den Lernprozess am besten. Solche Anwendungen werden ihr oft durch Professoren empfohlen. Auch wenn einige benutzte Anwendungen manchmal fehlerhaft waren, hatte es kaum Einfluss, wenn die Anwendung interessant war. Die Person benutzt für das Programmieren entweder Laptop oder Tablet. Wenn etwas schnell gegoogelt werden soll, benutzt sie gerne auch ihr Handy. Sie findet die Tastatur-Verfügbarkeit aber nicht kritisch und kann gerne ohne Tastatur mithilfe von anderen interaktiven Eingabemöglichkeiten umgehen. Sie lernt Programmiersprachen, indem sie nicht nur einfach Dokumentation durchliest, sondern auch kurze Videos, Diagramme, Schemas, Bilder dazu ansieht.

Eine Zusammenfassung aller erwünschten Features auf einer Lernanwendung von der zweiten Person ist hier aufgelistet:

- Hinweise, wie man es richtig machen kann
- Möglichkeiten der Rückkehr mit zurück-Button, Wiederholungen oder direkter Auswahl, wo man auf der Lernanwendung hin will
- Überblick über alle Möglichkeiten auf der Anwendung
- simple, nicht zu "trockene", aber mit guten Funktionalitäten, benutzerfreundliche, schnelle, intuitiv verständliche Interfaces

- gerne den Fortschritt sehen können. Der aktuelle Stand soll gespeichert werden können
- kurze Videos, Schemas, Diagrammen, Illustrationen, Bilder notwendig
- Malenmöglichkeiten, Möglichkeiten selbst ein Schema darzustellen
- Kommunikation lieber gleich zu dem Professor. Zuerst kann gegoogelt werden. Mit den Kommilitonen eher selten, nur in kritischen Situationen
- Chat und Forum wären aber gute Features
- gerne Belohnung beim Lernen
- Kontrolle, auf welchem Niveaus man sich befindet
- lernt lieber täglich ein bisschen, als sehr lange die Anwendung zu nutzen
- findet nicht normal, wenn man nur daran strebt, Punkte und neue Leistungen zu bekommen und zu erreichen
- gerne Ergebnis sehen können oder was noch vorkommt

Die dritte interviewte Person hat Medieninformatik studiert, wechselte aber danach zu einem anderen Fach. Sie hat Erfahrungen mit Java, JavaScript, HTML/CSS. Für das Lernen nutzt sie gerne Laptop und bevorzugt visuelle Wahrnehmungsarten. Den Lernprozess einer Programmiersprache stellt sie sich so vor, dass sie zuerst Grundlagen versteht und danach die fortgeschrittenen Übungen macht. Sie nimmt zuerst neue Informationen auf und wendet sie danach selbst auf ein Problem an. Sie hat noch keine Anwendungen für das Lernen der Programmiersprachen benutzt, nur einige Webplattformen für die Übungen der HTML/CSS-Erfahrung.

- theoretischer Teil, Aufgaben mit Ratschlägen, Lösungsvorschlägen und Korrekturen
- gerne offline-funktionierende Systeme
- Systeme, die wenig Speicherplatz nehmen
- Systeme, die die Aktivierung von anderen Dienstleistungen nicht brauchen - kein Bluetooth, keine Geolocation
- Systeme, die leicht bedienbar sind

- Systeme, die keine Erinnerungen schicken, es sei denn, die Notifications können ausgeschaltet werden
- kostenlose Anwendungen oder mit nicht teuren In-App Käufen
- Fortschritt soll gezeigt werden
- Übungen sollen wiederholbar sein, mit Erklärungen, in aufsteigendem Schwierigkeitsgrad
- Die Interaktion soll übersichtlich sein und strukturiert
- Fehler im Code sollen gleich angemerkt werden, z.B. syntaktische Fehler sollen unterriegt werden
- Bilder statt Videos. Kurz und knapp. oder vorhandene Zusammenfassung
- schlichtes und einfaches Design. Hauptziel-Funktionalitäten
- auch wenn Unterhaltungselemente nicht schlimm sind, kann darauf verzichtet werden
- Lernmöglichkeiten sollen auf der Anwendung ein Ende haben
- bevorzuglich allein-Lernprozess
- obwohl Forum-Möglichkeit soll auch vorhanden sein. Chat ist nicht nötig, weil immer online-Modus braucht und nicht alle zu gleichem Zeitpunkt online sein können
- Punkte sammeln ist unwichtig und nicht für den Lernprozess nötig
- Konkurrenz ist nicht schlecht. Es soll aber möglich sein, anonym zu bleiben
- Überblick, was richtig war und was falsch
- mobile und gleichzeitige Webanwendung wären erwünscht

Die vierte Interviewteilnehmerin, ist zur Zeit Masterandin im Fach Psychologie an der Uni Regensburg. Sie ist bereits geringfügig mit dem Thema Programmierung in Berührung gekommen, in dem sie einen vhb-Kurs für HTML und CSS besuchte. Jedoch brach sie ihn ab aufgrund des mangelnden Struktur des Kurses sowie fehlende Möglichkeiten, die Inhalte selbstständig zu implementieren. Ansonsten besitzt

sie keinerlei Programmierkenntnisse. Sie bevorzugt es visuell zu lernen und bevorzugt Laptops als ihre Lernplattform. Folgende Features wären Ihrer Ansicht nach wünschenswert:

- Hilfsmöglichkeiten, falls man bei einer Aufgabe stecken bleibt
- Verfügbarkeit von Zusatzmaterialien
- Verfügbarkeit von Skripten
- Feedback bei Fehlern
- Übungen sollen auf Transfer abzielen
- Erklärvideos (Länge soll sich einem bestimmten Rahmen befinden; ca. 5min.)
- Strukturierter Aufbau der Übungen und Kapitel
- Möglichkeiten gelerntes abzuprüfen/zu kontrollieren
- Möglichkeit, den Lernfortschritt zu sehen (via Level oder Fortschrittsbalken)
- Belohnungen begrüßenswert, jedoch nicht überzogen (im Sinne von Orden etc.)
- Input über Tastatur
- Forum, um mit anderen Lernenden kommunizieren zu können (Probleme, Fragenstellung etc.)
- Unterhaltungselemente sind in Ordnung, aber kein Muss

Der fünfte und letzte Teilnehmer unserer Interviews ist Student der Politikwissenschaft B.A. an der Universität Regensburg . Im Rahmen seines Studiums (Praxismoduls) absolvierte er einen R-Kurs zur Datenanalyse. Nebenbei brachte er sich selbst Grundkenntnisse in SPSS bei. Er ist am Erlernen einer Programmiersprache wie Python sehr interessiert, da er sich quantitative Fähigkeiten für die Methoden der Politikwissenschaft aneignen möchte und dies seiner Ansicht nach auch berufliche Vorteile mit sich bringt. Er bevorzugt es visuell und auditiv zu lernen sowie Lerninhalte ausführlich zu lesen. Der Teilnehmer präferiert Laptops als Eingabegegerät.

- Relevanz der vermittelten Inhalte sollte ersichtlich sein (Kontext) zum Zweck der Motivation
- Einführungsvideos
- Visualisierte Schritte
- Aufgabenstellung/Übung soll eine Mischung aus Theorie und Praxis darstellen
- Kompetenzlevel soll angegeben werden
- Möglichkeit zum Erlernen der reinen Syntax der Sprache
- Gamifizierung nicht notwendig

Nachdem alle interviewten TeilnehmerInnen analysiert wurden, wurden die wichtigsten Features gesammelt. Diese Features sind nach den Kriterien ausgewählt, dass sie unbedingt oder wünschenswerterweise in einer Anwendung vorhanden sein sollen.

Auch wenn es sich ergeben hat, dass die Mehrheit der Personen sich den Lernprozess eher mithilfe von Laptop und Tastatur vorstellen würden, Diese Beobachtung macht unsere Entscheidung klar, dass unsere Projektgruppe sich auf der Entwicklung einer mobilen Application konzentriert.

Allgemeine Liste der Anforderungen und Wünschen, die für unsere nächste Entwicklung berücksichtigt werden:

- Bilder, kurze Videos
- Kontrolle über Fortschritt, Progressleiste, via Level oder Fortschrittsbalken
- Möglichkeit der Wiederholungen
- ausschaltbare Benachrichtigungen, andere Dienste
- Struktur, Überblick, Übersicht
- Forum-Möglichkeiten
- Levels, Kapitel, Niveaus sollen unabhängig voneinander ausgewählt werden, somit wird strukturierter Aufbau der Übungen und Kapitel erwünscht

- Punkte, Sterne, Lebenschances sammeln und einlösen, ohne platte zu kommen, also unerschöpfbare Anzahl
- Mischung aus Theorie und Praxis
- Kompetenzlevel soll angegeben werden
- Belohnungen begrüßenswert, jedoch nicht zu übetrieben
- Ansporn, Lobaufufe
- Aufgabenstellungen sollen Relevanz herausstellen
- Unterhaltungselemente sind in Ordnung, aber nicht obligatorisch
- Gamifizierung nicht notwendig
- thematische Aufteilung
- Aufteilung nach Schwierigkeitsgrad
- Angabe, wie lange ein Kapitel oder ein Lernteil dauert
- andere Spracheinstellungen
- die Anwendung braucht nicht viel Speicherplatz

Die Ergebnisse aus den Interviews über das Unterhaltungsconcept mit Belohnungssystem, Kommunikations-, Explorations- und Achievementselementen fielen recht unterschiedlich aus. Während einige InterviewteilnehmerInnen der Meinung waren, dass die Funktionalitäten eher Vorrang hätten, waren Andere der Meinung, sie hätten gerne einen Unterhaltungssaspekt auf der Anwendung. Während einige interviewte Personen gerne mit den Punkten, Sternen belohnt wären, finden die anderen das eher unnötig.

Es hat sich ergeben, dass die Elemente Rangliste und Leaderboard, die den Killers und Achievers zugewiesen werden, keinen Zuspruch erhalten haben. Der Zuspruch auf die Aspekte der Kommunikation in Form von Foren mit Frage-/Antwortmöglichkeiten wurde bei der Mehrheit der interviewten Personen festgestellt.

Jeder stellt sich den Lernprozess als eine Kombination von Lernmaterial und Übungsaufgaben, als Kombination von Theorie und Praxis vor.

2.2 Wettbewerbsanalyse

Ohne Wettbewerb existiert kein Markt. Für mehrere Problemfragen wurden bereits Lösungen entwickelt. Mithilfe der Wettbewerbsanalyse kann jedoch vermutet werden, in wie weit ein neues Produkt innovativere und effektivere Leistungen anbieten könnte (Rogowski, 2016, S. 125).

Mithilfe dieser Wettbewerbsanalyse können spezifische Felder und Faktoren identifiziert werden (Kolano, 2014, S. 37), die in dem neuen Produkt bei der Entwicklung berücksichtigt werden sollen.

Bei der Unterscheidung von Konkurrentarten werden direkte und indirekte Wettbewerber abgegrenzt. Direkte Wettbewerber bieten die Produkte und Leistungen der gleichen Klasse an und werden von den Unternehmen entwickelt und hergestellt (Rogowski, 2016, S. 127).

Die Produkte und Leistungen von den indirekten Wettbewerbern werden allgemein bei der Problemlösung berücksichtigt. Die hergestellten Produkte und angebotenen Dienstleistungen von indirekten Konkurrenten bieten Güter gleicher Funktion wie direkte Wettbewerber, aber aus einer anderen Klasse (Rogowski, 2016, S. 127).

In unserem Fall sind die indirekten Wettbewerber die Institute, Hochschulen und Programmierkurse, die zwar die gleiche Funktion tragen, indem sie Kenntnisse vermitteln, aber anderen Klassen angehören. Der Fokus liegt in unserem Projekt auf den direkten Wettbewerbern, die die Güter der gleichen Klasse anbieten und die offensichtlichen und wesentlichen Konkurrenten sind. Da die Entwickler oft nicht nur direkte Konkurrenten analysieren, entsteht somit die Gefahr, dass die Entwickler aus einer eingeengten Perspektive den Blick auf den Markt haben (Rogowski, 2016, S. 127). In unserem Fall wird noch betont, dass nur direkte Wettbewerber berücksichtigt werden.

Laut Rogowski (2016) sollen aus dem nutzerzentrierten Designansatz auch unbedingt die Befragungen und Gespräche mit den potenziellen NutzerInnen durchgeführt werden, die über ihre Bedürfnisse und Probleme klar sprechen, was für die Marktanalyse miteinbezogen werden soll. In unserem Fall sind das die Interviews, die vorher durchgeführt und analysiert wurden.

Zwischen den potenziellen Wettbewerbern sind die Unternehmen in einem anderen geographischen Gebiet; Unternehmen, die schon ähnliche Güter erfolgreich anbieten und ihre Marktdominanz erweitern wollen; Forschungs- entwicklungsstarke Konkurrenten mit ihrer konkurrierenden Technologie, wie z.B. online-Programmierkurse; die anderen, die eine eine zusätzliche Wert- schöpfungsstufe erreichen wollen (Rogowski, 2016, S. 131).

In unserem Projekt werden die ersten zwei Typen der Konkurrenten analysiert, indem AppStore und GoogleStore nach den möglichen mobilen Anbietern ähnlicher Apps analysiert werden. Es werden Produkte analysiert, die unabhängig von der Geographie, aber in englischer Sprache angeboten werden. Obwohl potenzielle Wettbewerber mit verschiedenen Methoden analysiert werden können, wie z.B. Tests der Produkte, Unternehmenbesichtigung, Interviews, Forschungskooperationen, Datenaustausch, Beraterhilfe, Publikationen, Datenbanken für die zusätzliche Information und "Horizont Scanning", bleiben die Beobachtungen und Suche nach den Konkurrenten ziemlich schwierig, weil die Technologien geheim gehalten werden, die Daten intransparent oder Preise der Produkte selbst für das Testen hoch sind (Rogowski, 2016, S. 131-132).

Die Bewertung der Wettbewerber kann auf verschiedene Arten und Weise geschehen. Es können Stärken und Schwächen der Konkurrenten im Vergleich zum eigenen Unternehmen in Form einer Matrix analysiert werden. Oder eine tiefere Analyse für die geplanten Zielkategorien unseres eigenen Produktes kann in der Art der Matrix von Leistungsmerkmalen durchgeführt werden (Rogowski, 2016, S. 132-134).

In der Wettbewerbsanalyse können die Nutzwerte auch gewichtet und benotet werden (Hering, 2014, S. 15-18).

In unserem Projekt werden die notwendigsten Funktionalitäten analysiert. Dazu werden die Ergebnisse aus den Interviews herangezogen werden, weil alle befragten Personen ihre Wünsche und Anmerkungen uns geteilt haben, was als schlechtes Design gilt und was sie sich aus ihren Erfahrungen vorstellen würden. Außerdem werden zusätzlich Möglichkeiten der unterschiedlichen Anwendungen auch berücksichtigt, um herauszufinden, ob sie die Anwendung attraktiver und effektiver für die Nutzung machen.

Die Funktionalitäten, die die Anwendungen anbieten, werden mit einem Plus-Zeichen abgehackt. Wenn die Funktionalität nicht implementiert ist, dann entspricht das einem Minus-Zeichen.

Da eine befragte Person angegeben hat, dass ihr wichtig ist, zu merken, wie lange ein Kapitel dauert, wird diese Funktionalität auch in den Konkurrenten analysiert. Wenn die Struktur der information auf der Anwendung keine thematische Aufteilung auf Kapitel voraussetzt, wird das nicht als Kapitel angenommen und steht somit ein Minus-Zeichen in dem Feld "Dauerangabe eines Kapitels, als Zeit oder Progressleiste". Einige Applications wie z.B. "Learn Python iOS" haben keine bestimmte thematische Aufteilungsstruktur auf der Anwendung. Die Videos, die aber zur Verfügung stehen, zählen nicht zu der thematische Aufteilung, nur wenn die Videos nicht in bestimmten Kapiteln stehen.

Da auch einige Personen aus der Fokusgruppe die Lernplattform "Udemy" diskutiert haben, die ihnen sehr gut geholfen hat, andere Programmiersprachen zu lernen, wird "Udemy"-Webanwendung auch in der Analyse der Wettbewerber berücksichtigt.

In der Tabelle werden alle vorhandenen Funktionalitäten aufgelistet, die die potenziellen Konkurrenten charakterisieren.

Anforderungserhebung

Kategorie	"Mimo iOS"	"Learn Python iOS"	"Python for beginners iOS"	"Python recipes iOS"	"Codecademy GO", Web	"Solo learn iOS", Web	"Udemy", Web
Profil erstellen	+	+	-	-	-	+	-
Code anschauen	+	+	+	+	+	+	+
Code schreiben	+	-	-	-	-	+	+
Community, Team erstellen, anschließen	+	+	-	-	-	+	+
Thematische Aufteilung	+	-	-	+	+	+	+
Theoretisches Material	+	+	+	+	+	+	+
Übungsaufgaben, Quizzen	+	-	-	+	+	-	+
Übungsaufgaben, Quizzen für Punkte, Zeit oder Lebenschancen	+	+	+	-	+	+	+
Anspornaufrufe	-	-	+	-	-	+	+
Möglichkeiten für fortgeschrittene ProgrammierInnen oder (für extra Kauf)	-	-	+	-	+	+	+
Dauerangabe eines Kapitels, als Zeit oder Progressleiste	+	-	-	-	+	-	+
Activity-Kalender	+	-	-	-	+	+	-
Job-Suche	-	+	-	-	-	-	-
Interview-Fragen	-	+	-	-	-	-	+
Bookmarks	-	+	-	-	-	-	+

Kategorie	"Mimo iOS"	"Learn Python for beginners iOS"	"Python Recipes in iOS"	"Python GO", Web	"Codecademy iOS", Web"	"Solo learn iOS", Web"	"Udemy", Web"
Video Tutorials	-	+	+	+	+	+	+
Leaderboard, Ranglisten	+	+	-	-	+	+	-
Blog, Forum, Fragen/- Antworten	-	+	+	+	+	+	-
Erinnerungen, Notifications	-	-	-	-	+	+	+
Unterstützung in verschiedenen Sprachen funktioniert offline	-	-	-	-	-	+	+
	+	-	+	+	-	+	+

Mehrere Anwendungen haben Gamification-Elemente umgesetzt (Xu, 2011), wie z.B. Leaderboard, Quizen, Punktesammlung, mit Lebenschancen die Übungsaufgaben lösen zu können, Community-, Team-Erstellung, Fragen-, Diskussionsaus tausch.

Die Gamification-Elemente wie Leaderboard werden positiv zum "anspornen" des Lernprozesses bewertet. Selbst die Gestaltung in der englischen Sprache wird auch als sehr nützlich eingeschätzt (siehe Abbildung 1).

"Learn Python iOS" bietet eine umfangreiche Menge an Funktionalitäten an - von den ersten Programmierschritten, die mit Videos und umfangreichen Übungsaufgaben vermittelt werden, bis zu den fortgeschrittenen Themen und sogar Möglichkeiten auf dem Markt ein entsprechendes Job zu finden und mögliche Interviews-Fragen anzuschauen. Selbst die Beschreibung von den EntwicklerInnen klingt viel versprechend, ohne Zusatzkosten abzufragen (siehe Abbildung 2).

Über die Anwendung "Python for beginners" wird in den Bewertungen angegeben, dass die Anwendung über einige Bugs verfügt, welche aber bei unserer eige-

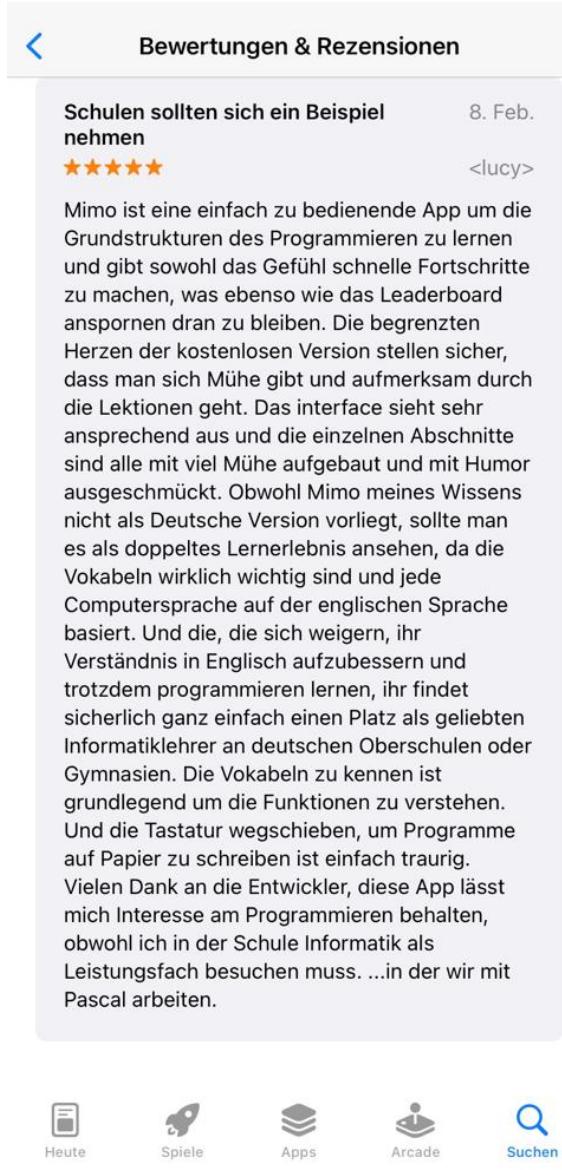


Abbildung 1: Screenshot der Bewertung von "Mimo iOS"-Anwendung

nen Analyse nicht festgestellt wurde. In der App sind grundlegende Elemente vorhanden wie z.B. theoretische Lerninhalte, Übungsaufgaben und ausreichende gute Code-Beispiele (siehe Abbildung 4). Das Design ist ziemlich nüchtern, verfügt aber über gängige Algorithmusbeispiele, die bei den ProgrammieranfängerInnen eine gute Übung sein sollen.

Die Anwendung "Python Recipes" 4.8 Sterne Bewertung und wird nicht nur als mobile Anwendung, sondern auch als Webanwendung angeboten, was mithilfe von responsivem Layout erreicht wird.

WsCube Tech presents you a step by step guide to Python Programming, Python is a general-purpose, versatile and popular programming language. It's great as a first language because it is concise and easy to read, and it is also a good language to have in any programmer's stack as it can be used for everything from web development to software development and Data Science.

Browse through our modules for:- Python programming, Python quiz test, Python interview Question, Python technical terms and many more with new content being added regularly.

Below are the topics provided in the app:-

Video tutorials:- This app provides both theoretical and practical videos on topics related to Python. The videos provide an interactive way to understand the topic and its concept.

Interview Questions:- This section covers the top Python programming interview questions along with answers that will help you crack the interviews and secure the job.

Quiz Tests:- Test your Python programming knowledge with quiz test. Challenge yourself and share Python knowledge with your network.

Challenges :- challenge your friends and bring out the best of you in the competitive quiz arena.

The topics covered in this app are:

-Introduction of Python

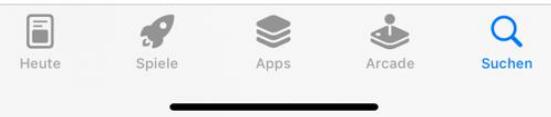


Abbildung 2: Beschreibung der App "Learn Python iOS" in AppStore

"Codecademy" zählt zu den beliebtesten Plattformen zum Erlernen von Programmieren beziehungsweise einer Programmiersprache. "Codecademy" bietet hierbei zwei Möglichkeiten um Programmieren zu lernen: "Codecademy" als Webapplikation und "Codecademy GO" als Apps für Android und iOS. Aufgrund der wenigen Übungen, die auf der Web Version kostenlos absolviert werden können, konzentriert sich die Analyse auf die mobile GO Version. Diese deckt weitestgehend alle Konzepte von Python ab, einschließlich Objekt-orientierter Programmierung. Mit dem Kauf der Pro-Version werden zusätzlich noch die Kapitel "Code Challenges" und "Next Steps" freigeschaltet.

