MASTERARBEIT / MASTER'S THESIS

Titel der Masterarbeit / Title of the Master's Thesis

„Erstellung und Durchführung eines didaktisch aufbereiteten Tutoriums über das Cross-Platform Framework Flutter im Zuge der Lehrveranstaltung Mensch-Computer-Interaktion"

verfasst von / submitted by

Andreas Öttl, BEd

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Master of Education (MEd)

Wien, 2022

Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet:	UA 199 500 514 02
Studienrichtung lt. Studienblatt / degree programme as it appears on the student record sheet:	Masterstudium Lehramt SEK (AB) Unterrichtsfach Bewegung und Sport. Unterrichtsfach Informatik
Betreut von / Supervisor:	Univ.-Prof. Dr. rer. nat. Dipl.-Inf. Maria Knobelsdorf

Abstract

Studierende können in bestimmten Lehrveranstaltungen an einem Tutorium teilnehmen, das sie bei der Erreichung der notwendigen Ziele unterstützt. Wenn die Universität aufgrund einer Pandemie nur eine Distanzlehre anbietet, wird die Planung und Abhaltung eines Tutoriums nochmals erschwert und es erfordert zusätzliche Änderungen im didaktischen Design.

Die vorliegende Masterarbeit beschäftigt sich mit der Erstellung und Durchführung eines didaktisch aufbereiteten Tutoriums, welches als Teil der Lehrveranstaltung „Mensch-Computer-Interaktion“ an der Universität Wien abgehalten wurde. Das Tutorium dient als Unterstützung zum Erlernen des Cross-Platform Frameworks Flutter, das zur Entwicklung von Apps verwendet werden kann.

In dieser Arbeit wird zu Beginn der theoretische Background beschrieben, der für die Erstellung des Tutoriums notwendig ist. Didaktische Richtlinien und Unterrichtsstrategien im Allgemeinen, und im Speziellen für die Fernlehre, als auch theoretische Grundlagen der Mensch-Computer-Interaktion und des Cross-Platform Framekworks Flutter dienen als Grundlage für das Forschungsprojekt.

Um die Forschungsfrage zu beantworten, wurde ein Tutorium didaktisch aufbereitet und erprobt. Zur Ergebnissicherung wurde ein, von den an der Lehrveranstaltung teilnehmenden Studierenden ausgefüllter, quantitativ-qualitativer Fragebogen, ausgewertet. Die Ergebnisse zeigen, dass das zusätzlich angebotene Tutorium einen Mehrwert für die Studierenden bietet, und sich dieses Tutorium in weiterer Folge positiv auf ihre Kompetenzentwicklung auswirkte.

Danksagung

Zu Beginn möchte ich meiner Betreuerin Frau Univ.-Prof. Dr. Maria Knobelsdorf für die außerordentlich gute Zusammenarbeit danken. Durch ihr fachdidaktisches Wissen und ihre Unterstützung bei Problemstellungen in der wissenschaftlichen Umsetzung konnte ich im Laufe dieser Arbeit noch sehr viel lernen.

Weiters möchte ich meinen Dank an Herrn Prof. Möller und dem Team der Forschungsgruppe Visualization and Data Analysis, insbesondere Frau Laura Koesten, für die Möglichkeit der Durchführung des Tutoriums und der guten Zusammenarbeit, aussprechen.

Für die großartige Unterstützung in finanzieller Hinsicht als auch im Allgemeinen über das ganze Studium hinweg möchte ich mich bei meinen Eltern bedanken. Für die Unterstützung in meinem Unterrichtsfach Sport möchte ich mich bei meiner Tante Anni Schopf bedanken.

Meiner Freundin Valentina, die mir immer zur Seite stand, unterstützte und zu einem raschen Abschluss motivierte, gebührt auch ein großer Dank. Abschließend möchte ich noch meinen Bruder und meinen Freunden danken, die ebenso einen Teil zum Studienabschluss beigetragen haben.