Um die App nutzen zu können, werden die NutzerInnen aufgefordert, einen Codecademy-

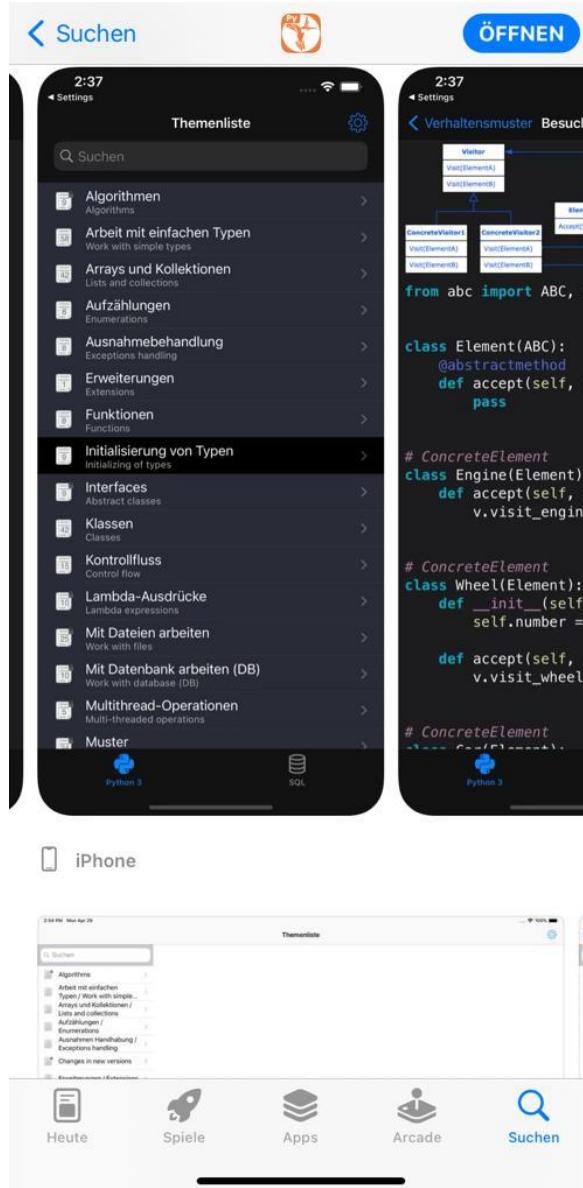


Abbildung 3: Beschreibung der App "Python Recipes iOS, MacOS" in AppStore

Account zu eröffnen. Hierbei fällt auf, dass lediglich zwei Optionen zur Registrierung bestehen: entweder wird der Account über einen bestehenden Google Account erstellt, oder über einen bestehenden Facebook Account. Dies fällt in der Analyse mehrfach negativ ins Gewicht: zum einen kann es durchaus der Fall sein, dass die NutzerInnen keinen Account der genannten Plattformen besitzen oder besitzen möchten. Zum anderen besteht keine Möglichkeit, wie bei anderen Apps (welche Funktionen und Ziele dabei sie verfolgt, ist in dem Falle unerheblich) sich über eine E-Mail Adresse zu registrieren.

Nach Abschluss der Registrierung, wird den NutzerInnen angeboten, Notifac-

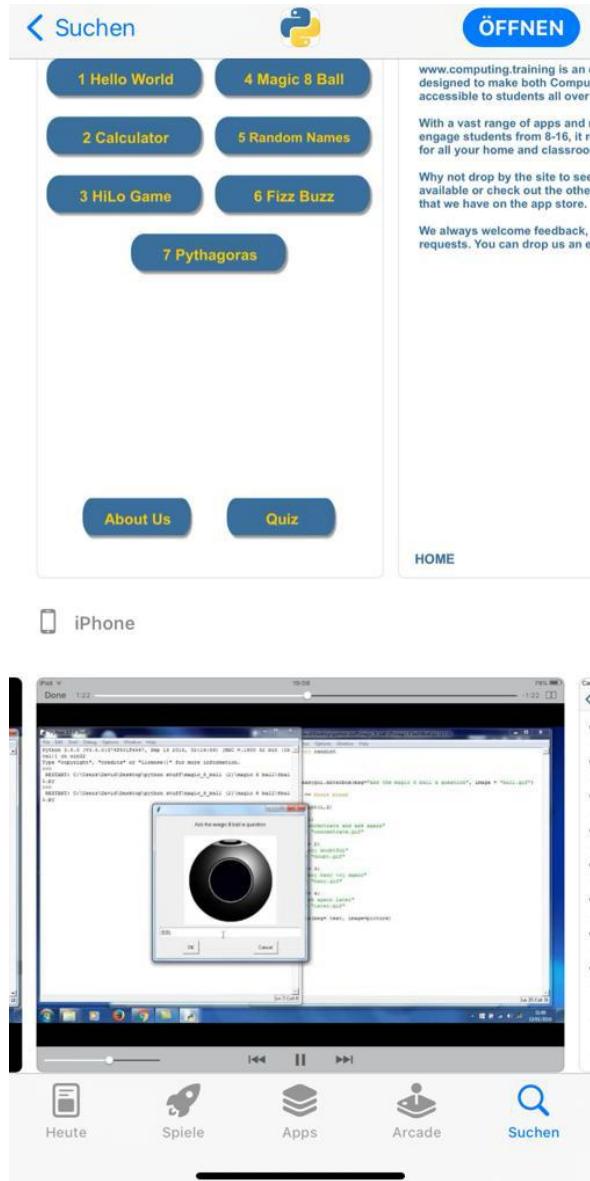


Abbildung 4: Beschreibung der App "Python for beginners iOS" in AppStore

tions zu aktivieren, um die NutzerInnen an ihre (täglichen) Trainingssessions zu erinnern. Es bestehen die Optionen, diese nicht zu aktivieren, sowie die Option "Occasionally", also sporadische Erinnerungen (wie oft diese vorkommen, wird nicht erwähnt), sowie tägliche Notifications.

Anschließend werden den NutzerInnen Kurse zu verschiedensten Programmiersprachen wie JavaScript, Java angeboten, sowie Kurse zu HTML und CSS. Da Python Gegenstand unseres Projektes ist, wird auf den angebotenen Python (Grund-)Kurs eingegangen. Zusätzlich bietet die App einen Vertiefungskurs in Python an. "Analyze Data with Python" behandelt Python Bibliotheken die für die Datenana-

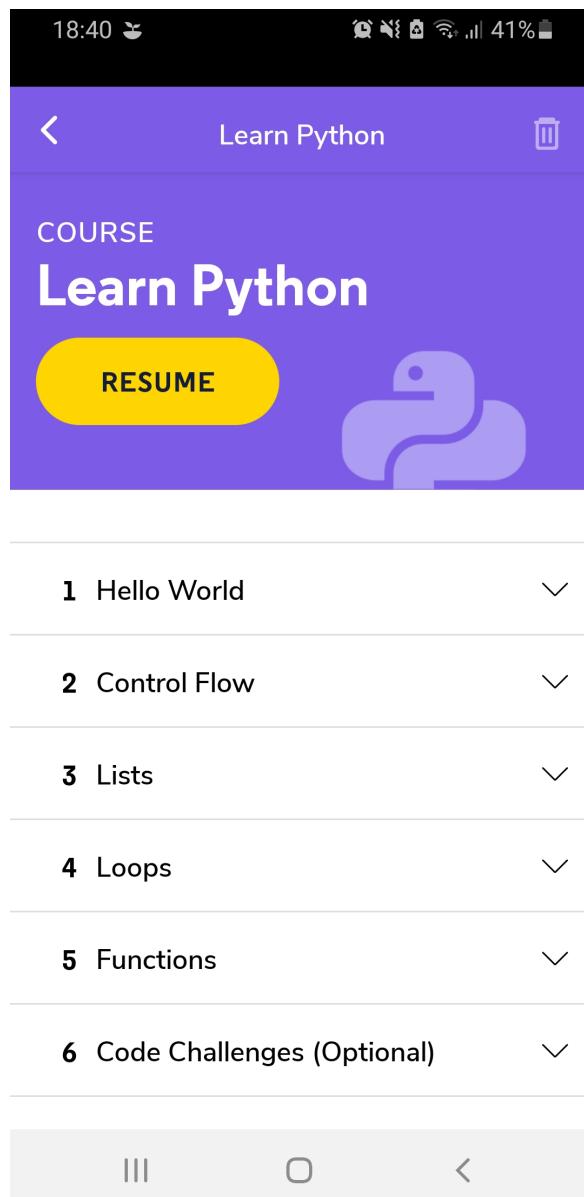


Abbildung 5: Startseite der App "Codecademy GO"

lyse und -visualisierung verwendet werden.

Die App bietet außerhalb der Lerneinheiten, Videos, darunter Tipps, die indirekt beim Erlernen einer Programmiersprache sinnvoll sein können. Beispielsweise verweist die Apps auf Videos, die den Lernenden Tipps geben, wie man damit umgehen soll, falls man bei einem Programmierproblem (oder übung) nicht weiter kommt. Zudem besitzt sie eine Suchleiste, in der nach bestimmten Konzepten gesucht werden kann.

Die Übungen bestehen aus einem Übungskapitel, welches ein Kernkonzept der Programmiersprache behandelt, wie z.B. "Lists" und "Loops". Das Kapitel selbst

unterteilt sich in "Review", in welcher das Konzept noch mal an Beispielen aufbereitet und erklärt wird. Im Unterkapitel "Practice" werden die entsprechenden Konzepte abgefragt bzw. geübt. Die Übungen bestehen aus "Multiple Choice" Fragen, sowie Lücken, die mit den richtigen Code "Snippets" gefüllt werden, um die Aufgabe korrekt zu lösen. Bei den Aufgaben handelt es jedoch nicht nur um einfache Fragen zur Syntax, sondern auch um logische Problemstellungen und das Verständnis von grundlegenden Funktionen (z.B. `.sort()` zur Sortierung einer Liste).

In den Übungen ist es möglich, noch mal in eine Übersicht des abgefragten Konzeptes zu wechseln. Sollte eine Übung nicht korrekt beantwortet werden, wechselt die App zur nächsten über. Die falsch beantworteten Fragen werden zu einem späteren Zeitpunkt in der Trainingssession abermals gestellt.

Nach Beendigung der "Practice"-Session erhält der Nutzer eine Übersicht der richtigen und falsch gelösten Übungen. Dies erweist sich als sinnvoll, um den NutzerInnen seine Schwächen zu beheben. Bezuglich der Motivation und den "Gamification"-Elementen werden sogenannte "Streaks" angezeigt. Bei den Streaks handelt es sich um eine Serie von täglich absolvierten Übungen. Ein Streak hält solange an, sofern man tägliche Übungseinheiten nach geht. Eine Timeline zeigt den NutzerInnen die bisherigen Streaks an, sowie ein Kalender, in der alle gemachten Übungseinheiten angezeigt werden. Neben diesen zwei Aspekten wird auch ein Leaderboard angezeigt. Dieser beschränkt sich jedoch nur auf den besten Streak, den derzeitigen Streak sowie die Gesamtzahl der Tage, an denen geübt wurde.

Die Trainingseinheiten zielen auf 10-minütige Übungen pro Tag ab. Diese bestehen aus "Review" und "Practice" und sollen jeweils 5 Minuten dauern. Ziel ist es, das Erlernen von Python (oder jeder anderen angebotenen Sprache) in möglichst kleine Lerneinheiten zu unterteilen, um die AnfängerInnen nicht allzu sehr zu überfordern.

Die Anwendung "Codeacademy GO" erlaubt eine gezielte thematische Kapitelauswahl, ohne dass davor andere Levels oder Leistungen erbracht werden sollen. Das kann positiv bewertet werden, denn aus den Interviews ergeben sich Wünsche nach einem Freiheitsgrad, bei dem man sich für ein bestimmtes Thema entscheiden will, ohne bestimmte Voraussetzungen.

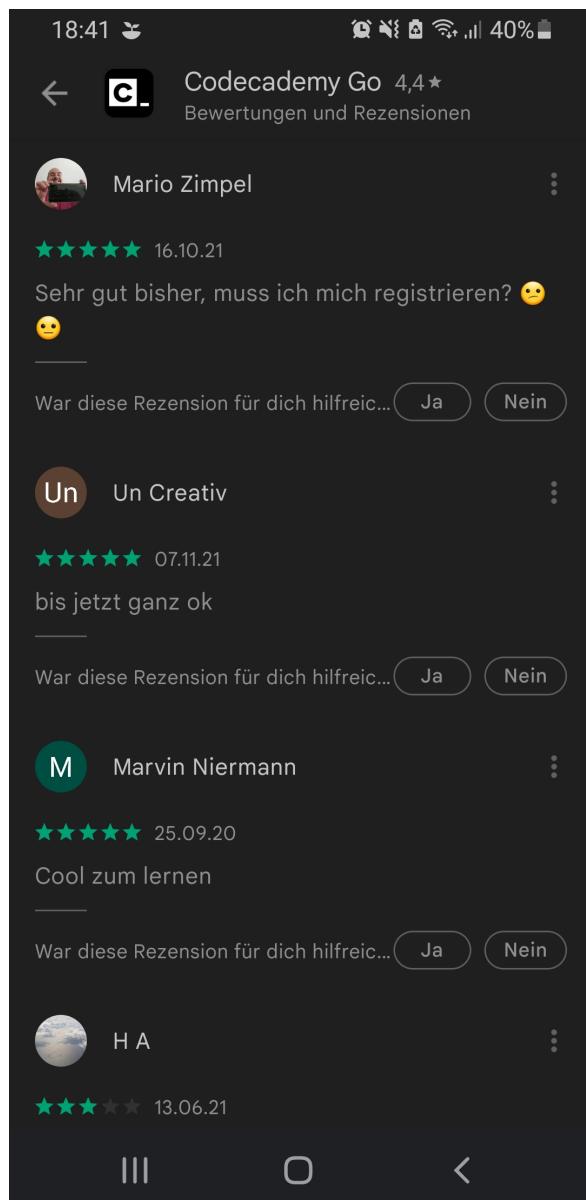


Abbildung 6: Rezensionen der App "Codecademy GO"

Die Bewertungen sind mit 4,4 Sternen (Stand: 02.02.2022) sehr gut. Die User bewerten die Inhalte der App als recht gut, jedoch äußerten einige User, dass sie die Desktop bzw. Web Version bevorzugen würden, da diese mehr Inhalte und Funktionalität anbietet. Insbesondere bei NutzerInnen ohne jegliche Programmiererfahrung stößt die Applikation auf positive Resonanz.

Die mobile Anwendung "SoloLearn iOS" wird in Apple Store hoch bewertet, mit einem Durchschnittswert 4.8 aus 5. Die Anwendung verfügt über die personalisierte Anpassung an die NutzerInnen, indem am Anfang Informationen über Kenntnisstand und Zweck gesammelt werden. Die Anwendung bietet die Möglichkeit an,

unterschiedliche Programmiersprachen auf verschiedenen Schwierigkeitsniveaus zu üben. Die Sprache, mit welcher man auf der Plattform aktiv ist, kann auch ausgewählt werden oder automatisch durch Geolocation angepasst werden.

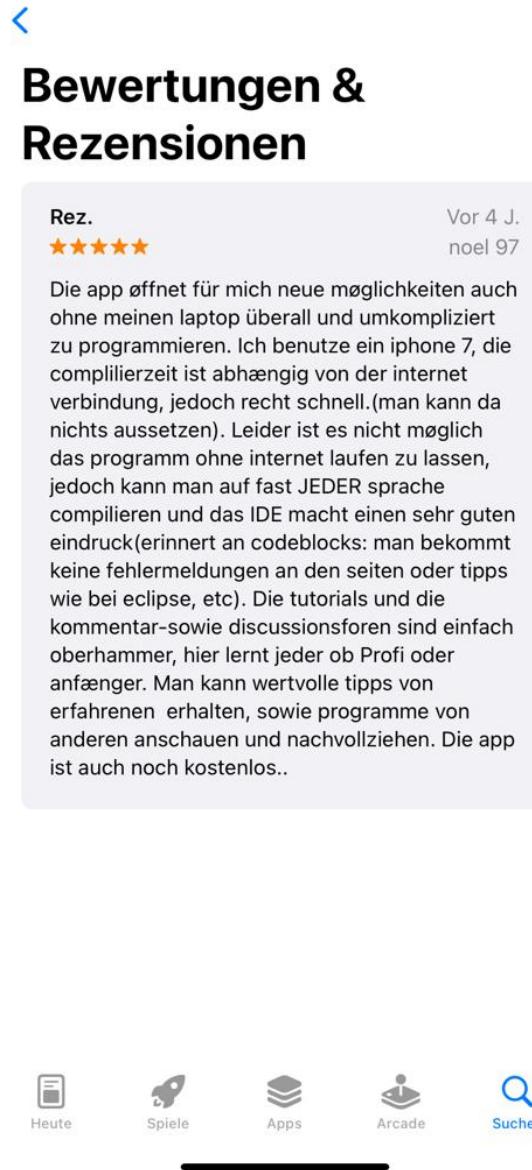


Abbildung 7: Screenshot der Bewertung von "SoloLearn iOS"

Alle analysierten Wettbewerbsanwendungen verfügen über die Möglichkeiten, die Codebeispiele als kleine Codeausschnitte oder als ganze Programmcodes anzuschauen, was gut für die ProgrammieranfängerInnen sein soll, weil sie sich an die Syntax und Semantik der Python-Sprache gewöhnen sollen. Wahrscheinlich sind auch alle AnwendungsentwicklerInnen davon ausgegangen, dass selbständiges Codeschreiben nicht nur kompliziert in der Anwendung umzusetzen ist, sondern auch

eher dem selbstständigen Lernen ausserhalb des Lernprozesses in der Anwendung gewidmet werden soll. Somit Code anschauen zu können ist ein relevanter Teil des Lernens in der Anwendung, andersrum ist selbstständige Codegenerierung eher optional und wird ausserhalb des Rahmens der Anwendung rausgenommen.

Vermittlung des theoretischen Lernstoffs ist auch ein relevanter Teil der Anwendung und somit des Lernens. Übungsaufgaben, ob mit Gamification-Elementen wie z.B. Quizen oder Punktesammlung, oder ohne, sollen den Lernstoff begleiten und praxisorientierte Möglichkeiten anbieten. Somit sind theoretisches Material und Übungsaufgaben nicht nur sinnvoll, sondern unmittelbar notwendig in der Lernanwendung. Ob diese Teile mit Gamification-Elementen höhere Motivation versprechen, wird genau in unserem Projekt herausgefunden.

Einige Applications setzen Gamification-Elemente um, wie beispielsweise Leaderboard, Team-Erstellung. Leaderboard-Implementierung, Fragen/Antworten, Blog/-Forum und Community-Erstellung setzen voraus, dass die NutzerInnen ein Profil erstellen sollen, über welches sie aktiv in der Lernanwendung kommunizieren.

Thematische Aufteilung vom Lernmaterial soll den ProgrammieranfängerInnen einen strukturierten Lernprozess ermöglichen, was auch von der Mehrheit der Anwendungen umgesetzt wird.

Bookmarks von beliebigen Kategorien in der Anwendung "Learn Python iOS" ist eine gute Funktionalität, die den NutzerInnen die Möglichkeit liefert, bestimmte wichtige Informationen für sich selbst lokal zu aufbewahren. Außerdem ist die Bookmarks-Kategorie selbst nach den Kategorien gruppiert und nach dem Datum des Hinzufügens sortiert. Somit können die NutzerInnen einen schnellen Zugriff zu ihren persönlichen Informationen haben. Diese Feature ist zwar wünschenswert, aber sehr hilfreich.

Video-Begleitung und Tutorials werden von meisten Anwendungen umgesetzt, was auch den vielfältigen Lernprozess fördert und gute visualisierte Unterstützung ist.

Einige Funktionalitäten oder designerische Anregungen, die interviewten Personen sich gewünscht haben - offline-Modus, Unterstützung in verschiedenen Sprachen, Anspornaufrufe - sind nicht in allen analysierten Wettbewerbern vorhanden.

Diese Funktionalitäten wären aber nötig, weil sie dann die Anwendung flexibel und personalisiert machen können. Gar nur eine Anwendung ermöglicht Einstellung auf eine andere Sprache, was aber möglich wäre, auch wenn dafür keine Geo-Location-Einstellung nötig wäre.

Spezifische Funktionalitäten wie z.B. Job-Suche, Interviews-Fragen, Activity-Kalender sind nur in einer Anwendung umgesetzt, was wahrscheinlich von anderen Konkurrenten gering geschätzt wird. Obwohl "Mimo"-Anwendung aus dem deutschen AppStore Job-Suche-Funktionalität anbietet, funktioniert sie aber nicht. Wahrscheinlich ist diese Feature von der Geoposition abhängig und dadurch mit nicht erlaubtem Geo-Tracking für Deutschland nicht zur Verfügung steht. Diese Funktionalitäten sind eher optional und dienen nicht unmittelbar einem Lernprozess für ProgrammieranfängerInnen.

Extra Einkäufe in einigen Anwendungen sind zwar möglich, beschreiben aber nicht, was für extra Zahlung angeboten wird. Dadurch machen sie einen nicht vertrauten Eindruck.

Somit ergibt sich eine Liste an wichtigsten Funktionalitäten:

- Codepraxis
 - Code anschauen
 - Code schreiben
- Thematische Aufteilung, Kapitel mit der Übersicht
- Theoretisches Material
- Audiovisuelle Untestützung
 - Bilder
 - Videos
 - Schemas
 - Diagramme
- Übungsaufgaben, Quizen
- Übungsaufgaben, Quizen für Punkte, Zeit oder Lebenschancen
- Anspornaufrufe

- Dauerangabe eines Kapitels, als Zeit oder Progressleiste
- Unterstützung in verschiedenen Sprachen
- funktioniert offline
- Leaderboard, Ranglisten
- Blog, Forum, Fragen/Antworten

Somit wird klar, dass Unternehmen seine Wettbewerber, ihre Potenziale und dadurch Potenziale eigener Güter erkennen und analysieren sollen, was am besten regelmässig und umfangreich durchgeführt wird. Die Stärken und Schwächen der Konkurrenten einzuschätzen erbringt gute Erkenntnisse über das, was die anderen Unternehmen schon anbieten und was noch an eigenem Produkt besser gestaltet werden kann (Kugler & von Janda-Eble, 2018, S. 42-43).

Auch Udemy erfreut sich größter Beliebtheit unter jenen Plattformen, welche Programmierkurse anbieten. Die Plattform unterscheidet sich deutlich von den bisher analysierten Konkurrenten in unserer Wettbewerbsanalyse denn: Udemy ist keine Webanwendung die dezidiert zum Erlernen einer oder mehrerer Programmiersprachen ist, sondern es handelt sich um eine Plattform, in der alle möglichen Kurse zu den unterschiedlichsten Themen wie z.B. Fotographie, Cyber-Security bis zu hin Kursen über Persönlichkeitsentwicklung. Diese Unterscheidung ist wichtig, da unzählige Kurse zu ein und dem selben Thema existieren und diese unterschiedlich strukturiert sein können. Dies betrifft auch das Erlernen der Programmiersprache für Python, insbesondere für Anfänger. Dennoch kann festgestellt werden, dass jeder Kurs grundlegende Prinzipien hinsichtlich der Struktur und Aufbau enthält, lediglich in der Aufgabenstellung der Übungen und dessen Erledigung bestehen geringfügige Unterschiede zwischen den angebotenen Kursen. Die Kurse sind kostenpflichtig und bewegen sich in einem Preisrahmen von ca. 200 Euro aufwärts. Jedoch bietet Udemy regelmäßig Kurse zu stark reduzierten Preisen an, sodass diese nur noch 12 bis 15 Euro kosten. Es ist davon auszugehen, dass die meisten Nutzer:innen ihre Kurse zu diesen Preisen erworben haben. Nach Angaben von Udemy bleiben die gekauften Kurse dauerhaft bestehen. Udemy zeichnet sich vor allem durch seinen Fokus auf Videos die zur Erklärung der jeweiligen Inhalte herangezo-

gen werden. In der Regel sind diese in Kapitel unterteilt, welche sich in weitere Unterkapitel aufteilen. Um die jeweiligen Kapitel und Unterkapitel abzuschließen, gilt es die Videos vollständig anzusehen. In einigen Fällen werden kleine, recht einfache Übungsaufgaben gestellt. Diese befinden sich einem von Udemy zu Verfügung gestellten Code-Editor. Dieser überprüft die eingegeben Lösung auf ihre Richtigkeit. Die Videos zeigen den Tutor beim programmieren von Codebeispielen. Dabei ist es dem Nutzer freigestellt, ob dieser sich die Inhalte nur ansieht oder diese auch nach programmiert - im Sinne eines "Code-alongs" - und mit dem Code gegebenenfalls experimentiert und ihn erweitert. Unterhalb der Video können Fragen an den Tutor bei Unklarheiten gestellt werden. Zudem besteht die Möglichkeit, Notizen unter dem Video zu machen, welche dann in der Zeitachse des jeweiligen Videos angezeigt werden. Auch die Notizen anderer Nutzer werden angezeigt. Nach Ende eines Kapitels werden Codeaufgaben gestellt, die der Nutzer zu lösen hat. Die Codes werden allerdings nicht im Udemy Code-Editor zur Verfügung gestellt. In der Regel verwenden Kursanbieter "Anaconda" als IDE und stellen dementsprechend die Aufgaben und dessen Lösungen als Notebook zur Verfügung. Im Verlauf des Kurses und fortschreiten der Kenntnisse des Users, werden am Ende von Kapiteln auch Projekte als Aufgaben gestellt. Diese sind so konzipiert, dass die erklärten Inhalte eines Kapitels in einem großen Projekt ("Capstone-Projekt") Anwendung finden. Die Nutzer:innen werden jedoch nicht angehalten, strikt dem Aufbau des Kurses zu folgen. Sie können nach Belieben einzelne Kapitel bearbeiten beziehungsweise weglassen. Nach erfolgreichen Abschluss des Kurses erhalten die Teilnehmer eines Urkunde beziehungsweise ein Zertifikat, dass u.a. auf LinkedIn zum Profil hinzugefügt werden kann. Mit 3,9 Sternen im Google PlayStore wird die App von Udemy als "Gut" bewertet. Die negativen Rezensionen beziehen sich lediglich auf Bugs, Störungen etc. Inhaltlich haben die Nutzer an Udemy nichts zu beanstanden. Gelegentlich wird allerdings die Desktop-Version als bevorzugte Plattform erwähnt.

Die durchgeführten Interviews und die dadurch entstandenen und analysierten Ergebnisse sind für die Wettbewerbsanalyse herangezogen Rogowski (2016), um die für potenzielle NutzerInnen relevantesten Funktionalitäten zu beachten und für die Entwicklung unseres Produktes zu berücksichtigen.

2.3 Fokusgruppe

Vorteile der Fokusgruppe im Vergleich zu Interviews sind zeitlicher Gewinn und Vielfalt an Ansichten und Meinungen (Lazar, 2017, S. 204). Die begrenzte Diskussionsfreiheit sowie semistrukturierte oder unstrukturierte Interviews-Fragen vermitteln vielfältige Standpunkte, ermöglichen die Inspiration und Dynamik im Gespräch. Dies ist aber nicht ohne Nachteile, dass einige aktive Sprecher aktiver sind und dadurch die andere Gesprächsteilnehmer zur Zurückhaltung verleiten (Lazar, 2017, S. 205). Solche Fälle sollen höflich umgegangen werden oder es soll Ansatz der individuellen Interviews ausgewählt werden, wenn die gruppendifferenten Faktoren nicht vorhanden sind (Goodwin, 2009, S. 56).

Fokusgruppen geben viel Information am Beginn des Projektes, in wie weit das Produkt realisierbar und rentabel sein kann (Goodwin, 2009, S. 56). Fokusgruppen sind nicht hilfreich, wenn es um die Nutzung des Produktes geht, sondern effektiv bei der Sammlung von Designideen und ästhetischen Aspekten (Goodwin, 2009, S. 56). Obwohl immer zu beachten ist, dass in den Fokusgruppen die Gefahr liegt, keine Unterschiede gefunden zu haben, sondern Konsensus. Außerdem können sich Leute nicht so wohl in einer Gruppe fühlen oder in einer ungewöhnlichen Umgebung, was auch die Ergebnisse beeinträchtigen kann (Goodwin, 2009, S. 56).

Bei allen qualitativen Methoden gibt es immer die Gefahr eines self-reporting Fehlers, wenn die Befragten falsche Information geben, aber nicht aus der Unehrlichkeit, sondern weil die Selbstwahrnehmung sich von dem aktuellen Verhalten und Aktionen unterscheiden (Goodwin, 2009, S. 55). Individuelle Interviews und Fokusgruppen haben hohes Risiko des self-reporting Fehlers. Dieser Fehler kann gut durch Beobachtungen in bestimmten Situationen minimiert werden (Goodwin, 2009, S. 57).

Die Fragen für Fokusgruppe sollen möglichst auf Fragen verzichten, welche Phrasen wie "was gefällt Ihnen" beinhalten (Goodwin, 2009, S. 650). Es soll eher auf die Fragen der Erfahrungen eingegangen werden, damit gute Information über praxisnahe Erlebnisse und existierende Erfahrungen gesammelt werden kann.

Da in unserer Studie auch davon ausgegangen wird, dass einige Befragten auch noch keine Erfahrungen mit den Anwendungen für das Lernen einer Programmier-

sprache haben können, werden die Fragen "was gefällt Ihnen" trotzdem gestellt.

In unserer Fokusgruppe wurde immer auf die Bilanz und Ausgleich der Rollen im Gespräch geachtet, damit alle sich beteiligt fühlen, ohne Dominanzen oder Zurückhaltungen (Goodwin, 2009, S. 195).

Wie auch im Interview werden in der Fokusgruppe einführende Fragen gestellt, um die Information über die Heterogenität der Gruppe einzusammeln und die Teilnehmenden sich einander kennenlernen zu lassen.

- Könnten Sie uns bitte etwas über sich selbst kurz erzählen-sind sie Studierende? In welchem Bereich sind Sie tätig? In welchen Programmiersprachen haben Sie schon Vorerfahrungen?

Da die Fragen formaler sein sollen, als die Fragen für individuelle Interviews (Goodwin, 2009, S. 195), und da wir bei den Interviews schon herausgefunden haben, dass potenzielle NutzerInnen sich eher mobile Anwendung vorstellen würden, um die Programmiersprache bequem und mobil zu lernen, verzichten wir auf die Fragen des Nutzungsformats und kommen gleich zu dem Problemfeld. Außerdem werden die Probanden gebeten, uns die Probleme, falls solche vorhanden sind, nicht nur zu beschreiben, sondern auch zu zeigen.

- Was sind Ihrer Meinung nach wichtige Schritte für das Lernen einer Programmiersprache?
- Wie lernen Sie die Programmiersprachen? oder wie würden Sie sich den Lernprozess vorstellen?
- Wie würden Sie sich den Ablauf einer Übung in der App vorstellen?
- Was erwarten Sie von einer mobilen Anwendung oder vom Websystem für den erfolgreichen Lernprozess für ProgrammieranfängerInnen? Welche Faktoren sind Ihnen bei der Produktauswahl am wichtigsten?
- Haben Sie schon mal eine Programmiersprache über eine Online Plattform erlernt?
- Was sind heute die größten Probleme oder Ineffizienzen in dem Lernprozess mithilfe von mobilen Applications?

- Haben Sie schon schlechte Erfahrungen oder Probleme bei der Interaktion mit den Produkten gehabt, welche für den Lernprozess der Programmiersprachen entwickelt sind?
- Können Sie uns jetzt solche Systeme zeigen oder nennen, die Sie heute nutzen für das Lernen der Programmiersprachen oder bei denen Sie auf Probleme gestoßen haben?
- Wollten Sie schon irgendwas auf der Web/Appanwendung tun, was aber nicht möglich war? Und was waren das für Schwierigkeiten?
- Auf welche anderen Probleme können Sie beim Lernen auf der Lernplattform stoßen?

Danach sollen die Teilnehmenden die Ziele eines solchen mobilen Produktes verdeutlichen, indem sie über wichtigste Funktionalitäten reden, z.B. die Fragen nach den Funktionalitäten beantworten, die noch nicht in anderen Systemen existieren (Goodwin, 2009, S. 116) und somit bei der Entwicklung unseres Produkten berücksichtigt werden können. Für diesen Zweck werden exlorative open-ended Fragen gestellt (Lazar, 2017, S. 194).

- Was sind Ihrer Meinung nach wichtige Schritte für das Lernen einer Programmiersprache?
- Wie lernen Sie die Programmiersprachen? oder wie würden Sie sich den Lernprozess vorstellen?
- Wie würden Sie sich den Ablauf einer Übung in der App vorstellen?
- Was erwarten Sie von einer mobilen Anwendung oder vom Websystem für den erfolgreichen Lernprozess für ProgrammieranfängerInnen? Welche Faktoren sind Ihnen bei der Produktauswahl am wichtigsten?

In der Fokusgruppe werden präzise Fragen über die Designmöglichkeiten gestellt. Diese Fragen sollen uns helfen, die Antwort zu finden, ob unsere Designabsichten erreicht wird. Also, die Antworten von den Befragten dienen nicht der Orientierung für unsere Designentscheidungen, sondern dem Verständnis und Berücksichtigung, welche Attribute das Design "powerful, simple, smart, or friendly" machen (Goodwin, 2009, S. 651).

Die Fragen werden in der Gruppe gestellt und es werden alle möglichen Ideen gesammelt. Es wird von uns beachtet, dass wir die NutzerInnen dazu nicht zwingen dürfen, Lösungen für das Design zu generieren (Goodwin, 2009, S. 124). Deshalb wird in der Diskussion, den Teilnehmenden angeboten, sich eigene Ideen zu notieren, welche dann in der Fokusgruppe erwähnt und später uns abgeben werden. Somit wird eine große Bandbreite an Meinungen gesammelt (Goodwin, 2009, S. 195).

Die Frage über den Unterhaltungsaspekt der Anwendung soll gleich den Überblick geben, wie stark Unterhaltungselemente oder motivationsfördernde Elemente erwartet werden.

- Würden Sie Python für die Unterhaltungsziele oder für beruflichen Einsatz lernen?
- Soll der Produkt Unterhaltungselemente beinhalten? Oder wie stellen Sie sich den Unterhaltungsaspekt vor?

In den nächsten Fragen wird auf das Thema der Gamification-Elemente im Lernprozess eingegangen. Die Fragen werden basierend auf den Taxonomien von Bartle (1996), Nacke et al. (2011) und Marczewski (2015) und dementsprechend zugewiesenen Spielmechaniken entwickelt (Xu, 2011; Webb, 2013; Tondello et al., 2016; Toda et al., 2019; Friedrich et al., 2020; Schöbel et al., 2020).

Ein guter Fragensatz wird von Schneider et al. (2017) entwickelt, mit den Fragen, die gestellt werden können, um die Spielertypen und gewünschte Unterhaltungselemente zu extrahieren.

- Würden Sie oder besprechen Sie Ihren Lernvorgang, Erfolg oder Unklarheiten beim Lernen mit KommilitonInnen, FreundInnen, ExpertInnen? Wie würden Sie das machen? Würden Sie einander telefonieren, miteinander chatten oder einander Ihre Fortschritte auf eine gewisse Weise zuschicken?
- Welche sozialen und kommunikativen Funktionalitäten würden Sie in einer Anwendung für das Python-Lernen bevorzugen?
- Wie finden Sie Belohnungen für Lernen? Oder wie finden Sie das System, dass Ihnen Abzeichen zuweist oder mit bestimmte Rewards, neuen Funktionalitä-

ten, Orden, Auszeichnungen, Leistungsfähigkeiten für erledigte Levels, Aufgaben, Quizen belohnt? Welche Belohnungen finden Sie am sinnvollsten?

- Welche explorativen Erkundungsfunktionalitäten würden Sie gerne in der Anwendung treffen? Würden Sie sich eine Anwendung wünschen, die kein Ende im Lernprozess oder stundelange Exploration anbietet? Würden Sie gerne nach Bugs suchen?
- Wie fühlen Sie sich in den Wettbewerben? Wie finden Sie die Funktionalitäten in der Anwendung, die eine Konkurrenz mit anderen voraussetzen? Unter welchen Bedingungen könnte bei Ihnen Aggressivität entstehen?
- Wie wollen Sie Ihren Lernfortschritt sehen oder kontrollieren können? Wie fühlen Sie sich, wenn Sie in neues Level oder neue Leistungen erreichen?

Im Hintergrund der Diskussion mit der Fokusgruppe wird immer von uns beachtet, dass es auch Nachteile und Erwartungen gibt. So ist die Bewertung des Interaktionsdesigns ziemlich nutzlos und kompliziert, weil self-reporting Berichte und die Ideen, die nicht auf den Erfahrungen mit ähnlichen Produkten basieren, keine reales Verhalten der Kunden und NutzerInnen vorhersagen und widerspiegeln können (Goodwin, 2009, S. 650). Denn bevor mit dem Produkt interagiert wird, anstatt das Produkt nur anzusehen oder sich es vorzustellen, sind die Meinungen von den potenziellen NutzerInnen wahrscheinlich uninformativ (Goodwin, 2009, S. 650).

Die Teilnehmenden in der Fokusgruppe sind in unterschiedlichen Bereichen tätig. Es gibt Personen, die Rechtswissenschaften und Wirtschaftswissenschaften fertig studiert haben und keine Erfahrungen in den Programmiersprachen haben. Die anderen Personen haben schon Vorerfahrungen in verschiedenen Programmiersprachen und studieren medizinische Informatik an der OTH Regensburg und Medieninformatik als Bachelorfach. Die Fokusgruppe ist demnach gemischt.

Die Diskussion in der Fokusgruppe hat gleich mit den Ideen angefangen, dass man beim Lernen gleich auch die Ergebnisse sehen können will und dass der Lernprozess schrittweise verlaufen soll, unterstützt von interaktivem Material und erklärenden Anweisungen, wie der Lernprozess und die Vorbereitung richtig sein sollen. Beispielsweise erwähnt einer Teilnehmende, dass er gerne die Entwicklungsumgebung problemlos vorbereitet haben will, indem jemand aus den Tutoren ihm helfen

würde, wenn er auf die Installationsprobleme stoßt. Für diese Person ist es wichtig, dass er sich bei Problemen oder Unklarheiten in den organisatorischen Situationen oder in den Situationen, wenn das Design intuitiv verständlich sein soll (ist aber nicht), gleich an jemanden wenden kann, um nicht viel Zeit damit zu verlieren.

Die Frage, wie man sich den Lernprozess einer Programmiersprache und den Ablauf einer Übung auf einer Anwendung vorstellt, kommen vielfältige Antworten und Anregungen. So ist die Möglichkeit, selbst Code zu tippen und auszuführen, scheint bei einer Person sehr relevant zu sein. Außerdem verläuft der Lernprozess desto schneller, je aktiver man mithilfe von solchen Elementen wie Illustrationen, Schemas, Karten in das Prozessflow involviert ist. Es wurde von mehreren Teilnehmenden betont, dass man nicht nur den Fortschritt sehen will, sondern auch Unterstützung wie z.B. Tipps, Hinweise oder begleitende Erklärungen in den Übungen hätte. Besonders in den schwierigeren Aufgaben oder an den potenziell problematischen Stellen sollen Hinweise umgesetzt werden.

Das Design der Anwendung für das Lernen der Programmiersprachen würde man ästhetisch schön, strukturiert, ordentlich, abwechselungsreich aber simple, minimalistisch, nicht zu bunt und nicht überfordernd bevorzugen. Man möchte die Möglichkeiten haben, nicht nur bestimmte Aufgabenart lösen zu können, sondern auch einen umfangreichen Set an unterschiedlichen Aufgaben zu haben, wie z.B. selbst den Code schreiben, selbst den Test machen.

Unter den schon bekannten und benutzten Anwendungen wurden "Udemy" Webplattform erwähnt, wo man sich bestimmte Kurse von verschiedenen Schwierigkeitsgrad und für verschiedene Programmiersprachen kaufen kann. Zwei teilnehmenden Personen haben sich schon einen Kurs gekauft und mitgemacht. Die Preise sind unterschiedlich. Die andere Hälfte der Teilnehmenden meinen, sie sind nicht bereit, sich für einen großen Preis einen Kurs zu kaufen. Der Preis von 2-3 Euro für einen Kurs oder Anwendung finden sie aber noch akzeptabel. Eine Person diskutierte die Gerechtigkeit der kostenpflichtigen Kurse, indem der Kauf eines Produktes somit die EntwicklerInnen unterstützen soll. Dazu sagte die Person, dass es auch so organisiert werden soll, dass die EntwicklerInnen, auch wenn sie ihre Produkte kostenlos anbieten, trotzdem profitieren könnten. In der Diskussion über die Notwen-

digkeit verschiedenen Anwendungen hat eine Person noch gemeint, dass sie lieber zusammen mit allen KomillitonInnen lernen würde, als sich eine Anwendung zu installieren.

Es wird auch die Frage der Werbung diskutiert. Dazu meinte die Mehrheit, dass es auf der Lernplattform auf die Werbung verzichten werden soll.

Während der Diskussion über verwendete Plattformen hat sich eine neue Frage rauskristalliert: Ob man sich gerne ein Zertifikat nach dem erfolgreichen Abschluss der Übungen in der Anwendung wünscht. Dahingende hab sie gegensätzliche Meinungen gebildet. Eine Hälfte meinte, solch ein anerkennbares Zertifikat wäre eine extra Motivation, um die Anwendung zu benutzen, oder eine große Möglichkeit für beruflichen Einstieg für Selbstständige. Ein Zertifikat zu erhalten wäre auch eine gute Möglichkeit in anderen Bereichen wie z.B. Kriminalität oder Medizin, wann man selbst die Programmiersprache erlernen kann und somit als RechtswissenschaftlerInnen in digitalem Kriminalitätsbereich tätig werden kann. Sowie an solchen Jobsuche-Plattformen wie Indeed oder LinkedIn wären Zertifikate vielleicht auch ein gutes Feature.

Die andere Hälfte widerspricht der ersten Hälfte und meint, solche Zertifikate sollen richtig einen entsprechenden Kenntnisstand nachweisen können, was auf einer Anwendung schwierig validierbar oder überprüfbar wäre, sowie die Metriken, die dieses Zertifikat anerkennbar machen können, sollen gut durchdacht werden. Dazu wird betont, dass eher universitäre Ausbildung eine wichtige Rolle spielen soll.

Zu dem Thema der Unterhaltungselemente zeigt sich auch eine Vielfalt an Meinungen. Während einige das Belohnungssystem schön und somit motivierend finden, würden die anderen sich eher einfaches, funktionales, nicht spielerisches Design vorstellen. Punkte, die man erreichen kann, wären für einige Befragte ein gutes Zeichen von dem, was man schon kann, wie erfolgreich man die Aufgaben löst und wie weit man vom Ziel entfernt ist. Ein Person meinte, sie hätte sich gerne ein "Carrot ans Stick"-System vorgestellt, bei dem man für gute Arbeit gelobt oder für schlechte Übungen auch bestraft wird.

Alle Teilnehmenden waren der Meinung, dass sie sich Leaderboard oder Rang-

listen nicht wünschen, um nicht immer in der Konkurrenz zu jemanden zu stehen. Gerne hätten sie Gruppenchallenges oder Gruppenaufgaben, die sie gerne in einem Kollektiv lösen würden. Direkter Wettbewerb oder direkte Bewertung der Kenntnisse im Vergleich zu anderen wären strengstens unerwünscht.

Während einige Personen sich den schrittweisen Ablauf des Lernens mithilfe von Anwendung vorstellen, von einer Aufgabe zu einer anderen, mit aufsteigendem Schwierigkeitsgrad, wünschen sich die anderen Befragten eher eine Freiheit bei der Auswahl der Themen, indem sie auch die Möglichkeit haben, bestimmte Information überspringen zu können und zu einem bestimmten Teil gezielt zu kommen.

Kommunikationsaspekte spielen für die Befragten keine Rollen. Somit brauchen sie keine Chat- oder Forum-Funktionalitäten.

Allgemein lässt sich sagen, dass die Teilnehmenden gut getraut haben, auch gegenüberstehende Meinungen auszuäußern. Die ganze Fokusgruppe ist in einem freundlichen, entspannten Klima gelaufen, bei dem man sich wohl gefühlt hat und immer gerne die Ansichten mit anderen geteilt hat.

Es ergeben sich sehr unterschiedliche Meinungen: während einige sich den Lernprodukt als rein funktionsorientierte, kostenlose und "nüchterne" Anwendung vorstellen, meinen die anderen, sie würden gerne doch in einem spielerischem Kontext üben, aber ohne direkte gegeneinander stehenden Challenges, sondern in einem kollektiven Klima und einer Gruppe. Gerne wären einige mit verschiedenen Belohnungsmitteln gelobt oder auch auf irgendwelche Art und Weise bestraft.

Gerne würden sich einige befragte Personen den Ablauf einer Übung als schrittweiser Vorgang von einem Niveau zu einem anderen vorstellen. Gleichzeitig meinen aber die anderen, dass sie sich lieber selbst entscheiden würden, welches Thema, welche Übung sie machen möchten, ohne durch vorherige Levels verhindert zu werden.

Die Diskussion über die Zertifikate enthüllt ein großes Spektrum an Meinungen, die gerne ausgetauscht wurden. Von der Akzeptanz der Idee, ein Zertifikat auf einer Lernanwendung nach dem bestimmten Kursabschluß erhalten zu können, bis zur Skepsis über dessen Gültigkeit und Anerkennung.

Aus einer Meinungsvielfalt und oft Kontrastmeinungen, kristallisiert sich eine

Liste an Anforderungen und Wünschen einer Fokusgruppe aus.

Hier wären folgende Aussage für eigene Entwicklung zu berücksichtigen:

- Unterhaltungssystem, wie Punkte, neue Levels, wird erwünscht
- Design soll trotzdem funktional und nicht spielerisch sein
- Zertifikat nach dem Kurs wäre erwünscht
- Zertifikat ist nicht die größte Motivation
- Hinweise, Tipps sollen umgesetzt werden
- Illustrationen, Bilder, Schemas sind hilfreich für das Lernen
- intuitiv verständliches Design
- Progressleiste, Kontrolle über den Stand, Übersicht der Ergebnisse
- Strukturiertes Lernmaterial mit hohem Freiheitsgrad der Auswahlmöglichkeiten
- schrittweiser Vorgang von einem Niveau zu einem anderen

Die Fokusgruppe wird als Video aufgenommen, damit das Material danach interpretiert werden kann. Dadurch kann auch analysiert werden, ob die ModeratorIn die Meinungen von Minderheiten nicht übersehen hat (Goodwin, 2009, S. 196).

Bevor die Fokusgruppe-Diskussion und Videoaufnahmen gestartet werden, unterzeichnen die Teilnehmenden das Einverständniserklärung (siehe Anhang).

3 Anforderungsspezifizierung

3.1 Personas

Personas helfen fundamentale Bedürfnisse und Charakteristiken der NutzerInnen und KundInnen zu verstehen. Ziele, mentales Model, Umgebung, Verhalten, Kenntnisse, Frustrationen und Spaßgefühle der Persona sollen die Muster, Pattern der potenziellen NutzerInnen verständlich machen (Goodwin, 2009, S. 229). Personas werden für unterschiedlich Ziele, Produkte und Dienstleistungen entwickelt: für Web-Seiten, Organisationsstrukturen, Business Prozesse, Kurse, Veranstaltungen, Massnahmen... (Goodwin, 2009, S. 231).

Personas sind auch Szenarios, die einen Überblick über die Bedürfnisse geben. Besonders spielen Personas eine große Rolle vor der Designentwicklung: Sie beantworten die Fragen über die Relevanz der Navigationsmöglichkeiten, farbliche Darstellung, benutzte Begriffe, Aufbau (Goodwin, 2009, S. 232).