Inhaltsverzeichnis

1 Einleitung ... 1

1.1 Motivation ... 2

1.2 Methodik und Aufbau der Arbeit ... 3

2 Theoretische Rahmung ... 4

2.1 Guter Unterricht ... 4

2.2 Lernmotivation ... 5

2.3 Weitere Unterrichtsstrategien ... 10

2.4 Distanzlehre ... 12

3 Gestaltungsprinzipien für Benutzeroberflächen ... 16

4 Cross Platform Frameworks ... 31

4.1 Entstehung von Cross-Platform Frameworks ... 31

4.2 Derzeitige Technologien ... 32

4.3 Dart ... 33

4.4 Flutter ... 34

5 Forschungsdesign ... 35

5.1 Forschungsfrage ... 35

5.2 Didaktisches Design des Tutoriums ... 35

5.3 Teilnahme und Durchführung des Tutoriums ... 37

5.4 Forschungsmethoden ... 42

5.5 Erhebung der Daten ... 45

6 Ergebnisse ... 46

6.1 Auswertung und Ergebnisse der quantitativen Daten ... 46

6.2 Quantitative Daten der Lehrveranstaltungsabgaben ... 56

6.3 Auswertung und Ergebnisse der qualitativen Daten ... 57

7 Diskussion ... 62

7.1 Didaktische Reflexion des Tutoriums ... 62

7.2 Interpretation der quantitativen Ergebnisse ... 64

7.3 Interpretation der qualitativen Ergebnisse ... 66

7.4 Zusammenfassung 70
8 Fazit und Ausblick 71
9 Abbildungsverzeichnis 72
10 Tabellenverzeichnis 73
11 Literaturverzeichnis 73

1 Einleitung

Aufgrund der stetig steigenden Vernetzung verschiedenster Geräte bekommt die digitale Grundbildung in der Schule einen immer höheren Stellenwert. Für die Vermittlung dieser Kompetenzen werden ausgebildete Lehrkräfte mit einem Schwerpunkt in Informatik benötigt. In den österreichischen Schulen herrscht derzeit jedoch ein Mangel an ausgebildeten Informatiklehrerinnen und Informatiklehrern. [1]

An der Universität Wien gibt es im Lehramtstudium die Möglichkeit, das Unterrichtsfach Informatik zu wählen. Für eine vollständige LehrerInnen-Ausbildung werden ein Bachelorabschluss im Umfang von 240 ECTS, sowie ein Masterabschluss im Umfang von 120 ECTS benötigt. Dabei müssen mitsamt der Erstellung der Masterarbeit insgesamt 156 ECTS an der Fakultät für Informatik absolviert werden. Hierbei erlangen Studierende ein fundiertes fachwissenschaftliches Wissen. [2], [3] Im Masterstudium der Lehrerinnen-Ausbildung wird der Fokus verstärkt auf die Fachdidaktik des jeweiligen Faches gelegt. Auf der Fakultät für Informatik gibt es dabei mehrere Lehrveranstaltungen, die einen tiefen Einblick in die Theorie und Praxis der Fachdidaktik, sowie speziellen Kapiteln der Fachdidaktik in Informatik ermöglichen. [2]

Im Curriculum der jeweiligen Studiengänge des Fachbereichs Informatik befinden sich Lehrveranstaltungen mit einer Vorlesung, als auch einer zusätzlichen Übung, in der die theoretischen Erkenntnisse praxisnah umgesetzt werden. Als Unterstützung werden in einigen Modulen kostenlose Tutorien angeboten.

Ein Tutorium kann Studierende bei der Erreichung der Lehrveranstaltungsziele in großem Maße unterstützen. Damit ein solches Tutoriums-Angebot von den Studentinnen und Studenten auch wahrgenommen wird, und diese bei der Teilnahme in bestmöglicher Form profitieren können, bedarf es einer durchdachten Planung, Vorbereitung und Umsetzung.

Die Lehrveranstaltung „Mensch-Computer-Interaktion“ wird sowohl als Pflichtmodul im Bachelorstudium für Informatik, als auch als Wahlmodul im Lehramtstudium von der Fakultät für Informatik an der Universität Wien angeboten. Der Umfang der Lehrveranstaltung beträgt fünf ECTS und beinhaltet neben dem Vorlesungsteil auch einen Übungsteil.

Für einen positiven Abschluss müssen die Studierenden folgende Ziele erreichen: [4]

1. Eine benutzerfreundliche Bedienoberfläche gestalten und die Usability dieser bewerten
2. Usability Guidelines praktisch anwenden können
3. Die wissenschaftlichen Erkenntnisse der Kommunikationspsychologie und der Kognition in der Erstellung von Apps berücksichtigen
4. Eine interaktive App im Team erstellen, welche die theoretischen Grundlagen der Mensch-Computer-Interaktion berücksichtigt

Da für die Erstellung der App eine geeignete Programmiersprache benötigt wird, welche die Studierenden zu Beginn der Lehrveranstaltung in der Regel noch nicht kennen, werden spezielle Tutorien für konkrete Programmier-Frameworks angeboten. Die in der Lehrveranstaltung angebotenen Tutorien beinhalten eine Einführungseinheit, in welcher die Einrichtung und Installation erklärt wird, und eine zusätzliche Einheit, welche als Q&A dient und Fragen der Studentinnen und Studenten klärt.