Personas helfen mit dem "user in mind" (Goodwin, 2009, S. 232) zu denken. In Marketing können Personas dem Unternehmen ein gutes Feedback vermitteln über die Funktionalitäten, die tatsächlich nützlich sein sollen oder die nicht besonders wichtig sind (Goodwin, 2009, S. 234).

Personas ist die Repräsentation realer Personen und keine bullet points der abstrakten Ideen. Personas werden in einem narrativen Aspekt geschrieben (Goodwin, 2009, S. 235). Persona ist nicht ein durchschnittlicher Kunde, sondern eine Menge an einzigartigen Patterns, Mustern und Zielen, die bei potenziellen Kunden betrachtet werden (Goodwin, 2009, S. 238). Somit werden in unserer Studie bestimmte Nutzergruppen bestimmt. Für jede wird eine Persona erstellt. Diese Personas entsprechen den realen Wünschen, Bedürfnissen und Gewohnheiten, die sich basierend auf Interviews und der Fokusgruppe rauskristallisiert haben.

Die Vorgehensweise beim Erstellen von Personas soll folgendemäßen sein:

1. Die Teilnehmenden der Interview und der Fokusgruppe sollen nach ihren Rollen kategorisiert werden. Das bedeutet, es soll analysiert werden, welche Aufgaben von den Teilnehmenden durchgeführt werden, und nicht wie sie durchgeführt werden. Jede identifizierte Gruppe wird einzeln behandelt, um

die Patterns jeder Gruppe ungestört zu analysieren (Goodwin, 2009, S. 244-246).

Da aus unseren Interviews und der Fokusgruppe sich ergeben hat, dass man sich den Lernprozess sowohl mithilfe von Laptop, als auch als mobile Anwendung vorstellt, wird sich in unserem Projekt geeinigt, dass die beiden Layouts entwickelt werden sollen - sowohl für den grösseren Bildschirm, als auch für den durchschnittlichen Handybildschirm.

Produkt	Erwartete Rollen	Beobachtetes Verhalten
Mobile und Webanwendung für das Lernen der Programmiersprache Python	<ul style="list-style-type: none"> – Personen, die gerne mit Belohnungssystem lernen – Personen, die sich nur Funktionalitäten wünschen – Personen, die sowohl mit Belohnungssystem, als auch ohne lernen – Personen, die Videos, Schemas, Diagramme, Bilder anschauen und Übungen, Quizen machen – Personen, die zu einem bestimmten Lernbereich kommen wollen – Personen, die zu einem bestimmten Niveau schrittweise kommen wollen – Personen, die ihren Fortschritt sehen und kontrollieren wollen – Personen, die sich Wiederholungen wünschen – Personen, die den Code praktisch ausprobieren wollen – Personen, die sich die notwendige Sprache auf der Anwendung einstellen wollen – Personen, die sich im Forum oder in Frage/Antwort austauschen – Personen, die alleine lernen – Personen, die für ein Zertifikat lernen – Personen, die Zertifikat nicht als Motivationsfaktor beachten 	Meherere Teilnehmenden in den Interviews oder Fokusgruppe waren in mehreren Rollen, weil jede Rolle einer bestimmten Funktionalität auf der Anwendung entspricht. Somit wünscht sich jede Rolle nicht nur eine Funktionalität.

2. Demographische und Verhaltensvariablen, die unter den befragten Teilnehmenden sich unterscheiden, sollen identifiziert werden. Für jede Rolle soll mentales Modell und Häufigkeit der Aufgabendurchführung analysiert werden, sowie auch die demographischen Daten sollen gesammelt werden. Beispielsweise die Umgebung, die die Aufgaben beeinflussen. Außerdem kann die Häufigkeit der Aufgabendurchführung berechnet werden, mit den Wörtern wie wenig, viel, oft, selten... Jeder Variable wird mehrere oder einer Optionen zugewiesen.

Mentales Model, wie z.B. Musikliebhaber, wird von Verhalten und Tätigkeit beeinflusst. Motivationen und Ziele, wie z.B. Liebe zu einer bestimmten Kleidungsmarke, sind wichtig um die Patterns herauszufinden. Häufigkeit der Aufgaben, der Aktivitäten, wie z.B. Häufigkeit der Organisation einer Musikbibliothek, sollen nicht mit der Dauer der Aktivität gemischt werden. In diesem Fall soll Häufigkeit und Dauer als zwei verschiedene Variablen gesetzt werden. Die Berechnung der Quantität in den Aktivitäten sind nicht kritisch, können aber die Muster klarer machen. Zu beachten sind die Beziehungen, emotionale Abhängigkeit zu den Aufgaben, wie z.B. Spaß bei bestimmter Aktivität. Der Unterschied zwischen den Kenntnissen und Technologie soll sehr aufmerksam behandelt werden. Eine Krankenschwester ist die Expertin in ihrer Tätigkeit, aber nicht die Expertin beim Nutzen von Computern. Auch wenn die Rollen und Nutzergruppen nach den Aufgaben klassifiziert werden, bedeutet das aber noch nicht, dass alle Nutzergruppenmitglieder sich bei der Aufgabe auf den gleichen Prozessen konzentrieren. Es gibt auch andere Variablen, die je nachdem berücksichtigt werden können, wie z.B. Familienstrukturen, Geographie der anzubietenden Güter oder die Business Umgebung. (Goodwin, 2009, S. 246-252)

Variable	Spektrum	
gerne mit Belohnungssystem lernen	für Punkte, Sterne	für neue Levels
nur Funktionalitäten	ohne Punkte, Sterne, Lebenschance	
sowohl mit Belohnungssystem, als auch ohne lernen	notwendig	nicht notwendig
Videos, Schemas, Diagramme, Bilder anschauen und Übungen, Quizen machen		
zu einem bestimmten Lernbereich kommen	unabhängig und flexibel	
zu einem bestimmten Niveau schrittweise kommen	spielerisches Aspekt	Fortschrittskontrolle
Code für Praxis	unbedingt	keine Angabe
Wiederholungsmöglichkeit	unbedingt	keine Angabe
den Fortschritt sehen und kontrollieren	via Progressleiste, Kontrollbalken	via Levels, Schwierigkeitsgrad
sich die notwendige Sprache auf der Anwendung einstellen	unbedingt	keine Angabe
sich im Forum oder in Frage/Antwort austauschen	ab und zu	nie
alleine lernen	immer	gelegentlich
für ein Zertifikat lernen	für beruflichen Einstieg	
ein Zertifikat nicht als Motivationsfaktor beachten	keine Anerkennung	keine Validierungen und Zuverlässigkeit

3. Mapping zu den Variablen. Jeder befragte Teilnehmende soll relativ zu den

anderen Teilnehmenden in dem ganzen Spektrum platziert werden. So können sich Cluster dynamisch rauskristallisieren und es kann sich ergeben, dass einige Variablen eigentlich nicht mehr nötig sind (Goodwin, 2009, S. 252-256)

Diese Etappe wird überprungen, weil die Definierung der Rollen und das entsprechende Model-Mapping unter Berücksichtigung der Aussagen der Teilnehmenden durchgeführt wurden. Somit werden keine unnötige Variablen in der 2. Etappe gefunden. Die "ohne Angabe" werden automatisch nicht berücksichtigt.

4. Identifikation der Patterns und Muster. Gemeinsame Eigenschaften, die einem Patter zugewiesen sind, sind Protopersonas (Goodwin, 2009, S. 256-270).

Von uns durchgeführte Clustering wird in der Abbildung 8 visualisiert.

Es gibt einige Variablen, die nicht von allen Personen befüllt werden konnten. Erstens, die Interviews tendierten zu einem semistrukturierten Model, indem Zweitens, die Daten, die für Erstellung von Personas benutzt wurden, stammen gemischt - aus den Interviews und aus der Fokusgruppe. Die Fokusgruppe war schwieriger zu interpretieren, weil es eine kollektive Diskussion war, bei der nicht auf einzene Person bei jeder farge eingegangen wurde. Außerdem wird in unserer Studie die Vielfalt der Meinungen aus der Fokusgruppe berücksichtigt, indem die Etappe 1. basierend auf den Ergebnissen aus den Interviews und aus der Fokusgruppe durchgeführt wurde. Somit ist die Abgrenzung von jeder einzelnen Person aus der Fokusgruppe nicht relevant.

5. Wenn nötig, andere Arten der Personas hinzuzufügen. Vielleicht sind für die Stakeholders nicht nur potenzielle NutzerInnen interessant, sondern auch die KäuferInnen, die Kunden. Oder vielleicht sind für die Stakeholders negative Personas wichtig, für die das Produkt nicht anlockend ist (Goodwin, 2009, S. 270-275).
6. Die Personas gruppieren und priorisieren: primäre Personas als Design Target, sekundäre sind nicht wesentlich anders als primäre (Goodwin, 2009, S. 275-277).

In unserem Fall gibt es keine anderen Personas. Potenzielle könnten es sein

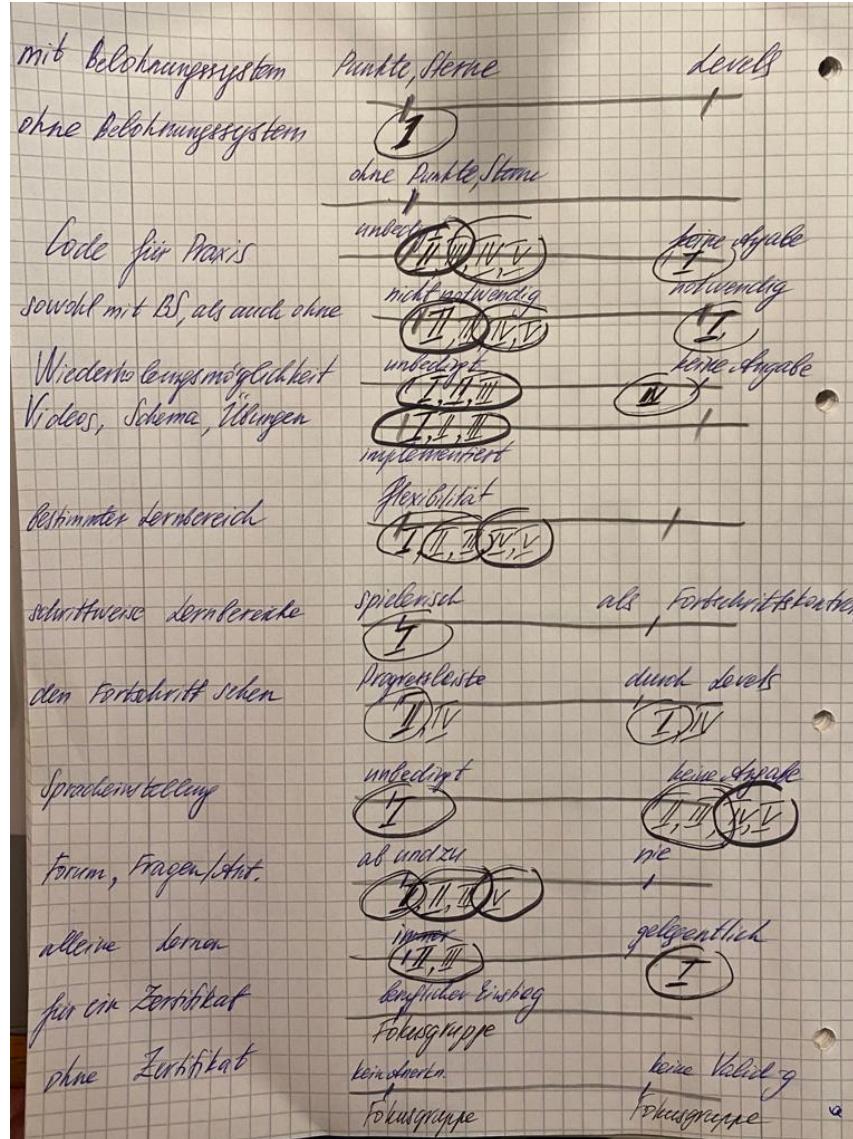


Abbildung 8: Clustering von Patterns

diejenigen, die sich die Anwendung für das Lernen der Programmiersprache runtergeladen oder gekauft haben.

7. Narrative Beschreibung der Personas mit der unmittelbaren Beschreibung, Photos, Diagrammen, Collages (Goodwin, 2009, S. 277-291).

Basierend auf den vorherigen Etappen 1.-6. werden narrative Beschreibungen erstellt (siehe Abbildung 9, 10, 11).

Die Abbildung 9 widerspiegelt die Clustergruppe I aus der Abbildung 8. Die Clustergruppe I charakterisiert sich durch eine klare Ausprägung der Bedürfnissen an Unterhaltungsaspekt auf der Anwendung. Das lässt sich gut mer-

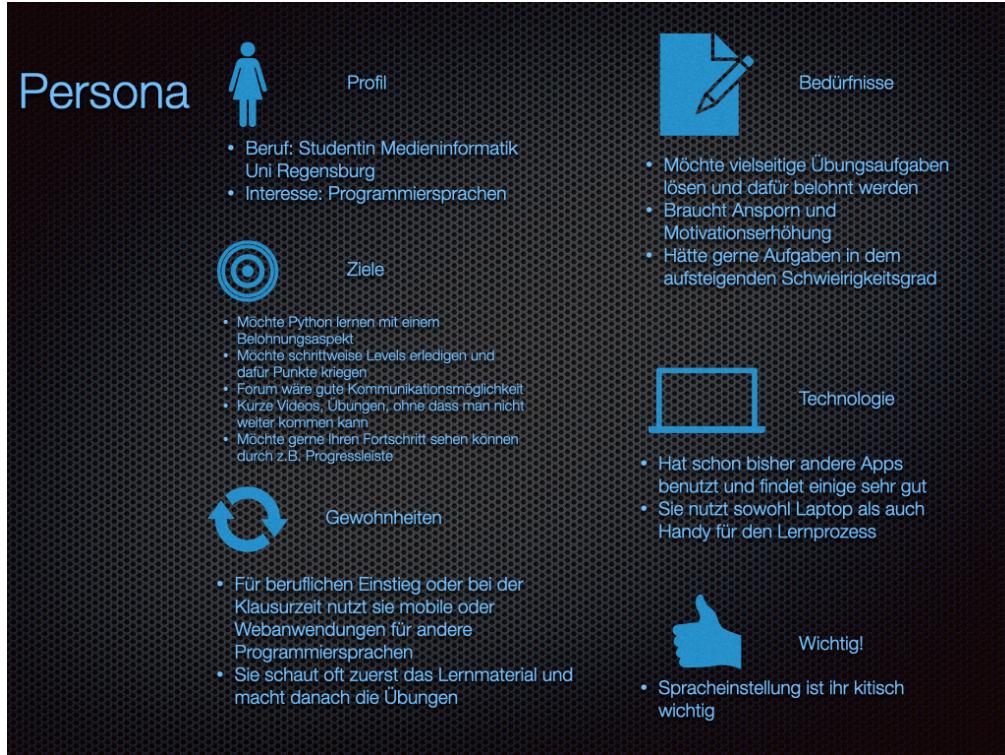


Abbildung 9: Persona 1

ken, weil die Persona sich nicht nur Belohnungen wünscht, sondern auch schrittweise Levelsstruktur, was bedeuten kann, dass die Persona Merkmale eines Achiever-Spielertyps zeigt (Nacke et al., 2011; Xu, 2011). Sie wünscht sich, dass ihre Belohnungselemente nie erschöpft werden. Für die Persona ist sehr wichtig, dass die Anwendung Möglichkeiten der Spracheinstellung ermöglicht.

Die Abbildung 10 stellt eine Persona II, III, IV, V dar, die noch keine Vorerfahrungen in anderen Programmiersprachen besitzt, ausser Layout-Sprache für Design der Webseiten. Die Persona hat noch nie Anwendungen für das Lernen der Programmiersprachen benutzt. Und bevorzugt ein nüchternes Design der Anwendung, ohne Spielelemente. Diese Persona würde gerne den Fortschritt kontrollieren können.

Die Abbildung 11 entspricht einer schon in Programmierkenntnissen erfahrenen Persona. Sie wünscht sich einen hohen Freiheitsgrad bei der Auswahl der Übungen und Kapiteln. Die Unterhaltungselemente spielen für diese Persona keine große Rolle. Wichtig ist es, dass die Persona die Hinweise und Hilfe vom



Abbildung 10: Persona 2

System bekommen kann, wenn die Fehler nicht selbstständig beheben werden können.

Zusammenfassend lässt sich sagen, dass während der Erstellung von Personas sich 3 Nutzergruppen ergeben: eine Nutzergruppe kennt sich schon im Bereich der Informatik aus und wird stark durch Wünsche nach spielerischen Elementen geprägt.

Die zweite Nutzergruppe deckt diejenigen potenzielle NutzerInnen ab, die noch keine Erfahrungen in den Programmiersprachen haben und ganz anderes Fach studiert haben. Von solchen NutzerInnen wird ein einfaches, funktionales und verständliches Design erwünscht. Diese Nutzergruppe hat noch keine gewissen Vorstellungen über das, welche Lernelemente auf der Anwendung umgesetzt werden sollen.

Die dritte Nutzergruppe wiederspiegelt die Mehrheit der potenziellen NutzerInnen: Studierende mit wenigeren oder großen Programmierkenntnissen, die aber noch Python lernen wollen. Für diese Nutzergruppe spielt der selbstständige Lernprozess eine große Rolle, welcher nicht unbedingt von Unter-



Abbildung 11: Persona 3

haltungselementen unterstützt werden soll. Diese Nutzergruppe kombiniert theoretischen Lernprozess mit dem praktischen. Somit wäre eine große Freiheit der Auswahl von bestimmten Lernkapiteln sehr wichtig. Außerdem brauchen diese potenziellen NutzerInnen eine gute Übersicht über die Struktur auf der Anwendung.

3.2 User Stories

User stories beschreiben nicht das gesichtslose Szenario von einem Startpunkt zu einem Endpunkt, sondern fokussieren sich auf den Personas, auf deren Bedürfnissen und Gefühlen (Goodwin, 2009, S. 310).

Als Persona 1 mit einigen Programmierkenntnissen in anderen Programmiersprachen möchte ich die Anwendung für das Lernen der Programmiersprache Python spielerisch nutzen, sodass mit aufsteigenden Schwierigkeitsgrad durch vielfältige Übungen neue Programmiersprache mit großer Motivation erlernen kann.

Persona 2 ohne Programmierkenntnisse kann neue Sprache auf einer guten funktionalen Anwendung ohne Spielemente erlernen, damit die Persona die Sprache

beherrschen kann, ohne sich vorstellen zu können, welche genau Elemente auf der Anwendung sein sollen.

Persona 3 mit guten Kenntnissen in unterschiedlichen Programmiersprachen möchte gerne eine strukturierte und übersichtliche Lernanwendung benutzen, indem die Persona selbst frei auswählen kann, welche Art der Aufgaben und welche Thematik sie selbstständig machen kann, um nicht unbedingt mit Unterhaltungselementen präzise Lernmaterial zu neuer Programmiersprache zu nutzen.

3.3 Use-Cases

Bei Use-Cases handelt es sich im groben um eine Interaktion zwischen einem Actor und einem System. Ein Actor kann dabei sowohl ein Mensch, als auch ein anderes System sein. Use-Cases werden im Allgemeinen in Form von Diagrammen beschrieben. Die Ziele des jeweiligen Akteurs sind dabei auf einem niedrigerem Level als die Ziele der Personas, außerdem spielt es hier keine Rolle, wie der Benutzer über eine Interaktion oder Aufgabe empfindet (Goodwin, 2009, S. 309).

Neben dem Actor und dem System kann Use-Case auch noch aus weiteren Elementen bestehen. Ein Stakeholder ist jemand oder etwas, der Interesse daran hat, auf welche Weise sich das System verhält. Es kann außerdem auch eine Precondition oder mehrere Preconditions geben, das sind Voraussetzungen, die erfüllt werden müssen, bevor oder nachdem der Anwendungsfall ausgeführt wird. Es kann außerdem Trigger geben, also einen Auslöser, durch den der Anwendungsfall initiiert wird. Neben den Hauptpfaden kann es auch noch Alternative Paths geben, also Pfade, die eine Variation des Hauptthemas darstellen.

Use-Cases können dabei helfen, verständlich zu machen und zu verdeutlichen, wie sich ein System verhalten soll. Außerdem können sie dem Projektteam durch eine die Zusammenstellung von Zielen und Aufgaben dabei helfen zu ermitteln, wie komplex das System werden soll. (*Use Cases*, o.J.).

Der Actor hat bei der Teach Me Code App eine Reihe an Zielen beziehungsweise Aufgaben die erfüllt werden können (siehe Abbildung 12).

Als erstes kann er die Sprachauswahl der App betätigen. Nachdem die Sprache ausgewählt ist, kann er sich für einige verschiedene Wege entscheiden. Er kann sich

zum Beispiel dafür entscheiden, ein Kapitel auszuwählen. Dort kann er dann entweder das Lernmaterial betrachten und gegebenenfalls das darin enthaltene Video abspeichern, oder sich für die zum Kapitel gehörigen Aufgaben entscheiden und diese lösen.

Statt der Auswahl eines Kapitels kann der Actor auch auf das Forum zugreifen. Dort ist er dazu in der Lage, mit anderen Usern zu interagieren und sich mit diesen auszutauschen.

Außerdem ist es für den Nutzer möglich, auf einen Kalender zuzugreifen.

Er ist außerdem dazu in der Lage, sich über seinen Fortschritt zu informieren.

Zusätzlich kann er die App auch mit Außenstehenden teilen.

3.4 Hierarchische Taskanalyse

Bei der hierarchischen Taskanalyse (HTA) geht es nicht um die Handlungen oder Auflistung von physischen oder kognitiven Prozessen, sondern im Gegensatz, es geht um ein Verfahren, das darauf abzielt, Leistungsprobleme zu identifizieren und Lösungen vorzuschlagen. Das bedeutet, es sollen Lösungen für anfänglich spezifizierte Probleme bereitgestellt werden (Diaper & Stanton, o.J., S. 70).

HTA beginnt mit der Identifizierung der Ziele der Aufgabe. Aktionen sollen nicht einfach aufgelistet werden, sondern sollen verstanden werden. Komplexe Systeme werden zielgerichtet entworfen, und zu verstehen, wie ein System sein festgelegtes Ziel erreicht oder nicht erreicht, ist der Hauptzweck der Analyse. Da Ziele oft komplex sind, soll die Analyse diese Komponentenzielzustände durch den Prozess der Zerlegung spezifizieren. Es werden zuerst Zielzustände identifiziert, die durch mehrere Kriterien spezifiziert sind. Danach können Unterziele abgegrenzt werden, die dem Erreichen des Gesamtzielzustands dienen sollen. Verschachtelte Hierarchie von Zielen und Teilzielen soll auf eine gewisse sukzessive Weise entpackt werden (Diaper & Stanton, o.J., S. 70).

In unserer Studie sind die Input-Variablen die Ergebnisse aus den Interviews, Fokusgruppe.

Die wichtigsten Nutzerziele ist effektiver Lernprozess einer neuen Programmiersprache. Dafür sollen Lernumgebungen entwickelt werden, die diesen Lernprozess

fördern. Diese Lernumgebung beinhalten in einem Prototyp Gamification-Elemente und in einem anderen keine Gamification-Komponente.

Schlüsselaufgaben für die gamifizierten Prototypen und nicht gamifizierten Prototypen sind:

- Sprache für die ganze Lernanwendung einstellen können
- Lernthema/Kapitel auswählen können
- In jedem Kapitel sich entscheiden können, ob der theoretische Teil oder praktische Teil ausgewählt wird
- In dem theoretischen Teil werden textuelle und ausdio-viduelle Elemente dargestellt
- In dem praktischen Teil werden Aufgaben gemacht
- Möglichkeit, bookmarkte Videoelemente anzuschauen
- Möglichkeit, Benachrichtigungen zu kontrollieren
- Möglichkeit, die App den anderen zu teilen

Extra Schlüsselaufgaben für die gamifizierten Prototypen sind:

- Kommunikationsmöglichkeiten im Forum/Chat
- prozentuelle Progresskontrolle durch einen Progresskreis für alle Kapitel
- Aktivitätskalender, um eigene Aktivität zu kontrollieren
- Kontrolle über die Anzahl an Punkten/Sternen

Das erste Ziel ist die Auswahl eines Themas, welches man erlernen will. In den gamifizierten Prototypen ist diese Auswahl nicht frei, sondern hierarchisch: erst nachdem man ein Thema erarbeitet hat, kann man zu einem anderen Thema kommen. In den nicht gamifizierten Prototypen ist die Auswahl des Lernkapitels frei und flexibel.

Die zweite Gruppe, die in jedem Kapitel inkludiert ist, besteht aus zwei Komponenten: man soll sich entscheiden, entweder theoretischen Teil durchzulesen und am Videoelement anzuschauen, oder die Übung zu diesem Thema zu starten, wo man die Aufgaben mit Single-Choice oder Multi-Choice-Antworten bekommt. In

den gamifizierten Prototypen werden nach der Erledigung beide Teile-Theorieteil sowie Übungsteil abgehackt (siehe Abbildung 13).

Die Unterstruktur des theoretischen Teils eines zu erlernenden Themas wird in zwei andere Komponente verschachtelt: ein Video-Element und begleitender Text. Hier befindet sich also das Lernkomponent, welches aus verschiedenen Medienarten besteht.

Die Unterstruktur des praktischen Übungsteils eines zu erlernenden Themas wird in andere Komponente verschachtelt-Übungsaufgaben, die nacheinander gemacht werden können. In den gamifizierten Prototypen wird nach der Erledigung der Aufgaben nicht nur ein Häckchen gestellt, sondern werden auch Punkte/Sterne hochgezählt.

Die anderen Komponente, die in den gamifizierten Prototypen dargestellt werden, sind die Möglichkeiten, die monatliche Aktivität auf der mobilen Lernapp in der Art eines Diagramms zu kontrollieren. Sowie wird der Progress in der Art eines Kreises repräsentiert. Punkte nach dem Abschluss des Übungsteils werden auch hier gerechnet und visualisiert. Außerdem erhält man die Möglichkeiten, mithilfe von Chats in einem Forum mit anderen zu kommunizieren: bei irgendwelchen Fragen oder Unverständnissen kann man im Forum posten und kommentieren. Die Aktivität passiert anonym. Man erhält einen zufälligen Tiernamen per default oder kann einen beliebigen Namen selbst aufschreiben.

Die anderen Komponente werden sowohl in den gamifizierten Prototypen, als auch in den nicht gamifizierten Prototypen angeboten: Notifications, oder Benachrichtigungen über die Aktivität auf der Lernapp, Bookmarks für bookmarkte Videos und weitere Optionen, unter welchen die Möglichkeit existiert, die App mit anderen zu teilen.

Bei der Konzeption der Funktionalitäten werden die Ziele sowie entsprechende Mittel für den erfolgreichen Lernprozess berücksichtigt. Die Komponente, die unbedingt auf der Lernplattform umgesetzt werden sollen, sind solche Hauptkomponente wie z.B. Spracheinstellung, thematische Struktur mit der Übersicht von Kapiteln, sowie die Komponente jedes einzelnen Kapitels selbst - Theorie mit den Video- und Textelementen, sowie Übungselemente.

Die Elemente, die in den gamifizierten Prototypen, sowie in den nicht gamifizierten Prototypen angeboten werden, wie z.B. Bookmarks, Notifications und andere Optionen, sind keine obligatorische Elemente, sondern eher wünschenswerte. Diese werden basierend auf der Wettbewerbsanalyse entwickelt.

Die Gamification-Elemente, die in den gamifizierten Prototypen implementiert werden: Punkte, Progresskreis, Aktivitätskalender, Forum entsprechen den Achiever- und Socializer-Spielertypen. Deren Notwendigkeit wurde in den Interviews und Fokusgruppe entdeckt und somit bei der Entwicklung der gamifizierten Prototypen berücksichtigt.

3.5 Anforderungsliste

3.5.1 Nicht-funktionale Anforderungen

Nicht-funktionale Anforderungen bedeuten, wie das System bestimmte Anforderungen leistet. Nicht-funktionale Anforderungen sind normalerweise unspezifische Anforderungen und passen oft mehreren Produkten. Zu den nicht-funktionalen Anforderungen zählen (*Funktionale Anforderungen versus nicht-funktionale Anforderungen, o.J.; https://files_ifi_uzh_ch_rerg_amadeus_teaching_courses_spezifikation_und_entwurf_w_s0506_kapitel_1_o_j*):

- Effizienz, wie z.B. Genauigkeit von Berechnungen, Zeitverhalten, wenn zeitliche Verzögerungen beim Berechnen nicht sehr kritisch sein sollen, oder Resourcenverbrauch
- Zuverlässigkeit, wie z.B. Robustheit, IT-Sicherheit, Vertraulichkeit
- Benutzbarkeit, Gebrauchstauglichkeit, wie z.B. Bedienbarkeit, Verständlichkeit, Erlernbarkeit, Fehlertoleranz
- Änderbarkeit, wie z.B. Stabilität, Prüfbarkeit, Analysierbarkeit, Modifizierbarkeit
- Übertragbarkeit, wie z.B. Anpassbarkeit, Installierbarkeit, Konformität, Austauschbarkeit
- Funktionalität wie z.B. Angemessenheit, Richtigkeit, Interoperabilität, Ordnungsmäßigkeit, Sicherheit

Die Ergebnisse aus der Erhebung der Nutzeranforderungen sowie die Wettbewerber wurden analysiert und für die Erstellung einer Anforderungsliste buntzt. Die von uns anerkannte Prioritätsscala von 1 (unbedingt umsetzbar) bis 5 (nicht kritisch) vermittelt die Information darüber, welche Elemente unbedingt auf der Anwendung vorhanden sein sollen und welche eher erwünscht und nicht obligatorisch sind.

nicht-funkt. Anforder.	Pr.	Kommentare	Quelle
1. Effizienz	4	a. Das Mediamaterial, wie z.B. Videos oder Schemas, sollen nicht direkt embedded sein, sondern auf anderen Servern liegen und durch den link zur Verfügung gestellt werden, wie z.B. YouTube-Videos zu dem Lernmaterial sind durch einen Zugriff auf YouTube-Service verfügbar. Somit wird nicht sehr viel Speicherplatz benötigt. Das beeinträchtigt aber die Möglichkeit, die Anwendung in offline-Modus zu nutzen	Interview: 3. Person
1. Effizienz	4	b. Ausschaltbare Benachrichtigungen und andere Dienste somit ist Ressourcenverbrauch, wie z.B. Bluetooth, Geolocation, Akku gering	Interview: 3. Person

nicht-funkt. Anforder.	Pr.	Kommentare	Quelle
2. Benutz-bark.	1	Die Anwendung ist schnell erlernbar, denn sie verfügt über klare Übersichtsseite und thematische Aufteilung auf Kapiteln und Übungen. Somit ist die Struktur bei allen Themen gleich. Es wird keine Zeit verloren, um sich zu erinnern, wie man zu einer bestimmten Funktionalität noch mal kommen kann. Somit wird auf die Konsistenz, Struktur und intuitive Verständlichkeit der Nutzung geachtet	Fokusgruppe, Inter- views
3. Zuverlä-sigkeit	1	Da die Ergebnisse aus Fokusgruppe und Interviews zeigten, dass man auf Elemente der Rangliste, Leaderboard-Darstellung und des Wettbewerbs mit anderen eher ganz verzichtet, wird deshalb für die Nutzung der Anwendung keine Profil-Erstellung benötigt. Somit werden keine vertraulichen persönlichen Daten abgefragt und es gibt kein Risiko, dass die Daten verlieren gehen können. Die analysierten Wettbewerber vermitteln oft die Möglichkeit der Erstellung eines Profils. Auf unserer Anwendung wird selbst die Forum-Funktion als anonyme Funktion angeboten, durch z.B. Zuweisung eines zufällig generierten Tiernamens	Fokusgruppe, Inter- views

nicht-funkt. Anforder.	Pr.	Kommentare	Quelle
4. Änderbarkeit	1	sobald irgendwelche Aktivitäten auf der Anwendung durchgeführt werden, wird der aktuelle Stand gespeichert (die Überprüfung des Standes wird immer zu Beginn der Interaktion mit der Anwendung durchgeführt, sowie während der Interaktion mit einem bestimmten Zeirhythmus), damit man bei neuen Öffnen der Anwendung zu dem zuletzt abgeschlossenen Stand kommt und weiter fortsetzen kann. Somit ist nicht nur der Fortschritt zu merken, sondern wird jede Interaktion auf der Anwendung gespeichert	Interviews, Fokusgruppe

nicht-funkt. Anforder.	Pr.	Kommentare	Quelle
5. Übertragbarkeit.		<p>die Anwendung wird in zwei verschiedenen Layouts und somit Installationsdesign angeboten - als Webanwendung, welche durch einen Link benutzt wird, sowie als mobile Anwendung. Die Mehrheit der interviewten und befragten Personen bevorzugt den Lernprozess am Laptop. Mobile Nutzung ist für die Fälle, wenn man z.B. unterwegs ist. Das wird in zwei angebotenen Layout-Designs berücksichtigt. Die gleichen umgesetzten Features haben auch die analysierten Wettbewerber, wie z.B. "Codeacademy GO" oder "SoloLearn". Die Testphase schränkt sich aber in unserer Studie nur auf die Nutzung der mobilen Version. Der einzige kritischer Punkt ist, dass der Fortschritt und die Nutzung der Anwendung nur von Gerät abhängig. Somit bei der Reinstallation oder Installation auf einem Betriebssystem kann nicht mit vorherigen NutzerInnen synchronisiert werden</p>	Interviews, Fokusgruppe, Wettbewerbsanalyse
5. Übertragbarkeit.		<p>die regelmäßig angebotene Updates für beide Betriebssysteme für mobile Anwendungen (Android-System und iOS-System) sorgen für neue und erneubare Features</p>	Interviews, Fokusgruppe, Wettbewerbsanalyse

nicht-funkt. Anforder.	Pr.	Kommentare	Quelle
6. Funktionalität Spracheinstellungen			
6. Funktionalität	1	Die Information von den befragten Personen, die schon andere Programmiererfahrungen besitzen, über das, welche Elemente auf einer Lernplattform für Programmiersprachen erwartet wird, wie z.B. thematische Aufteilung, Kombination von Theorie und Praxis, Fortschrittskontrolle werden umgesetzt. Somit ist die erwartete Funktionalität für den Lernprozess der Programmiersprache unter der Berücksichtigung der Nutzeranforderungen vollständig	Interviews, Fokusgruppe, Wettbewerbsanalyse

3.5.2 Funktionale Anforderungen

Funktionale Anforderungen sind das, was das System leisten soll.

Funktionale Elemente sind Lösungskomponenten, wie z.B. Aktionen, die vom System ausgeführt werden sollen, oder Systeminteraktionen, die den NutzerInnen ermöglicht werden sollen. Diese Komponente sollen den BenutzerInnen sichtbar sein. Sie werden auf dem Bildschirm dargestellt und haben eine physische Repräsentation, wie z.B. eine Taste (Goodwin, 2009, S. 434).

Am Beginn soll eine Liste der Hauptelemente erstellt werden und danach sollen während des Zeichnens schrittweise weitere Details hinzugefügt werden. Es sollen aber noch die Lösungen im Kopf immer behalten werden, welche die Antworten auf die Fragen sind, wie eine umfassende Erkundung für eine neuartige Plattform durchgeführt wird (Goodwin, 2009, S. 435).

Allgemein ist die Lernanwendung ohne Gamification-Elemente so strukturiert, dass man nach der Öffnung der Anwendung auf dem Handy die Spracheinstellung machen soll. Per default ist die Sprache Englisch. Dann kommt eine Übersicht von allen Thematiken/Kapiteln. Wenn man sich ein beliebiges Thema frei aussucht, klickt man darauf und kommt zu einer Auswahlmöglichkeit - entweder Theorieteil anzuschauen (Video + Zusammenfassung) oder Testfragen zu üben.

funkt. der. Gamification- Elemente	Anfor- der. ohne	Pr.	Kommentare	Quelle
1. Spracheinstellung vor dem Beginn der Nutzung	1		auf Deutsch oder auf Englisch	Interviews
2. Allgemeine Struktur: Thematische Theorieauswahl in jedem thematischen Teil/Kapitel	1		theoretische Zusammenfassung vor dem Video	Fokusgruppe, Interviews, Wettbewerbsanalyse
2. Allgemeine Struktur: Thematische Übungsauswahl in jedem thematischen Teil/Kapitel	1			Fokusgruppe, Interviews, Wettbewerbsanalyse

funkt. der.	Anfor- der. ohne Gamification- Elemente	Pr.	Kommentare	Quelle
3. Unterstruktur des Theorieteils: Video-Tutorials	1			Fokusgruppe, Inter- views, Wettbe- werbs- analyse
3. Unterstruktur des Theorie- teils: Kurze theoretische Zu- sammenfassung vor dem Video	1			Fokusgruppe, Inter- views, Wettbe- werbs- analyse
3. Unterstruktur des Übungsteils: Übungen mit Single-Choice und Multi- Choice Antwort- smöglichkeiten	1	Codebeispiele werden veranschau- licht, Fragen werden gestellt		Fokusgruppe, Inter- views, Wettbe- werbs- analyse
4. Anderes, paral- lel: weitere Optio- nen, z.B. Teile die App mit Freun- den	5			Wettberwerbsanalyse

funkt. der.	Anfor- der. ohne Gamification- Elemente	Pr.	Kommentare	Quelle
4. Anderes. par- allel: Bookmarks, Kalenderaktivität	4		Sofern das Video für den Task da- für vorgesehen ist, kann es bookmarkt werden	Wettberwerbsanalyse
5. parallel: Feed- back von den Aufgaben	1		ständiges Feedback, wenn die Aufgabe falsch oder korrekt gelöst wird	Fokusgruppe, Inter- views, Wettbe- werbs- analyse

Für die Entwicklung einer Anwendung mit Gamification-Elementen werden folgende in den Interviews und Fokusgruppe diskutierten Unterhaltungelemente beachtet: Kommunikationselemente, die einen Socializer-Spielertyp zufrieden machen sollen (Marczewski, 2015; Bartle, 1996; Barata et al., 2014). Auf die Elemente, die den Interessen von Spielertypen Achievers und Killers entsprechen, wie z.B. Leaderboard, Wettbewerb oder Rangliste (Nacke et al., 2011; Xu, 2011), wird verzichtet, weil deren Notwendigkeit sich aus den Diskussionen in Fokusgruppe und Interviews nicht gezeigt hat. Die Elemente aus dem Belohnungssystem, die den Achievers und Killers auch noch interessant sein können (Nacke et al., 2011; Xu, 2011), wurden nicht von allen Befragten erwünscht, sind aber in unserem between-group-design Experiment in dem gamifizierten Prototyp umgesetzt. Insgesamt sobald Übungen erledigt werden, werden der ganze Lernfortschritt, prozentuell bezüglich allen Kapiteln, und entsprechende Anzahl an Sternen hochgerechnet.

funkt. forder.	An- mit	Pr.	Kommentare	Quelle
	Gamification- Elementen			
1. Spracheinstel- lung vor dem Beginn	1		auf Deutsch oder auf Englisch	Interviews
2. Allgemeine Struktur eines Kapitels: Thema- tische Theorie- auswahl in jedem thematischen Teil/Kapitel	1		theoretische Zusammenfassung vor dem Video	Fokusgruppe, Inter- views, Wettbe- werbs- analyse
2. Allgemeine Struktur eines Kapitels: Thema- tische Übungs- auswahl in jedem thematischen Teil/Kapitel	1		Sobald die Aufgaben geschafft wer- den, wird allgemeine Punktenanzahl berechnet und in der unteren Menü- leiste angezeigt. Die Punkte werden nie abzogen.	

funkt. forder.	An- mit	Pr.	Kommentare	Quelle
	Gamification- Elementen			
2.	Allgemeine Struktur eines Kapitels: Zertifikat-Ausstellung Fokusgruppe, Interviews, Wettbewerbsanalyse	2	Zertifikat kann ausgestellt werden, sobald alle thematischen Kapitel bestanden sind	Fokusgruppe, Wettbewerbsanalyse
3.	Unterstruktur des Theorieteils: Video-Tutorials	1		Fokusgruppe, Interviews, Wettbewerbsanalyse
3.	Unterstruktur des Theorie- teils: Kurze theoretische Zusammenfassung vor dem Video	1		Fokusgruppe, Interviews, Wettbewerbsanalyse

funkt. forder.	An- mit	Pr.	Kommentare	Quelle
	Gamification- Elementen			
3. Unterstruktur des Übungsteils: Übungen mit Single-Choice und Multi- Choice Antwort- smöglichkeiten	1		Codebeispiele werden veranschau- licht, Fragen werden gestellt.	Fokusgruppe, Inter- views, Wettbe- werbs- analyse
4. Anderes, par- allel: Forum, Fra- gen/Antworten	1		Die Kommunikations-Elemente auf der Lernplattform. Socializer- Gamification-Elemente	Fokusgruppe, Inter- views, Wettbe- werbs- analyse
4. Anderes, paral- lel: weitere Optio- nen, z.B. Teile die App mit Freun- den	5			Wettberwerbsanalyse
4. Anderes. par- allel: Bookmarks, Kalenderaktivität	4		Sofern das Video für den Task da- für vorgesehen ist, kann es bookmarkt werden	Wettberwerbsanalyse

funkt. forder.	An- mit	Pr.	Kommentare	Quelle
	Gamification- Elementen			
4. Anderes, par- allel: Systemfeed- back	1		Wenn die Aufgabe in den Übun- gen falsch gemacht wird, wird das als Fehler gemeldet und es werden Motivations- und Ansporn-Aufrufe angezeigt, Achiever-Gamification- Element	Fokusgruppe, Inter- views
4. Anderes, paral- lel: Progresskreis	1		Unten wird ein Progresskreis für die ganze Lernplattform dargestellt. Der Progress lässt sich aus den erledigten Kapiteln berechnen (Übungen werden alle erledigt). Achiever-Gamification- Element	Fokusgruppe, Inter- views
4. Anderes, paral- lel: Benachrichti- gung, Erinnerun- gen	3		Die Benachrichtigung kann man aber ausschalten, sobald man auf die Be- nachrichtigung klickt und diese aus- schaltet	Fokusgruppe, Inter- views
4. Anderes, paral- lel: Sterneanzahl	1		Sobald der ganze Kapitel abgeschlos- sen wird, werden Punkte-Sterne hochgezählt. Achiever-Gamification- Element	Fokusgruppe, Inter- views, Wettbe- werb- analyse

4 Konzept

Es werden 2 high-fidelity Prototype für die Anwendung erstellt, basierend auf den erhobenen Nutzerdaten und Anforderungsspezifizierung. Das erste Prototyp zielt einem nüchternen Design ab, ohne Spielelemente. Das zweite Prorotyp entspricht den Bedürfnissen und Erwartungen der Personas 1 und 3 und hat implementierte Gamification-Elemente. Für die Erarbeitung der möglichen Gamification-Elemente werden 3 Spielertyp-Taxonomien berücksichtigt: von Marczewski (2015), von Barata et al. (2014) und von Bartle (1996). Verschiedene Spielertypen werden verschiedene Spielmechaniken zugewiesen. Der Bedarf nach Gamification-Elementen wurde früher in den konzipierten Interviews und Fokusgruppe, sowie in Wettbewerbsanalyse analysiert, was für eigenes Entwicklungsvorgehen berücksichtigt wird. Die Studie wird als within-groups-design Experiment durchgeführt, indem einer zufälligen Hälfte der Probanden die entwickelte Anwendung zuerst als gamifiziertes Prototyp zum Testen angeboten wird und danach als nicht-gamifiziertes Prototyp. Der anderen Hälfte der Gruppe wird zuerst nicht-gamifiziertes Prototyp zum Testen angeboten und danach gamifiziertes Prototyp. Die Auswertung von NASA-TLX Questionnaire soll unsere Vermutung beweisen oder ablehnen, ob gamifizierte Anwendung bessere Interaktionserfahrung verspricht.

4.1 Apparat

4.1.1 *Spielertyp und Gamification-Elemente*

In der Forschung von Mensch-Computer Interaktion soll Verhaltensmotivation analysiert werden (Tondello et al., 2016). Basierend auf der Selbstbestimmungstheorie wird die Ausprägung von innerer und äusserer Motivation geforscht (*Coding Bat*, o.J.). Die innere Motivation ist mit individueller Wahrnehmung der Herausforderung als einer angenehmen Aufgabe verbunden. Die innere Motivation besteht aus mehreren Komponenten-Kompetenz mit Kenntnissen für die Aufgabelösung, Autonomität mit der Kontrolle über die Situation, Zugehörigkeit mit Gefühl in den Prozess zusammen mit anderen Menschen involviert zu sein und Zielorientierung

mit dem Überwinden von uninteressanten, aber wichtigen Aufgaben, die letztendlich zu einem Gefühl der Zufriedenheit führen (Tondello et al., 2016, S. 230).

Die äussere Motivation ist von äusseren Faktoren abhängig, wie z.B. erwartete Belohnung nach dem erledigen der Aufgabe (Tondello et al., 2016, S. 230).

Da jede Person verschiedene Motivationen hat, gehört sie dementsprechend verschiedenen Spieltypen. Somit besitzen verschiedene Spieler verschiedene Spielprioritäten (Tondello et al., 2016, S. 230). Die Analyse der Person hilft bestimmte personalisierte Funktionalitäten in einem spielerischen Kontext zu entwickeln.

Eine der Taxonomien, die Spielertyp kategorisiert, ist Bartle's Taxonomie. Es werden 4 Type der Spieler ausarbeitet (Xu, 2011): für Achievers sind Ziele innerhalb des Spiels wichtig wie z.B. Menge an Punkten, Geld oder Spiellevels zu erreichen. Achievers wollen die Aufgabe schnell und/oder komplett erledigen (Webb, 2013, S. 611).

Explorer sind Forscher oder Pfadfinder, die gerne virtuelle Welt erkunden und die Spielmechanik verstehen wollen (Xu, 2011). Explorer können stundelang ins Spiel eingetaucht werden und bevorzugen gar endlose Spielarten (Schneider et al., 2017).

Socializers nutzen das Virtuelle, um sich gerne mit anderen Spielern und Kommilitonen zu unterhalten. Sie knüpfen gerne neue Kontakte, schliessen sich den Communities an und sind auf der Such nach neuen FreundInnen (Webb, 2013).

Killers bleiben höchst zufrieden, wenn sie Angst oder "Schmerz" im Spiel verursachen oder Gegner vernichten. Dieses Bartle's Spielermodell vermittelt Grundverständnis über die Motivation der Spieler (Xu, 2011). Unmittelbare Wettbewerbe, Konkurrenz und Gewinnwünsche sind prägende Merkmale dieser Spielerart (Webb, 2013).

Das Bartle's Modell hat einige Limitierungen beispielsweise Zugehörigkeit nur einem bestimmten Spielertyp innerhalb der Taxonomie (Barata et al., 2014, S. 18). Deshalb werden andere Taxonomie entwickelt, wie z.B. die Taxonomie von Marczewski (2015) oder von Barata et al. (2014), die die Überlappungen der Spielereigenschaften und Motivationsfaktoren zulassen.

Marczewski (2015) entwickelt eine ähnliche zu Bartle's Taxonomie, die die inner-

lichen und äußerlichen Motivationen berücksichtigt.

Philanthropists finden das Motivationskomponent Zweck und dessen Erreichen ohne einen Belohnung zu erwarten wichtig. Socializers sind von Zugehörigkeitsgefühlen motiviert. Free Spirits schätzen die Autonomität wert, indem sie die Kontrolle über ihre Freiheit haben wollen, ohne einen Druck über sich selbst zu fühlen. Achievers finden ihre Kompetenzen entscheidend, die sie während des Lösen von Aufgaben aufzeigen können. Players sind diejenigen, die alles mögliche machen wollen, um äußerliche Belohnungen in der Art von Systemantwort zu erhalten. Disruptors genießen die Änderungen, indem sie negative oder positive Änderungen im System erwarten, nachdem sie irgendwas im System zerstören oder was gutes tun Tondello et al. (2016).