1.1 Motivation

Ich bin in der Lehrveranstaltung Mensch-Computer-Interaktion das erste Mal mit einer App-Entwicklung über eine Cross-Plattform in Berührung gekommen. Dabei wurde *Cordova* als Plattform angeboten, die ich mir im Selbststudium beibrachte. Dabei bemerkte ich, dass diese neue Form der App-Erstellung große Vorteile bietet und Prototypen mit wenig Zeitaufwand entwickelt werden können. Leider wurde im Teamprojekt nicht die Cross-Plattform verwendet, da die Teammitglieder die Einzelaufgabe „Hello-World“ mittels Cordova nicht umsetzen konnten.

Durch meine nebenberufliche Tätigkeit lernte ich das neue plattformunabhängige Framework Flutter von Google kennen, welche eine bessere Performance als Cordova bietet und einen exzellenten Workflow aufweist, womit sehr schnell mehrere Prototypen entworfen werden können. Dieses schnelle Entwickeln von Prototypen weist großes Potential für die Lehrveranstaltung Mensch-Computer-Interaktion auf. Sobald die Sprache erlernt wurde, kann der Großteil der Arbeitszeit für die Verbesserung der User Experience verwendet werden.

Das Ziel dieser Masterarbeit ist die Erstellung und Durchführung eines umfangreichen Tutoriums, welches die verschiedenen Funktionen des Cross-Platform Frameworks Flutter vermittelt, und dabei die Ziele der Lehrveranstaltung beachtet und miteinbezieht. Das Tutorium wird zusätzlich so strukturiert und aufgebaut, dass es die im Studium erlernten didaktischen Richtlinien berücksichtigt.

1.2 Methodik und Aufbau der Arbeit

In dieser Arbeit werden zu Beginn die didaktischen Grundlagen für einen guten Unterricht, die für die Erstellung des Tutoriums notwendig sind, beschrieben. Aufgrund des Distance-Learning wird auch ein theoretischer Background über die Lernmotivation und die Distanzlehre benötigt. Als weitere theoretische Rahmung werden die Gestaltungsprinzipien der Benutzeroberflächen beschrieben und Einblicke in Cross-Platform Frameworks im Allgemeinen und Flutter im Speziellen gewährt.

Anschließend wird das Forschungsprojekt dieser Arbeit beschrieben. Dabei wurde ein Tutorium mit den didaktischen Prinzipien aus der theoretischen Rahmung geplant und im Zuge der Lehrveranstaltung „Mensch-Computer-Interkation“ an der Universität Wien abgehalten. Die Studierenden beantworteten nach der Durchführung einen quantitativ-qualitativen Fragebogen über die Qualität des Tutoriums und der eigenen Kompetenzentwicklung. Die daraus gewonnenen Daten werden im Diskussionsteil für die Beantwortung der Forschungsfrage verwendet.

Abschließend wird ein Fazit gezogen, auf Limitationen eingegangen und ein Ausblick auf mögliche folgende Forschungsarbeiten gegeben.

2 Theoretische Rahmung

2.1 Guter Unterricht

Eine hohe Qualität des Unterrichts fördert das Lernen und führt infolgedessen zu einem besseren Kompetenzerwerb der teilnehmenden Personen. Für die Planung und Durchführung eines qualitativ hochwertigen Unterrichtes müssen die dafür gegebenen Kriterien beachtet werden. Die von Hilbert Meyer definierten zehn Merkmale leisten hierbei Hilfe. Hilbert Meyer definiert zehn Merkmale für einen guten Unterricht. [5]

- **Klare Strukturierung des Unterrichts**
  Die Klarheit der Prozessstruktur ist ein wesentliches Merkmal eines guten Unterrichts. Der „rote Faden“ muss für die teilnehmenden Personen erkennbar sein. Die gewählte Sprache sowie die Aufgabenstellungen müssen für alle Beteiligten verständlich sein.
- **Hoher Anteil echter Lernzeit**
  Damit die Schülerinnen und Schüler im Unterricht aktiv im Unterricht lernen können, darf es wenig Störaktionen und Ablenkungen geben. Langeweile soll vermieden und sinnvolle Pausen zur Erholung eingesetzt werden.
- **Lernförderliches Klima**
  Eine angenehme Atmosphäre im Unterricht sorgt für ein lernförderliches Klima. Dabei müssen Lehrerinnen und Lehrer mit den Schülerinnen und Schüler, als auch diese untereinander respektvoll miteinander umgehen. Rivalitäten, Beschimpfungen, Diskriminierungen und Regelbrüche hemmen ein lernförderliches Klima.
- **Inhaltliche Klarheit**
  Ein gut gewählter Unterrichtseinstieg, regelmäßiges Wiederholen des Wissens und das Festhalten der Zwischenergebnisse liefern den Schülerinnen und Schülern eine gewisse Klarheit. Die Aufgabenstellungen müssen zum aktuellen Thema passen und ebenfalls verständlich sein.
- **Sinnstiftendes Kommunizieren**
  Hierbei wird der Prozess bezeichnet, in dem die Schülerinnen und Schüler mit der Lehrperson in Kontakt treten und dabei kritische Fragen stellen, Rückmeldung zu Lernschwierigkeiten geben und ihre eigenen fachlichen Interessen mitteilen.
- **Methodenvielfalt**
  Verschiedene Unterrichtsformen, Handlungsmuster und Inszenierungstechniken bringen eine Abwechslung und steigern die Unterrichtsqualität.