Eine andere Taxonomie, die von Barata et al. (2014) als gültige vorgeschlagen wird, ist das BrainHex Modell, welches sich für seine Archetypen von neurobiologischer Forschung und Diskussionen über Spielmuster und Spileemotionen inspirieren lässt (Nacke et al., 2011, S. 2). Das Modell besitzt noch keine validen neurobiologischen Techniken, sondern eher die Fragebögen (Nacke et al., 2011, S. 2), was in unserem Projekt die Interviews mit den Fokusgruppen und die Erarbeitung von Personas und Nutzeranforderungen als methodologische Mittel sind.

Hier werden die Kategorien des BrainHex Modells erwähnt, die in unserer Studie genauso berücksichtigt werden.

Seeker ist ein Suchende, dessen Aktivität durch einen Interessenmechanismus geweckt wird, der sich auf die Gehirnregion bezieht, die sensorische Informationen und Gedächtnisassoziationen verarbeitet. Reiche an den Interpretationen Spielmomente verursachen Produktion des Endomorphins und Aktivierung des Lustzentrums. Seeker genießen das Entdecken von virtueller Welt und Momente des Erstaunens (Nacke et al., 2011, S. 3). Die Beschreibung von Seekern ist ähnlich den Explorern aus der Bartle's Taxonomie.

Survivor genießt gerne Erfahrung, die nach dem Überleben von hohen Risiken und Terror-ähnlichen Momenten entstehen. Epinephrin als Hormon der Erregung und Dopamin, wenn Belohnungen erhalten werden, aktivieren den Angstzentrums und lösen den Schreckenzustand aus. Angstgenuss und die danach folgende Er-

leichterung prägen diesen Spielertyp (Nacke et al., 2011, S. 3).

Daredevil ist wie ein Teufelskerl, begeht gerne Risiken und hat immer Lust auf die Jagd oder Nervenkitzel. Hohe Geschwindigkeit, schwindelerregende Plattformen, wenn Daredevil die Kontrolle über Situation mit hoher Konzentration behalten soll. Epinephrin-Hormon verstärkt die Freude von den Belohnungen (Nacke et al., 2011, S. 2).

Mastermind, oder Kopfbrecher, genießt gerne das Erraten von Rätseln, Entwicklung von Strategien und Algorithmen. Entscheidungszentrum des Gehirns und die enge Beziehung zwischen diesem und dem Lustzentrum sorgt dafür, dass es sich lohnt, gute Entscheidungen zu treffen (Nacke et al., 2011, S. 3).

Conquerer, oder Eroberer, wollen nicht nur einfach gewinnen, sondern möglichst schwierige Feinden bekämpfen. Sie spielen aktiv, feurig und wütend. Epinephrin und Norepinephrin verursachen Erregung und Aufregung und machen die Emotionen wütender. Beharrlichkeit und Hartnäckigkeit sind die Merkmale von diesem Spielertyp (Nacke et al., 2011, S. 3).

Socializer sind wieder den Socializers aus der Bartle's Taxonomie ähnlich. Sie helfen den anderen gerne und kommunizieren mit den Leuten in ihrer Umgebung. Oxytocin ist der verantwortliche Neurotransmitter, der den Aufbau der Vertrauensgefühle fördert (Nacke et al., 2011, S. 3).

Achiever wie in der Bartle's Taxonomie sind Leistungsträger, sind zielorientiert, langfristig tätig. Die Abgrenzung zwischen Conquerer und Achiever ist sehr verschwommen und liegt darin, dass Achiever gerne die Spiele spielen, die unbedingt endlich sind, und zeigen eine obsessive, zwanghafte Fixierung auf das Erreichen von Zielen wie ein "Abhauen" und "Ankreuzen" von gelösten Aufgaben. Conquerer dagegen meistert sich selbst die Herausforderungen oder geht gerne auf die Suche von neuen Herausforderungen (Nacke et al., 2011, S. 3).

Es lohnt sich zu erwähnen, dass die NutzerInnen nicht ausschließlich einem bestimmten Motivationscluster gehören, sondern eher die Tendenz zeigen, bestimmte Motivationen zu haben. Außerdem zeigen die Studien unterschiedliche Ergebnisse, welche Taxonomie gültiger ist, als die anderen (Tondello et al., 2016; Jia et al., 2016, S. 232).

Für alle möglichen Spielertypen werden prägende Gamification-Elemente entwickelt (Friedrich et al., 2020; Xu, 2011). Einige haben Überlappungen und werden deshalb für mehrere Spielertypen gruppiert.

Für das Erledigen der Ziele können den Achievers, Killers, Explorers und Socializers folgende Gamification-Elemente helfen (Xu, 2011):

- Information, die für das Erreichen des Levels freigegeben wird
- Belohnungen, Rewards für erledigte Aufgabe
- Leistungen als Belohnungen
- Lotterie, Zufallsauswahl

Für das Erledigen der Zielen können den Achievers, Killers und Explorers folgende Gamification-Elemente helfen (Xu, 2011):

- Levels, die mit bestimmten erreichten Zielen freigegeben werden
- Punkte, die für bestimmte erledigte Herausforderung hochgezählt werden
- Leistungen, die wie ein Abzeichen oder Badges repräsentiert werden
- Quests, Herausforderungen mit eingeschränkter Zeit, die als eine Menge an Hindernissen umgesetzt werden und das Bemühen der SpielerInnen steuern
- Rewards-Tabellen in einem bestimmten Zeitrahmen

Für das Erledigen der Zielen können den Achievers und Killers folgende Gamification-Elemente helfen (Xu, 2011):

- Progress, der als Progressleiste dargestellt wird

Socializer-Spielertyp auf einer e-Learning Plattform kann zufrieden werden, wenn seine sozialen Bedürfnisse durch folgende Spielelemente befriedigt werden (Gil et al., 2015):

- Team/Community erstellen oder sich anschließen
- Ein Abzeichen oder soziales Status wie z.B. "most cooperative", "most participative", "most help-ready" erhalten

Die Motivation von Explorer kann durch folgende Mechanik erhöht werden (Gil et al., 2015):

- Ein Abzeichen wie z.B. "explorer badge", "adventurer badge" erhalten

Achiever werden durch folgende Mechanik zufrieden (Gil et al., 2015):

- "mastery certificate", "level-X expertise badge" erhalten für bestimmte erleidige Aufgaben

Für die Motivationen von Players aus Marczewski's Taxonomie werden solche Elemente umgesetzt, wie z.B. (Tondello et al., 2016):

- Punkte
- Belohnungen und Preise
- Abzeichen, Badges
- virtuelle Ökonomie
- Lotterien
- Leistungen

Mögliche Gamification-Elemente für Philanthropists sind (Tondello et al., 2016):

- Sammlungen
- Geschenke
- Erfahrungen teilen
- Administrative Rollen

Für die Free Spirits können folgende Design-Elemente motivierend sein (Tondello et al., 2016):

- Erkundung-Aufgaben
- nicht lineare Spielzeit
- Kreativitätsinstrumente

Einige ForscherInnen strukturieren die Spielmechanik nach den Komponenten der inneren Motivation. So können für die Spielertypen, für die ihre Autonomie wichtig ist, folgende Elemente umgesetzt werden: Profile, Avatare, anpassbare Interfaces, Kontrolle des Privatbereiches und Kontrolle über die Benachrichtigungen, Progressverlauf, Belohnungen, klare Ziele (Aparicio et al., 2012; Jia et al.,

2016; Khaleel et al., 2016). Die Spieler, die ihre Kompetenzen zeigen wollen, freuen sich über Punktesammlung, Levels-Erledigung, Leaderboard-Darstellung, positives Feedback vom System, optimale Hearusforderungen und intuitive Kontrollmöglichkeiten. Diejenigen, bei denen das Zugehörigkeitsgefühl ausprägend sind, schätzen Mechaniken wie z.B. Nachrichtenaustausch, Blog, Gruppenbildung, Mitgliedschaft, Chat-Teilnahme oder die Nutzung von sozialen Netzwerken (Aparicio et al., 2012).

Die Forschung der Motivationsfaktoren grenzt 6 Perspektiven aus, die für die Entwicklung des Gamification-Konzeptes wichtig sind. Die Perspektiven widersprechen einander nicht, konzentrieren sich aber auf die verschiedenen Komponenten (Sailer et al., 2013, S. 33). Jede Perspektive gibt ein gutes Verständnis für die Praxis und somit darauf wie die Gamification-Elemente in welchen Situationen die Motivation beeinflussen.

Aus der Perspektive der Charakterzüge sind persönliche Charakteristiken stabil über die Zeit hinweg und beeinflussen dadurch das Verhalten, wie z.B. Leistungsorientierung, Machtbedürfnis und Zugehörigkeitsbedürfnis. Diese Motivationsfaktoren und Eigenschaften der Spieler sollen berücksichtigt werden, wenn es um die Entwicklung in einem Gamification-Kontext handelt (Sailer et al., 2013).

Verhaltensbasierte Perspektive beachtet negative und positive Vergangenheitsergebnisse, die das zukünftige Verhalten beeinflussen können. Somit soll analysiert werden, welche Stimuli welche Antwort verursachen (Sailer et al., 2013).

Die kognitive Perspektive bezieht sich auf die Abhängigkeit zwischen Motivation und situation-spezifischen Zwecken, Erwartungen der Situationsergebnissen und Situationseinschätzungen (Sailer et al., 2013).

Die Perspektive der Selbtsbestimmung wurde schon früher erwähnt und basiert auf den Komponenten der Autonomität, Kompetenzen und Zugehörigkeit (Sailer et al., 2013).

Die Perspektive des Interesses setzt die Interaktion mit der Umgebung voraus. Das bedeutet, die Person erreicht komplettte Immersion mit der Aufgabe und taucht in den so genannten Flow ein (Sailer et al., 2013).

Die Emotionale Ebene kann von Strategien beeinflusst werden, zum Beispiel ist die

Motivation höher wenn positive Emotionen wie Sympathie oder Zufriedenheit ausgelöst werden (Sailer et al., 2013). Dementsprechend werden auch die Gamification-Elemente angepasst (Sailer et al., 2013, S.32-36).

Alle Taxonomie haben den Anspruch auf die Gültigkeit. In unserer Studie wird davon ausgegangen, dass es nicht nur auf eine Taxonomie eingeschränkt wird, sondern allgemein über ähnliche Spielertypen gesprochen wird.

Man kann gut erkennen, dass die Spielertypen aus verschiedenen Taxonomien auch gleiche Spielmechaniken motivierend finden. Wie z.B. Socializers aus der Taxonomie von Bartle (1996), aus der Taxonomie von Marczewski Marczewski (2015) und von Nacke et al. (2011). Oder Achievers aus der Marczewski's, BrainHex und Bartle's Taxonomien. Oder Free Spirits aus der Marczewski's Taxonomie, Seeker aus dem BrainHex Modell und Explorer aus der Bartle's Taxonomie.

Bei der Entwicklung der Anwendung ist Persona ein grundlegender Baustein, um die Nutzerprofile und Anforderungen verstehen zu können. Für Gamification ist Persona ein genauso wichtiges Mittel, um herauszufinden, was die Ziele der NutzerInnen und was die Motivationsfaktoren sind (Webb, 2013, S 612). Die Personalität der potenziellen NutzerInnen erklärt ihre Auswahl von Spielgenres und Gamification Elementen (Tondello et al., 2016, S. 230).

E-Learning Systeme sind gute Anwendungsfälle für Spiel-Elemente, weil sie eine Vielzahl an Funktionalitäten beinhalten wie z.B. Lernprogresskontrolle, die gut mithilfe von Gamification-Elementen unterstützt werden können, um die Motivation für deren Nutzung und für den aktiven Lernprozess zu unterstützen (Glover, 2013).

Somit können die Entwickler passende Spielmechaniken in die Anwendung implementieren.

4.2 Entscheidung für eine mobile Applikation

Nach reichlicher Überlegung hat sich unsere Gruppe dazu entschieden, eine mobile Applikation zu entwickeln. Hierzu liegen mehrere Gründe vor:

4.2.1 Entwicklung mobiler Application

Die Entwicklung unserer Mid- und Highfidelity Prototypen erfolgte über Axure Pro 10 RP, eine Software, die für die schnelle Entwicklung von Prototypen verwendet wird. Entsprechend der Anforderungsspezifizierung wurden die Elemente, je nach Version ("gamified" und "non-gamified") in der Entwicklung umgesetzt. Im Mid-prototype wurde zunächst die Grundlegenden Funktionalitäten unserer App implementiert, d.h. es wurden die vorgesehenen Seiten der App umgesetzt (Kapitel, Übungen, Kapitelübersicht, Fortschritt, Forum, Kalender, Übersicht der vermerkten Videos sowie die Anzahl der erhaltenen Sterne in der gamifizierten Version), welche dann mit Interaktionsmöglichkeiten (Click und Tap) versehen worden sind. In der gamifizierten Version wurden Ereignisse implementiert, die nach richtiger Beantwortung der Frage getriggert wurden. In der High-Fidelity Phase wurde die Applikation grafisch noch einmal überarbeitet, sowie die Interaktionslogik, damit eine korrekte Darstellung der App gewährleistet wird. Die fertigen Versionen der App konnten anschließend im Browser getestet werden.

4.2.2 Interaktion auf der Lernplattform

Die Interaktion auf der App erfolgt durch zwei simple Eingabemöglichkeiten: Rechts- bzw. Linkswischen und Antappen. Das Wischen ist für die Betrachtung der Kapitel sowie der Unterkapitel notwendig, in denen sich Codebeispiele befinden und die Theorie dahinter erläutert wird. Das Antappen ist für die Navigation und das Lösen der Aufgaben erforderlich. Da die Aufgaben des Übungskapitels aus Single Choice Fragen bestehen, muss einer der gestellten Antworten angetippt werden. Nach der richtigen Beantwortung der Frage erscheint ein Button der zur nächsten Frage führt. Sollte die Frage übersprungen werden, so kann auf den "Vorwärts"-Button rechts oben im Bildschirm getippt werden. Es öffnet sich ein Fenster in dem der Nutzer gefragt wird, ob er die gegenwärtige Fragestellung überspringen möchte. Wird auf "Nein" getippt, schließt sich das Fenster, wird stattdessen auf "Ja" getippt, wird zur nächsten Frage gewechselt. Sollte der Nutzer die Frage richtig beantworten, erscheint kurzerhand ein Button unter den Antwortmöglichkeiten, mit dieser zur nächsten Frage gewechselt werden kann. Wenn die Übungen abgeschlossen sind,

erscheint wieder Button auf der Abschlussseite, welcher wieder zur Übersicht aller Kapitel führt. Alle Seiten der App verfügen über einen "Zurück" Button, mit welchen zur vorherigen Seite gewechselt werden kann.

Bookmarks lassen sich mithilfe von Axure nur eingeschränkt implementieren. Bookmarks können gesetzt werden, allerdings ist für eine Übersicht der vermerkten Videos eine Speicherung und zusätzliche Programmlogik notwendig, die die Funktionalität von Axure nur bedingt zu lässt. Deshalb wird das Bookmarke folgendermassen "simulieren": nachdem die Probanden das Video im Lernmaterial für entsprechenden Kapitel finden, können sie dieses Video "bookmarken". Bookmarkt wird dieses Video erst dann dargestellt, wenn die Probanden die Übungen abschliessen und bei der Übersicht aller Kapitel dann das bookmarkte Video finden kann.

Einige Features, wie Bookmarks, Forum, Teile App mit Freunden, Benachrichtigungen, lassen lediglich "simulieren" d.h. es werden entsprechende Buttons implementiert, die jedoch keine besondere Interaktion bieten können.

Auch die Implementierung eines Zertifikats für die gamifizierte Version konnte nicht umgesetzt werden. Die hierzu nötigen Funktionalitäten lassen sich mit Axure leider nicht umsetzen.

4.3 Studiendesign

Die Studie wird als within-groups-design Experiment remote oder auch live durchgeführt. Die unabhängigen Variablen sind zwei unterschiedliche Ansätze: der Prototyp mit Gamification-Elementen und das Prototyp ohne Gamification-Elemente.

Die abhängigen Variablen sind die Ergebnisse aus dem NASA-TLX Questionnaire, welches die ProbandInnen in der Experimentgruppe im Anschluss der Interaktion auf der Lernplattform erhalten. Die somit erfassten Erfahrungen von zwei konkurrierenden Ansätzen werden miteinander verglichen, um herauszufinden, welches Design eine bessere Nutzererfahrung verspricht.

4.4 Studie

Allen Probanden in unterschiedlicher Reihenfolge wurden 2 Aufgaben gegeben. Beim Testen des gamifizierten Prototyps sollen die Probanden folgende Aufgabe lösen: "Machen Sie die ganze Übung aus dem zweiten Kapitel und finden Sie danach heraus, wie viele Sterne Sie dafür erhalten und was Ihr aktueller Fortschritt auf der Lernplattform ist". Die erwartete Vorgehensweise war, dass die Probanden empirisch verstehen, dass der zweite Kapitel erst dann freigeschaltet wird, sobald man den ersten Kapitel mit der "Hello World"-Aufgabe durchführt. In der Übung im zweiten Kapitel hat man die Möglichkeit bestimmte Aufgaben zu überspringen und somit schneller ans Ende der Übung kommen. Gleich nach dem Abschluß der Übung aus dem zweiten Kapitel erreicht man die Overview-Seite, bei welcher die Anzahl an erhaltenen Sternen in unterer Menüleiste repräsentiert wird, sowie der Progresskreis bezogen prozentuell auf die gesamte Anzahl der Kapitel. Somit ist die Aufgabe abgeschlossen.

Die Aufgabe für Usability-Testen am nicht gamifizierten Prototyp ist folgende: "Finden Sie das Video im Lernmaterial des zweiten Kapitels und bookmarken Sie dieses Video. Danach gehen Sie zu der Overview-Seite und finden Sie dieses bookmarkte Video heraus". Der einfache Lösungsvorschlag wäre gleich zu dem zweiten Kapitel zu navigieren, ohne vorher der ersten Kapitel "Hello World" anzuschauen, was möglich ist, denn der nicht gamifizierte Prototyp volle Flexibilität der Navigation auf der Lernplattform ermöglicht. Denn dieser nicht gamifizierte Prototyp soll keine spielerische Elemente beinhalten. Somit ist Freischalten von Kapiteln als Achiever-Element nicht nötig.

Sobald man das Video im zweiten Kapitel rausfindet und dieses bookmarkt, soll man noch die ganze Übung erledigen und erst danach zu der Overview-Seite kommen. Da wird das "Bookmarks"-Zeichen dargestellt, unter welchem man das bookmarkte Video finden kann. Somit gilt die Aufgabe erledigt.

NASA-TLX Questionnaire ist eines der bekanntesten Bewertungstools der subjektiven Arbeitsbelastung. Dieser Fragebogen zeigt sich als ein valides und robustes Maß für die subjektive Arbeitsbelastung (Noyes & Bruneau, 2007). Die Vergleiche der Sensitivität zwischen dem NASA-TLX Questionnaire und anderen subjektiven

Messtools und Messgrößen zeigen eine hohe Zuverlässigkeit, Benutzerakzeptanz und geringe individuelle Variabilität (Ramkumar et al., 2017). Insgesamt ist es eine Scala mit 21 Punkten. Je kleiner der Wert in dem Fragebogen ist, desto geringer ist die entsprechende Belastung. Höhere Bewertungen entsprechen der höheren Arbeitsbelastung.

NASA-TLX Questionnaire beschreibt 6 Belastungskategorien: Mental Demand, Physical Demand, Temporal Demand, Performance, Effort, Frustartion. Mental Demand ist die kognitive Belastung bei der Interaktion auf der Lernplattform. Physical Demand entspricht der körperlichen Belastung. Temporal Demand beantwortet die Frage, ob man bei der Interaktion einen Zeitdruck spürt. Performance spiegelt den Erfolg und die Effektivität der Erledigung von der Interaktionsaufgaben wieder. Effort entspricht dem Schwierigkeitsgrad der Interaktion und der somit erlebten Erfahrung. Frustration bedeutet Frustrationsgrad und Empörungsgrad, wenn irgendwas nicht intuitiv verständlich ist und somit schwierig zu erledigen ist. Das sind die Emotionen, die man verspürt, wenn bei der Interaktion irgendwas nicht klappt.

4.5 Teilnehmer

Die TeilnehmerInnen für die Studie wurden innerhalb des Medieninformatik-Lehrstuhls der Universität Regensburg über ein gemeinsames Forum und über Soziale Netzwerke erreicht. Es nahmen insgesamt 6 Probanden teil. Alle Probanden waren Studierende der Universität oder OTH Regensburg.

Die Bedingung für die Teilnahme waren Kenntnisse in Python auf einem Programmieranfängerniveau oder gar keine Vorerfahrungen in den Programmiersprachen. Als Gegenleistung für die Teilnahme an der Studie erhielten Studierende bis zu einer Versuchspersonenstunde für die Studiengänge Medieninformatik oder Informationswissenschaft.

Die Probanden wussten im Voraus nicht, was sie genau testen sollten. Es wurde ihnen nur mitgeteilt, dass es bei der Studie um die Interaktion mit einer Lernanwendung geht.

4.6 Prozedur

Die Probanden sollen das Programm Axure herunterladen und das entsprechende Projekt starten. Der Vorbereitungsprozess wird von den StudienleiterInnen unterstützt.

Die ProbandInnen starten das Projekt und werden gleich zu dem Browser weitergeleitet. Den 3 zufällig ausgewählten TeilnehmerInnen wird zuerst der gamiziertes Prototyp zum Testen angeboten und danach der nicht-gamifiziertes Prototyp. Den anderen 3 ProbandInnen wird zuerst der nicht-gamiziertes Prototyp angeboten und danach der gamifiziertes Prototyp.

Die Studie richtet sich an die Zielgruppe der ProgrammieranfängerInnen oder derjenigen, die keine Erfahrungen mit den Programmiersprachen haben, insbesondere nicht mit Python, worauf vor der Studienteilnahme als eine Bedingung hingewiesen wird.

5 Ergebnisse

Im Folgenden wird auf die Ergebnisse aus der Studie eingegangen. Als erstes werden die Ergebnisse aus dem NASA-TLX Questionnaire der Studien-TeilnehmerInnen zu den beiden Prototypen einander gegenübergestellt. Darauffolgend werden die Kommentare der StudienteilnehmerInnen während der Interaktion betrachtet und vorhandene Wünsche und Änderungsvorschläge wiedergegeben. Danach werden die Ergebnisse ausgewertet und die Werte für den nicht-gamifizierten und gamifizierten Prototypen miteinander verglichen.

5.1 NASA-TLX Questionnaire

Nachdem die Studie durchgeführt wurde, ergeben sich folgende Werte aus den NASA-TLX-Fragebögen für gamiziertes Prototyp (siehe Tabelle "gamiziertes Design"):

Tabelle für **gamiziertes Design**:

Number	Mental	Physical	Temporal	Performance	Effort	Frustration
	Demand	Demand	Demand			
1.	7	5	1	3	5	8
2.	4	1	3	2	4	2
3.	6	3	1	3	6	2
4.	4	2	2	4	7	4
5.	16	1	3	11	4	2
6.	3	1	1	5	4	1

Die Werte aus den NASA-TLX Questionnaire für nicht gamiziertes Design sind folgende (siehe Tabelle "nicht gamiziertes Design"):

Tabelle für **nicht gamiziertes Design**:

Number	Mental	Physical	Temporal	Performance	Effort	Frustration
	Demand	Demand	Demand			
1.	7	7	1	4	6	7
2.	3	1	6	15	6	15
3.	6	2	1	4	7	5
4.	4	1	1	5	7	5
5.	19	1	4	14	5	2
6.	3	1	1	6	4	3

Auf die Gewichtung von NASA-TLX Werten wird verzichtet, weil es nicht nur zeitaufwändig ist, sondern auch laut sämtlichen Studien unterschiedliche, umstrittene Ergebnisse über die Validität der Gewichtung zeigt (*10 Things to Know about the NASA TLX*, o.J.). Somit wird "Raw NASA-TLX" Questionnaire verwendet.

5.2 Kommentare von der Interaktion

Die Personen teilen meistens ihre guten Eindrücke von der Interaktion auf der mobilen Lernplattform, wie z.B. "schön, sieht so aus, wir bei der Plattform Duolingo, welche ich ein mal schon benutzt hat".

Die Plattform wurde als übersichtlich bezeichnet, es wurde zum Beispiel positiv angemerkt, dass sie "nicht überladen" wirkt. Sie wurde u.a. als "intuitiv" bezeichnet aufgrund ihrer sehr simplen Navigation.

Eine Person hat die Aufgabe für das nicht gamifiziertes Prototyp aufgehört, weil er nicht mehr weiter kommen konnte. Das war die Aufgabe, bei welcher man das Video bei der Lerneinheit herausfinden und bookmarken sollte. Der Proband meinte, es ist nicht intuitiv verständlich, wie man weiter zu dem Video kommen kann. Der Proband hat schon vertikales Scrolling ausprobiert und hätte erwartet, ein Bürgermenü, wo das Video liegen könnte, oder andere Elemente für weitere Navigation zu finden. Man hat gemeint, es ist schwierig, eine mobile Application im Browser zu testen, weil die Sensoren und Interaktionsmechanismen nicht miteinander auf verschiedenen Geräten übereinstimmen.

Beim Testen des gamifizierten Prototyps wurde sich von mehreren ProbandInnen gewünscht, eine Begründung für eine falsche Antwort zu bekommen. Dies wird als

nötig angesehen, um zu verstehen, was man falsch gemacht hat. Bei der Erklärung könnte auch ein Beispiel beigelegt sein. Auch bei dem richtigen Lösen einer Aufgabe wurde sich mehr Feedback gewünscht um genau verstehen zu können, weshalb eine Aufgabe richtig ist.

Ein Proband hätte die Kapitelübersicht lieber auf einer anderen Art und nicht als einer langen Kette dargestellt.

Eine Probandin, äußerte den Vorschlag, bei der gamifizierten Version die Belobigungen nach erfolgreichem Lösen der Aufgabe auditiv zu unterstützen, d.h. das ein Geräusch ertönen soll, wenn eine Aufgabe richtig gelöst wurde. Aus Erfahrung mit anderen Lernapplikationen empfand sie diesen Mechanismus als durchaus motivierend.

Ein weiterer Proband empfand die Vergabe von Sternen als Belohnung in der gamifizierten Version als "zu kindisch". Statt Sterne schlug er eine Art Progressbar vor, die mit Punkten anwächst. Dies empfand er als größeren Ansporn um Aufgaben (besser) zu lösen.

5.3 Auswertung

An der Studie haben insgesamt 6 Probanden teilgenommen. 3 zufällig ausgewählten Probanden wurden zuerst der gamifizierte Prototype zum Testen angeboten und danach der nicht-gamifizierte. Den anderen 3 Teilnehmenden wurden die Prototype in der anderen Reihenfolge angeboten.

Aus den NASA-TLX Questionnaires ergibt sich folgendes (siehe Abbildung 14: der Mittelwert der subjektiven Arbeitsbelastung laut NASA-TLX-Fragebogen erreicht in der Gruppe des gamifizierten Prototypen den Wert 23.5. Währenddessen liegt der Mittelwert in der Gruppe des nicht-gamifizierten Prototypen bei 30.67).

Die Werte für jede einzelne Kategorie des Fragebogens werden für den gamifizierten Prototype folgendermaßen verteilt (siehe Abbildung 15):

Es lässt sich merken, dass Mental Demand und Performance in dem gamifizierten Prototyp einen Probanden mehr überfordernd zu sein scheint, als andere Kategorien. Physical Demand, sowie Temporal Demand sind von keinem großen Einfluss auf die Teilnehmenden.

Die Werte für jede einzelne Kategorie des Fragebogens werden für den nicht gamifizierten Prototype folgendermaßen verteilt (siehe Abbildung 15):

Es lässt sich merken, dass die Werte für Mental Demand, Frustration und Performance in dem nicht gamifizierten Prototyp relativ höher sind, als andere Kategorien. Physical Demand, sowie Temporal Demand bleibt immer noch von relativ kleinem Einfluss auf die Teilnehmenden.

Da unsere Studie ein within-groups Experiment ist, wird somit t-Test für abhängige Stichproben durchgeführt (*Effektstärke Cohens d*, o.J.), um herauszufinden, ob zwei unterschiedliche Designgruppen sich voneinander unterscheiden. Es hat sich ergeben, dass es kein relevanter Unterschied zwischen den gamifizierten und nicht gamifizierten Prototypen existiert ($p=0.137$).

In dieser Grafik (siehe Abbildung 17) sind die Ergebnisse aus dem NASA-TLX Questionnaire in den sechs untersuchten Kategorien (Mental Demand, Physical Performance, Temporal Demand, Performance, Effort und Frustration) abgebildet.

Bei den einzelnen Kategorien zeigt der jeweils linke, schwarz-gefärbte Balken die Ergebnisse des nicht-gamifizierten Prototypens an, während der rechte, grau-gefärbte Balken die Ergebnisse des gamifizierten Prototypens zeigt. Man kann hier gut erkennen, dass die Werte für den nicht-gamifizierten Prototypen im Allgemeinen höher auf der Skala angesiedelt sind als die des gamifizierten Prototypens. Allerdings ist dieser Unterschied nicht sehr hoch und damit nicht statistisch relevant.

Aus dieser Grafik (Abbildung 17) lässt sich auf den ersten Blick vermuten, dass die Kategorien Performance, sowie die Kategorie Frustration sich in beiden Prototypen unterscheiden könnten. Das wird von uns überprüft, indem wiederum der t-Test für entsprechende Vergleichsgruppen durchgeführt wird. Es hat sich aber ergeben, dass es kein relevanter Unterschied der Kategorie Performance zwischen den gamifizierten und nicht-gamifizierten Prototypen entdeckt wurde ($p=0.150$). Gleichzeitig wird auch kein signifikanter Unterschied gefunden in den Vergleichsgruppen der Kategorie Frustration ($p=0.194$).

6 Summative Evaluation

6.1 Projektplan und Management

Nach der Gruppenfindung wurde sich auf das Projektthema "Teach Me Code" geeignet. Da beide Gruppenmitglieder zu Beginn des Projektes mit der Programmiersprache Python vertraut sind, war keine Einarbeitung notwendig. In einer kurzen Absprache wurde bereits grob der Zeitplan und Zeitaufwand des Projektes skizziert, unter Berücksichtigung des Terminplans der Gruppenteilnehmer. Anschließend wurden auf Github ein Sprint Backlog erstellt. Hier wurden die ersten Sprints für die ersten 3-4 Wochen erstellt. Zudem wurden erste Backlogs gesetzt, die die Erstellung der Interviews beziehungsweise Interviewfragen und die Rekrutierung von Interviewteilnehmern vorsah. Die Erstellung und Verständnis des Sprintbacklogs bereitete keinem der Gruppenmitglieder Probleme.

6.2 Interviews, Fokusgruppe

Die Fragen zu den Interviews und Fokusgruppe sollten maximal unterschiedliche mögliche Zielgruppen rausfinden. Unter denen sind die potenziellen NutzerInnen, die schon Vorkenntnisse in anderen Programmiersprachen oder auch in Python besitzen und sich durch die Erfahrung mit anderen Lernsystemen ausprägen. NutzerInnen können auch diejenigen sein, die noch nie programmiert haben und dadurch auch keine Erfahrung mit den E-Lernsystemen haben, gerne aber Programmieren lernen möchten. Um rauszufinden, ob unsere EndnutzerInnen tatsächlich so heterogen sein können, wurden gezielt die Fragen über ihre Vorkenntnisse und Vorerfahrungen beim Programmieren und Nutzen von Lernplattformen abgefragt.

Danach sollte beobachtet werden, welche Mittel, welche Lernart und in welchen Fällen die potenziellen Kunden nutzen können. So präferierten einige Programmieraufgaben mithilfe von Laptop und Tastatur zu lösen.

Danach wurden Fragen über individuelle Lernprozessvortstellungen gefragt, welche auch helfen sollten zu verstehen, ob befragten Personen sich schon mit dem Lernprozess des Programmierens auseinandergesetzt haben.

Was auch wichtig war, gleich abzufragen, welche Erfahrung man schon hat, falls

die befragte Person sich schon mit anderen Lernplattformen auskennt. Daraus konnten Erkenntnisse gezogen werden, welche Interaktionsprobleme schon getroffen wurden.

Abschließend wurden die Fragen bezüglich der präferierten Spielelementen gestellt, um ungefähr verstehen zu können, ob NutzerInnen sich den Lernprozess in einem spielerischen Kontext wünschen oder nicht. Die Fragen über alle möglichen Spielmechaniken sollten helfen, Motivationen und entsprechende Spielertypen zu definieren. Somit konnten erwünschte Spielelemente analysiert und auf die Lernplattform umgesetzt werden.

Sowohl in den Interviews, als auch in der Fokusgruppe ergaben sich vielfältige Ansichten bezüglich dieser Unterhaltungselemente. Entweder die befragten Personen fanden solche Elemente unnötig, oder umgekehrt als gute Unterhaltungsfeatures, welche ihnen Ansporn geben könnten, bessere Kontrolle über den Lernfortschritt oder auch andere Unterhaltungen ermöglichen, wie z.B. Möglichkeiten, mit anderen Lernenden zu kommunizieren.

Aus den Interviews und Fokusgruppe ergab sich auch noch, dass alle Probanden ausschließen wollen, in einer direkten Konkurrenz mit anderen zu stehen, wie z.B. Rangliste-Elemente mit unbekannten Personen.

Das wurde von uns berücksichtigt und auf solche Elemente wurde bei der Entwicklung der Lernplattform ganz verzichtet. Außerdem ergab sich keine Erkenntnis bezüglich der explorativen Elemente. Die befragten Personen tendieren zu einem endlichen Lernprozess auf der Lernplattform.

Alle Probanden schliessen sich der Meinung an, dass die Lernplattform strukturiert gebaut werden soll und dass diese Struktur auch große Freiheit und Flexibilität geben soll. Das betrifft die Auswahl der Thematik von Übungsmaterial und Lerneinheiten.

6.3 Wettbewerbsanalyse

Die Analyse von Konkurrenten ermöglichte besseres Verständnis von heutigen Problemfeldern. Es wurden 7 unterschiedliche berühmte marktführende Lernsysteme für Programmiersprachen ausgewählt, die nicht nur als mobile Applikationen, son-

dern auch als Webplattformen angeboten werden, um die Vielfalt von ihnen angeboten Funktionalitäten miteinander vergleichen zu können. Dadurch konnten Relationen zwischen den Erkenntnissen aus den Interviews und Fokusgruppe und aktuellem Marktangebot festgestellt werden. Die Analyse von Wettbewerbern beantwortete die Fragen, was und auf welche Art und Weise schon umgesetzt wurde, wie die Produkte bewertet und von den Kunden aktuell wahrgenommen werden. Ausserdem vermittelten diese Produkten Inspirationen für den Entwicklungsprozess unseres Produktes gemäß unseren Nutzergruppen und Nutzeranforderungen.

6.4 Personas, Use-Cases, User Stories, HTA

Die Erstellung von Personas vermittelte Repräsentation von realen Personen, basierend auf den Erkenntnissen aus den Interviews und Fokusgruppe. Personas ließen einzigartige Patterns und Muster potenzieller Kunden besser verstehen können, indem die Muster, Habitäten und Ziele der NutzerInnen detektiert, gruppiert, analysiert und als wichtige Merkmale gesetzt werden. Somit entstanden verschiedene Rolle der potenziellen Kunden, verschiedene Verhaltensvarietäten, verschiedene mentale Modelle und dementsprechend verschiedenes Mapping-Spektrum von den Variablen und potenziellen Kunden.

Personas widerspiegeln wichtige Eigenschaften und Patterns und somit mögliche Verhaltenssituationen. Die drei definierten Personas entsprechen den Hauptclustern von den befragten Personen: diejenigen, die sich gerne Unterhaltungsaspekt auf der Lernplattform vorstellen würden, und diejenigen, die sich für "nüchternes" Design entscheiden. Es sind in Personas auch widerspiegelt, dass die potenziellen Kunden entweder Vorkenntnisse in den Programmiersprachen besitzen oder überhaupt nicht. Somit können die potenziellen NutzerInnen entweder Studierende sein, oder auch nicht. Entweder die Studierenden von technischen Fächern, oder aus den anderen Studiengängen.

Die Arbeit an den Use-Cases hat mit einer Recherche von Quellen begonnen. Auf der einen Seite sollten die Grundlagen gefunden werden, um ein aussagekräftiges UML-Diagramm für die Teach Me Code App zu erstellen. Auf der anderen Seite soll innerhalb dieser Arbeit auch ausreichend erklärt werden, wobei es sich um

Use-Cases und dem UML-Diagramm handelt, und welche Bedeutung sie für den Entwicklungsprozess haben.

Grundlage für das UML-Diagramm zu der Teach Me Code App waren dabei die aus der Anforderungserhebung gewonnen Einsichten über den Aufbau und die mögliche Funktionsweise der App. Aspekte, wie dass man die Sprache ändern kann oder dass man auf seinen Lernfortschritt zugreifen kann, sollen in das UML-Diagramm integriert werden.

Bei der Erstellung des UML-Diagramms war der erste Schritt die Überlegung, welche Elemente für dieses konkrete System für Bedeutung sind. Dabei war schnell erkennbar, dass für die Teach Me Code App einige Elemente die für die Use-Cases relevant sein können nicht benötigt werden. Es ist kein Trigger vorhanden, der den Anwendungsfall initiiert, und keine relevanten Preconditions, also keine Konditionen, die erfüllt sein müssen um bevor es zu einer Anwendung kommt.

Letztlich wurde sich dafür entschieden, dass es einen Primary Actor gibt, der den Anwendungsfall initiiert und dadurch als Stakeholder fungiert und ein direktes Interesse am Verhalten des Systems hat. Das System ist in diesem Fall die Teach Me Code App an sich.

Innerhalb dieses Systems kann der Actor einige sich für einige unterschiedliche Aktionen unterscheiden, die er ausführen kann. Einige dieser Aktionen, wie zum Beispiel das Speichern von Lernvideos, können nur nach dem vorherigen Ausführen von anderen Aktionen, in diesem Fall das öffnen des Lernmaterials, erfolgen.

Es wurde sich entschieden, in dem UML-Diagramm zu der Teach Me Code App neben dem Primary Actor auch noch einen zweiten Benutzer aufzuführen. Mit diesem kann der Actor über das System interagieren, zum Beispiel in dem er in das integrierte Forum schreibt oder in diesem liest, oder indem er die App nach außen teilt.

Grundsätzlich soll das UML-Diagramm zu der Teach Me Code App sehr überschaubar und übersichtlich sein, um einen guten Überblick über die notwendigen Interaktionsmöglichkeiten mit der App zu schaffen und zu veranschaulichen, auf welche Weise die BenutzerInnen mit der App agieren können.

6.5 Design

Damit das Design von gamifizierten Prototypen entwickelt werden konnte, wurden vorerst die Interviews und Fokusgruppe durchgeführt, um bestimmte erwünschte Spielelemente zu entdecken. Die Erkenntnisse von den befragten Personen, sowie aus der Analyse von Konkurrenten zeigten, welche Elemente besonders wünschenswert sind und auf welche gar verzichtet werden konnte.

Die Fragen zu den Interviews und Fokusgruppe wurden basierend auf unterschiedlichen analysierten Spielmechaniken entwickelt. Unterschiedliche spielerische Elemente werden unterschiedlichen Spielertypen zugewiesen. Dabei sollten diese Spilertypen analysiert werden, um zu verstehen, welche Motivationsfaktoren welche Spielertypen besitzen.

Die Spielertypen und deren Charakterzüge wurden gemäß drei Taxonomien analysiert. Diese drei Taxonomie haben Ansprüche auf die Gültigkeit. Eine Taxonomie - Bartle's Taxonomie hat ihre Limitierungen, wie z.B. die eins-zu-eins Relation zwischen dem Spielertyp und potenziellen Kunden. Andere Taxonomien erlauben Überlappungen von den Motivationsfaktoren. Wichtig war, diese Taxonomien zu erlernen, sowie den Zusammenhang zwischen den Spielertypen und Spielelementen festzustellen.

Die Beachtung, dass die potenziellen NutzerInnen unterschiedliche Spielelemente und welche Spielelemente genau bevorzugen, ermöglichte das Festlegen von Präferenzen der Unterhaltungsaspekte auf der Lernplattform.

6.6 Erstellung der Prototypen

Die Erstellung der Prototypen erfolgte in drei aufeinanderfolgenden Stufen, dem Low-, Mid-, und Highfidelity-Prototype. Nach Abschluss der Anforderungspezifizierung wurde zunächst zum Low-Fidelity Prototype erstellt. Unter Präsenz der Gruppenmitglieder wurden die Seiten unserer App mit Papier erstellt. Während des Entwurfs berieten wir uns über Designspezifische Aspekte und der jeweiligen potenziellen Verlauf der Nutzung der App ähnlich den Use-Cases. Nach Erstellung der Papier-Prototypen sortierten wir diese und gingen über zur Videoaufnahme, in der wir die Nutzung der App simulierten. Zugleich kommentierten wir unsere

Interaktionen mit dem Prototypen.

Erwartungsgemäß handelte es sich um die einfachste Phase der Prototypen Erstellung. Der Entwurf und Verwendung der Papierprototypen erstreckte sich über einen Zeitraum von ca. 2 Stunden. Probleme oder Meinungsverschiedenheiten hinsichtlich der Konzeption des Entwurfs kamen nicht auf.

Nach dem der Low-Fidelity Prototype erstellt wurde, wurde zum Mid-Prototype übergegangen. Dieser wurde mithilfe von Axure, einer Software zur Ausarbeitung von Highfidelity Prototypen, erstellt. Die Entwicklung nahm hierbei die meiste Zeit in Anspruch. Nach einer kurzen Eingewöhnungszeit an die Software, sowie die Einarbeitung in die grundlegendsten Funktionen der Software, konnte erfolgreich der Mid-Fidelity Prototyp erstellt. Die Implementierung der Interaktionsmöglichkeiten erwies sich als recht einfach. Jedoch ließen sich bestimmte geplante Features nicht implementieren wie z.B. die Speicherung von Videos, die mit einem Lesezeichen versehen wurden. Stattdessen wurde die Interaktion lediglich visualisiert. Auch Features wie dem Kalender, dass hintereinander absolvierte Übungen visualisiert beziehungsweise anzeigt konnten nicht realisiert werden. Lediglich die Seite in dem sich der Kalender und die "Streaks" befanden, wurden angelegt. Die Entwicklung des Mid-Prototype wurde einem Gruppenmitglied aufgetragen. In dieser Entwicklungsphase wurden die beiden Versionen der App erstellt. Dementsprechend erhielt die zu gamifizierende Version die Gamification-Elemente. Insgesamt nahm die Entwicklung des Prototypen wenige Tage in Anspruch.

Nach ausgiebiger Testung des Mid-Prototype auf fehlende oder fehlerhafte Interaktionen etc. erfolgte die Aufwertung zum High-Fidelity Prototype. Hier übernahm ein anderes Gruppenmitglied die Aufgabe und überarbeitete die App noch einmal grafisch. Abermals wurde die App noch einmal getestet und von den anderen Gruppenmitgliedern begutachtet. Die grafische Überarbeitung bedurfte ebenfalls nur weniger Tage. Ihre Implementierung gestaltete sich ebenfalls recht einfach.

6.7 Ergebnisse

Ergebnisse aus den NASA-TLX-Questionnaire vermittelte das Verständnis, in wie weit unterschiedliche Ansätze der Lernplattformen attraktiv sind. Der Vergleich der

Attraktivität von der Nutzung zwischen dem gamifizierten Prototyp und nicht gamifizierten ergab sich in keinem signifikanten Unterschied. Das lässt uns vermuten, dass die Interaktionsrefahrung von der Nutzung der gamifizierten und nicht gamifizierten Prototypen sich nicht voneinander unterscheidet.

7 Zusammenfassung

Ziel unseres Gruppen-Projektes "Teach Me Code!" eine App zum Erlernen der beliebten Programmiersprache Python zu entwerfen und zu erstellen. Hierbei sollte eine gamifizierte und eine nicht-gamifizierte Version erstellt und an Nutzern getestet werden. Gemäß des User-Centered-Design Prozesses welches den konzeptuellen Leitfaden unseres Projektes darstellt, wurde zunächst der Nutzungskontext anhand unseres auserwählten Themas erstellt. Da aus dem Thema der Kontext eindeutig hervorging, wurde gleich zu den Interviews übergegangen. In den 5 geführten Interviews mit semi-strukturierten Interviewfragen, welche uns Flexibilität in den Fragestellungen als auch in der Führung der Interviews ermöglichte, gingen wichtige Anforderungen an unseren Prototypen hervor. Im Interview wurden unter Anderem auch Fragen bezüglich des Spielertyps gestellt.

Mit Blick auf die gängigen Taxonomien der Spielertypen soll der primäre Spielertyp des Interviewten hervorgehen, welcher für die Gestaltung der Features für die gamifizierte Version von großer Bedeutung ist. Dank der flexiblen Gestaltung des Interviewdesigns und den regen Anmerkungen und Vorschlägen aus den Interviews und Fokusgruppe konnten wir wertvolle Informationen für unsere App extrahieren. Nach der Analyse der Wettbewerber und Auswertung der Interviews und der Konzipierung der Fokusgruppe, kristallisierten sich in der Anforderungserhebung die Nutzergruppen, sowie gewünschte Anforderung und Features (die Features/Anforderungen beziehen sich auch auf gamifizierte Elemente) heraus, welche dann anschließend als App-Prototyp - sofern diese technisch umsetzbar sind - implementiert worden sind.

Die Gestaltung der Prototypen verlief 3-stufig: Zuerst wurde ein Papier-Prototyp (Low-Prototype) erstellt mit welchen dann verschiedenste Interaktionsmöglichkeiten per Hand simuliert wurden. Im Anschluss wurde mit Axure, der Software für Prototypen, ein sogenannter "Mid-Prototype" erstellt welches die vorgesehenen Funktionalitäten beherrscht. Abschließend wurde die App grafisch und funktional noch einmal aufgewertet, wodurch die App den Status des "High-Fidelity" Prototyps erreichte. Abschließend wurden die beiden Versionen der Prototypen mittels des "Raw NASA TLX Questionnaire" getestet und evaluiert. Insgesamt kann festgestellt

werden, dass der gesamte Prozess des User-Centered Designs vollständig implementiert werden konnte. Ursprünglich war ein A/B Test für die beiden Versionen vorgesehen. Jedoch konnte dies zeitlich nicht bewerkstelligt werden, sodass stattdessen ein Usability Test angewendet wurde.

Die Ergebnisse unseres Usability Tests ergaben, dass es keinen signifikanten Unterschied zwischen der gamifizierten und nicht-gamifizierten Version der App hinsichtlich der Arbeitsbelastung ergab. Neben dem Usability Test haben sich die Kommentare der Probanden als wertvolle Bewertung der Usability unseres Prototypen erwiesen. Von mehreren ProbandInnen wurde positiv angemerkt, dass die App nicht zu überladen sei, d.h. die Darstellung des Designs überforderte die ProbandInnen nicht. Ein negativer Punkt - jedoch ein recht wichtiger - wurde von einer Probandin erwähnt: Bei der falschen Beantwortung der Frage fehlte ihr ein konkretes Fehlerfeedback; Sie hätte gerne gewusst, warum ihre Antwort falsch war.

Obwohl in der Anforderungsspezifizierung ein Systemfeedback vorgesehen war, bestand diese jedoch nur darin, dass angegeben wurde, ob die Frage richtig oder falsch beantwortet wurde.

Insgesamt sind die Ergebnisse unseres Projektes hinsichtlich ihrer ursprünglichen Zielsetzung wenig aussagekräftig. Sofern dies zeitlich einrichtbar gewesen wäre, wäre ein anderes Testverfahren welches beispielsweise die Learnability des Prototypen getestet hätte weitaus sinnvoller gewesen. Zudem schränkten die Interaktionsmöglichkeiten die von der Prototypen-Software zur Verfügung standen unsere Verwirklichung weiterer Features ein. Um dieses Problem beheben zu können, wäre gegebenfalls zusätzlicher Programmieraufwand von Nöten gewesen. Dennoch lassen sich positive Erkenntnisse aus dem Projekt ziehen. So haben sich die Interviews als auch die Gruppeninterviews als wertvollen Impulsgeber erwiesen. Besonders die Vorerfahrung einiger Interviewteilnehmer lieferten unserem Projekt wichtige Hinweise, wie unser Prototyp zu gestalten wäre. Auch zeigte sich die das Feedback außerhalb unseres Testframeworks - d.h. Kommentare unserer Probanden während der Testung - ein wichtige Informationsquelle für Verbesserung oder auch Probleme mit unserem Prototypen, welche von dem eigentlichen Usability-Test nicht erfasst werden können.

Die Gruppenarbeit beziehungsweise die Arbeitsteilung verlief weitestgehend reibungslos. Das Arbeitsklima zwischen den Gruppenmitgliedern kann als angenehm bezeichnet werden. Die Arbeitsteilung und die Vergabe von Aufgabe erfolgte über Sprintlogs auf Github in den jeweils die Aufgaben und ihr Bearbeitungsstatus einsehbar waren. Bedingt durch gegenwärtige pandemische Lage wurden in einem 2 Wochen Takt Zoom-Calls vereinbart, in denen die Aufgabe und ihre Verteilung näher besprochen worden sind. Daneben wurden zusätzlich Ideen ausgetauscht und Verbesserungen bisheriger erfüllter Aufgaben gemacht und diese auch im Sinne des iterativen Design-Prozesses auch umgesetzt. Ansonsten wurden kleiner Aspekte des Projektes über einen Gruppenchat gehandhabt, sofern eine direktes Gespräch mit den Mitgliedern nicht nötig war. Auch die Umstellung, nach dem ein zusätzliches Gruppenmitglied hinzugekommen war, verlief reibungslos. Bezuglich des Zeitplans muss jedoch festgestellt werden, dass Ziele des Projektes nicht immer eingehalten werden konnte. Die Gründe hierfür lagen unter Anderem in den universitären Pflichten der Gruppenteilnehmer neben dem Usability-Engineering Projekt, aber auch im generellen Arbeitstempo welches zu Beginn des Projektes recht hoch war, allerdings im Verlauf der Wochen spätestens mit Beginn der Anforderungserhebung eher nach ließ.

Literatur

10 things to know about the nasa tlx. (o.J.). Zugriff am 20.03.2022 auf <https://measuringu.com/nasa-tlx/>

Aparicio, A., Vela, F. L., González-Sánchez, J. & Isla-Montes, J.-L. (2012, 10). Analysis and application of gamification.. doi: 10.1145/2379636.2379653

Barata, G., Gama, S., Jorge, J. A. & Gonçalves, D. J. (2014). Relating gaming habits with student performance in a gamified learning experience. In *Proceedings of the first acm sigchi annual symposium on computer-human interaction in play* (S. 17–25). New York, NY, USA: Association for Computing Machinery. Zugriff auf <https://doi.org/10.1145/2658537.2658692> doi: 10.1145/2658537.2658692

Bartle, R. (1996, 06). Hearts, clubs, diamonds, spades: Players who suit muds.

Coding bat. (o.J.). Zugriff am 19.02.2022 auf <https://de.wikipedia.org/wiki/Selbstbestimmungstheorie>

Diaper, D. & Stanton, N. A. (Hrsg.). (o.J.). *The handbook of task analysis for human-computer interaction.* Mahwah, NJ: Lawrence Erlbaum.

Effektstärke cohens d. (o.J.). Zugriff am 19.03.2022 auf <https://lindaregber.com/cohens-d/>

Friedrich, J., Becker, M., Kramer, F., Wirth, M. & Schneider, M. (2020, Januar). Incentive design and gamification for knowledge management. *Journal of Business Research*, 106, 341–352. Zugriff am 2022-01-25 auf <https://linkinghub.elsevier.com/retrieve/pii/S0148296319300992> doi: 10.1016/j.jbusres.2019.02.009

Funktionale anforderungen versus nicht-funktionale anforderungen. (o.J.). Zugriff am 18.02.2022 auf <https://www.johner-institut.de/blog/iec-62304-medizinische-software/funktionale-und-nicht-funktionale-anforderungen/>

Gil, B., Cantador, I. & Marczewski, A. (2015). Validating Gamification Mechanics and Player Types in an E-learning Environment. In G. Conole, T. Klobučar, C. Rensing, J. Konert & E. Lavoué (Hrsg.), *Design for Teaching and Learning in a Networked World* (Bd. 9307, S. 568–572). Cham: Springer International Publishing. Zugriff am 2022-01-26 auf http://link.springer.com/10.1007/978-3-319-24258-3_61 (Series Title: Lecture Notes in Computer Science) doi: 10.1007/978-3-319-24258-3_61

Glover, I. (2013, 01). Play as you learn: Gamification as a technique for motivating learners. In (S. 1999-2008).

Goodwin, K. (2009). *Designing for the digital age: how to create human-centered products and services.* Indianapolis, IN: Wiley Pub. (OCLC: ocn244764258)

- Hering, E. (2014). *Wettbewerbsanalyse für Ingenieure*. Wiesbaden: Springer Vieweg.
 (o.J.). Zugriff am 18.02.2022 auf https://files.ifi.uzh.ch/rerg/amadeus/teaching/courses/spezifikation_und_entwurf_ws0506/kapitel_12.pdf
- Jia, Y., Xu, B., Masterson, Y. & Voida, S. (2016, 05). Personality-targeted gamification: A survey study on personality traits and motivational affordances. In (S. 2001-2013). doi: 10.1145/2858036.2858515
- Khaleel, F. L., Sahari@Ashaari, N., Tengku Wook, T. S. M. & Ismail, A. (2016, Dezember). Gamification Elements for Learning Applications. *International Journal on Advanced Science, Engineering and Information Technology*, 6 (6), 868. Zugriff am 2022-01-25 auf http://ijaseit.insightsociety.org/index.php?option=com_content&view=article&id=9&Itemid=1&article_id=1379 doi: 10.18517/ijaseit.6.6.1379
- Kolano, A. (2014). *Geschäftsmodelle für Fachverlage in der Weiterbildung: Eine Analyse auf Basis berufsbegleitender Angebote in Deutschland*. Wiesbaden: Springer Fachmedien Wiesbaden Imprint Springer Gabler.
- Kugler, S. & von Janda-Eble, H. (2018). *Markenmanagement mit System*. Wiesbaden: Springer Fachmedien Wiesbaden. Zugriff am 2022-01-28 auf <http://link.springer.com/10.1007/978-3-658-16225-2> doi: 10.1007/978-3-658-16225-2
- Lazar, J. (2017). *Research methods in human computer interaction* (2nd edition Aufl.). Cambridge, MA: Elsevier.
- Marczewski, A. (2015). *Even ninja monkeys like to play: gamification, game thinking & motivational design*. United Kingdom: Gamified UK. Zugriff auf <https://www.gamified.uk/user-types/>
- Nacke, L. E., Bateman, C. & Mandryk, R. L. (2011). Brainhex: Preliminary results from a neurobiological gamer typology survey. In *Proceedings of the 10th international conference on entertainment computing* (S. 288–293). Berlin, Heidelberg: Springer-Verlag. Zugriff auf https://doi.org/10.1007/978-3-642-24500-8_31 doi: 10.1007/978-3-642-24500-8_31
- Noyes, J. M. & Bruneau, D. P. J. (2007, April). A self-analysis of the NASA-TLX workload measure. *Ergonomics*, 50 (4), 514–519. Zugriff am 2022-03-17 auf <http://www.tandfonline.com/doi/abs/10.1080/00140130701235232> doi: 10.1080/00140130701235232
- Ramkumar, A., Stappers, P. J., Niessen, W. J., Adebahr, S., Schimek-Jasch, T., Nestle, U. & Song, Y. (2017, Februar). Using GOMS and NASA-TLX to Evaluate Human–Computer Interaction Process in Interactive Segmentation. *International Journal of Human–Computer Interaction*, 33 (2), 123–134. Zugriff am 2022-03-17 auf

<https://www.tandfonline.com/doi/full/10.1080/10447318.2016.1220729> doi: 10.1080/10447318.2016.1220729

Rogowski, W. (Hrsg.). (2016). *Business Planning im Gesundheitswesen: die Bewertung neuer Gesundheitsleistungen aus unternehmerischer Perspektive*. Wiesbaden: Springer Gabler. doi: 10.1007/978-3-658-08186-7

Sailer, M., Hense, J., Mandl, H. & Klevers, M. (2013, 12). Psychological perspectives on motivation through gamification. *Interaction Design and Architecture(s) Journal*, 19, 18-37.

Schneider, M. O., Moriya, É. T. U., da Silva, A. S. & Néto, J. C. (2017). Analysis of player profiles in electronic games applying bartle's taxonomy.. Zugriff auf <http://www.sbgames.org/sbgames2016/downloads/anais/157721.pdf>

Schöbel, S. M., Janson, A. & Söllner, M. (2020, November). Capturing the complexity of gamification elements: a holistic approach for analysing existing and deriving novel gamification designs. *European Journal of Information Systems*, 29 (6), 641–668. Zugriff am 2022-01-25 auf <https://www.tandfonline.com/doi/full/10.1080/0960085X.2020.1796531> doi: 10.1080/0960085X.2020.1796531

Toda, A. M., Cristea, A. I., Oliveira, W., Klock, A. C., Palomino, P. T., Pimenta, M., ... Isotani, S. (2019, Juli). A Taxonomy of Game Elements for Gamification in Educational Contexts: Proposal and Evaluation. In *2019 IEEE 19th International Conference on Advanced Learning Technologies (ICALT)* (S. 84–88). Maceió, Brazil: IEEE. Zugriff am 2022-01-31 auf <https://ieeexplore.ieee.org/document/8820847/> doi: 10.1109/ICALT.2019.00028

Tondello, G. F., Wehbe, R. R., Diamond, L., Busch, M., Marczewski, A. & Nacke, L. E. (2016). The gamification user types hexad scale. In *Proceedings of the 2016 annual symposium on computer-human interaction in play* (S. 229–243). New York, NY, USA: Association for Computing Machinery. Zugriff auf <https://doi.org/10.1145/2967934.2968082> doi: 10.1145/2967934.2968082

Use cases. (o. J.). Zugriff am 20.03.2022 auf <https://www.usability.gov/how-to-and-tools/methods/use-cases.html>

Webb, E. N. (2013). Gamification: When it works, when it doesn't. In A. Marcus (Hrsg.), *Design, user experience, and usability. health, learning, playing, cultural, and cross-cultural user experience* (S. 608–614). Berlin, Heidelberg: Springer Berlin Heidelberg.

Xu, Y. (2011, 09). Literature review on web application gamification and analytics. *CSDL Technical Report*.

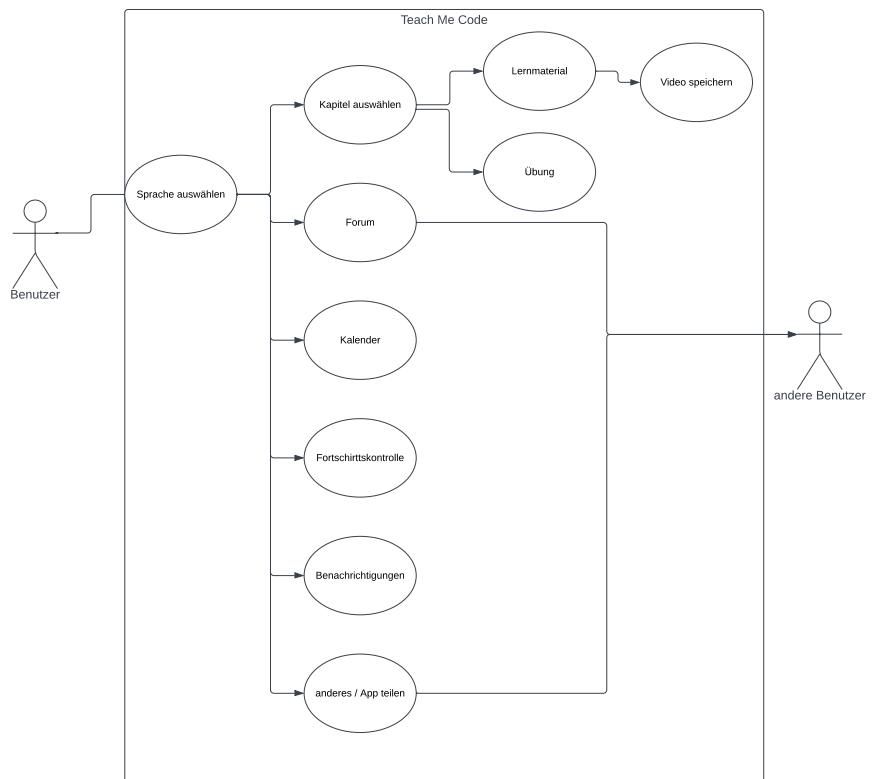


Abbildung 12: Use-Cases

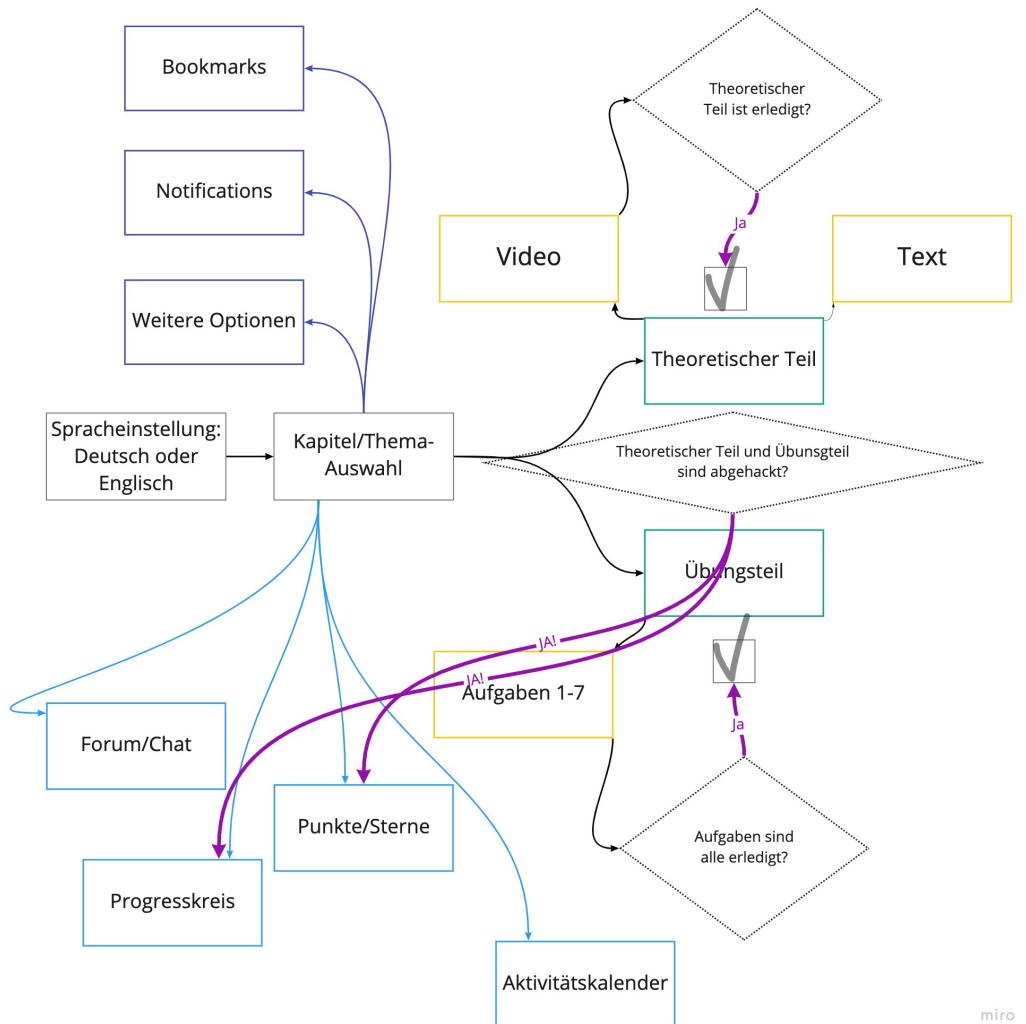


Abbildung 13: Hierarchische Taskanalyse

Statistik bei gepaarten Stichproben

		Mittelwert	N	Std.-Abweichung	Standardfehler des Mittelwertes
Paaren 1	gamified	23,5000	6	8,33667	3,40343
	non_gamified	30,6667	6	11,87715	4,84883

Abbildung 14: Deskriptive Statistik

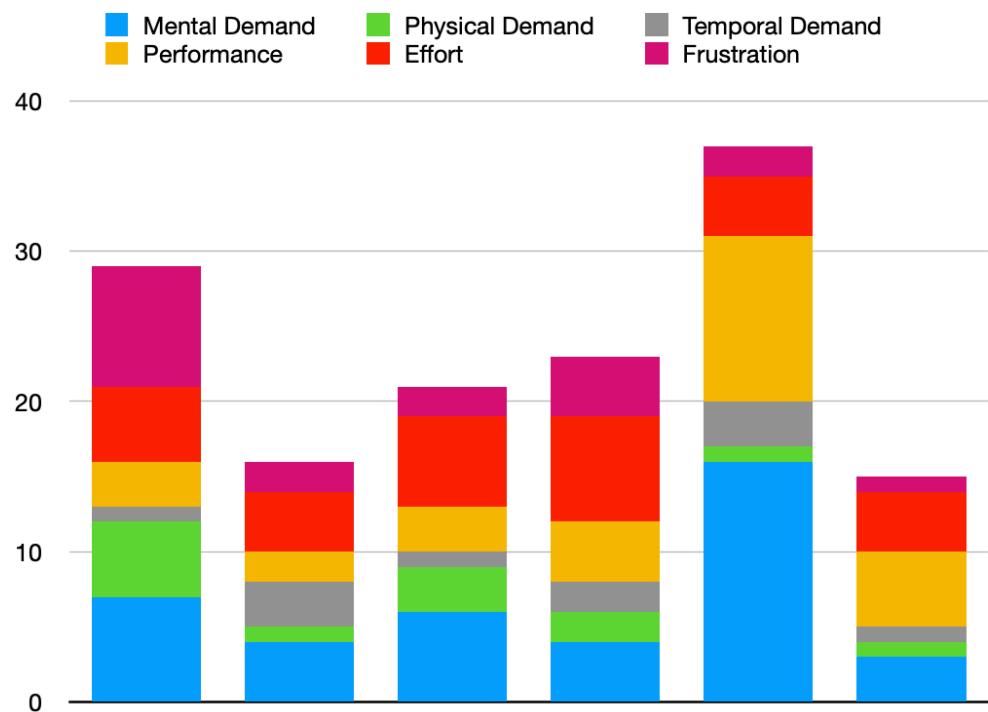


Abbildung 15: NASA-TLX-Kategorie für gamifizierte Prototype

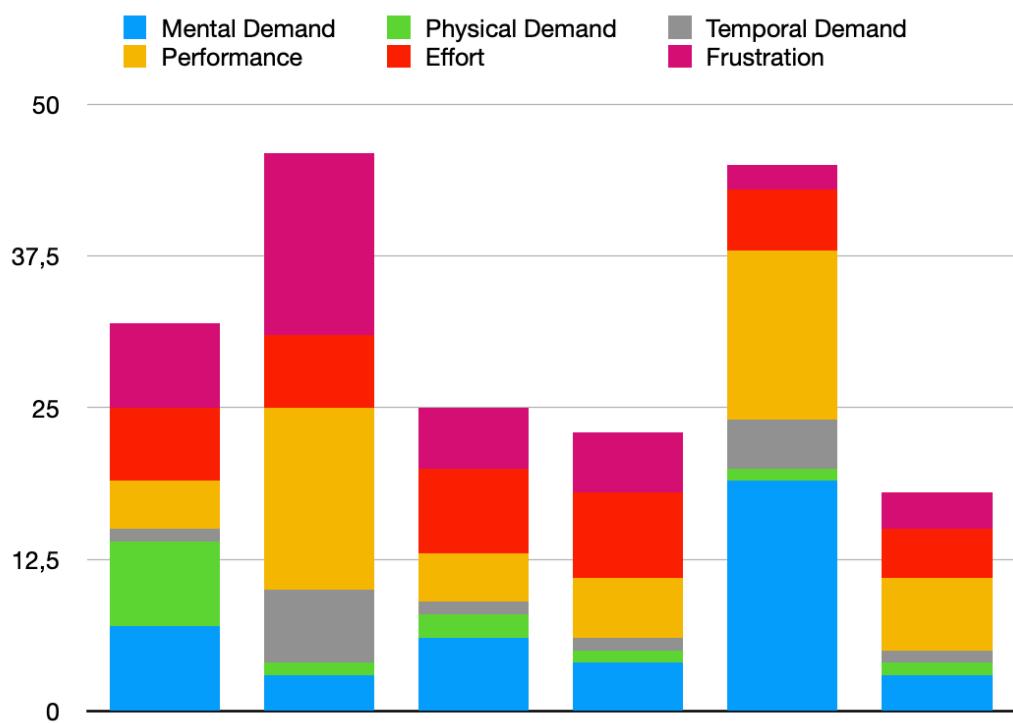


Abbildung 16: NASA-TLX-Kategorie für nicht gamifizierte Prototype

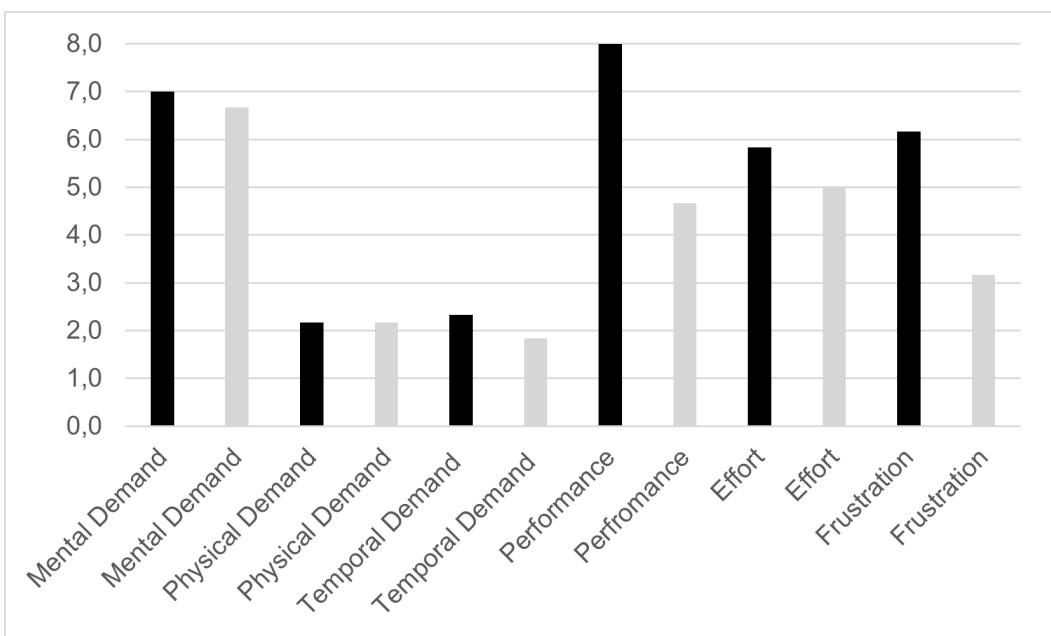


Abbildung 17: Mittelwerte von NASA-TLX Kategorien für nicht-gamifizierte und gamifizierte Prototypen