- **Individuelles Fördern**
  Da die einzelnen Schülerinnen und Schüler nicht den gleichen Leistungsstand aufweisen, sollten mehrere Aufgaben mit unterschiedlichen Schwierigkeitsgraden angeboten werden. Bei leistungsschwachen Personen wird ein zusätzliches Lernangebot benötigt, Leistungsstarke sollten die Möglichkeit bekommen, nach der Fertigstellung der Aufgaben an ihren eigenen gesetzten Schwerpunkten weiterarbeiten zu können.
- **Intelligentes Üben**
  Das gelernte Wissen wird mit passend dafür ausgewählten Arbeitsmaterialien geübt. Die Phasen des Übens sind kurz, werden jedoch häufiger im Unterricht eingebaut. In der Übungszeit herrscht eine angenehme und ruhige Arbeitsatmosphäre, damit die Schülerinnen und Schüler konzentriert bleiben.
- **Transparente Leistungserwartungen**
  Die Leistung wird anhand der Bildungsstandards und den gültigen Richtlinien gesetzt und den Schülerinnen und Schülern verständlich vermittelt. Sie erhalten regelmäßig Feedback zu ihrer Leistung. Für die Beurteilung werden verschiedene Formen der Leistungskontrolle angewendet.
- **Vorbereitende Umgebung**
  Eine vorbereitende Umgebung ist vorhanden, wenn im Klassenraum Ordnung herrscht. Die vorhandene Einrichtung muss intakt sein und die gegebene Akustik, Beleuchtung und Belüftung sollen ein angenehmes Arbeiten ermöglichen.

2.2 Lernmotivation

Die Theorie hinter der Lernmotivation und welche Arten von Motivation es bei einer lernenden Person gibt, ist für die Erstellung des Tutoriums sehr wichtig. In der bereits in der Einleitung beschriebenen Lehrveranstaltung Mensch-Computer-Interaktion ist der Arbeitsumfang sehr hoch. Es gibt viele Abgaben, Vorlesungstermine und verpflichtende Übungseinheiten. Das Tutorium wird zusätzlich angeboten und kann freiwillig besucht werden. Damit dieses Angebot von den Studierenden auch angenommen wird, muss in der Gestaltung des Tutoriums darauf geachtet werden, die Motivation zur Teilnahme zu stärken. Diese Motivationsförderung steigert in der Folge auch den Fokus auf die Lerninhalte und Lernziele. [6]

Bevor Faktoren der Lernmotivation beschriebenen werden, ist eine Definition von *Motivation* erforderlich. Pschyrembel (2010, S.1) definierte dies folgendermaßen:

> *„Motivation bezeichnet die Gesamtheit aller Motive (Beweggründe), die zur Handlungsbereitschaft führen, und das auf emotionaler und neuronaler Aktivität*

beruhende Streben des Menschen nach Zielen oder wünschenswerten Zielobjekten.“

Bei der Lernmotivation handelt es sich um einen Sammelbegriff, der alle möglichen Beweggründe einer Person vereint, die diese zum Entschluss bringen, etwas Neues zu lernen. Wie stark die Lernmotivation einer lernenden Person ausgeprägt ist, hängt von den Persönlichkeitsmerkmalen, wie die Fähigkeiten und Fertigkeiten als auch die generelle Motivationsausprägung, und den Anreizen der Situation selbst ab, die von der lehrenden Person beeinflussbar sind. [7]

Generell kann eine Person sehr viele Beweggründe haben, die ihm zum Lernen leiten. Ebenfalls können Zuneigung und Geborgenheit, als auch Geltung und Anerkennung ein Grund zum Lernen sein, diese werden als soziale Lernmotive bezeichnet. [7]

Diese sozialen Lernmotive können neben der Motivationssteigerung auch ein Störfaktor sein, und somit hemmend wirken. Auch das generelle Interesse an diesem Bereich und die Neugier können die Lernmotivation maßgeblich beeinflussen. [7]

2.2.1 Extrinsische Motivation

Die extrinsische Motivation liegt außerhalb der Beziehung der lernenden Person zum Lernstoff, und kann die generelle Lernmotivation verstärken. Lernende Personen, die extrinsisch motiviert sind, nehmen diese Arbeit auf sich, um bessere Noten, Lob oder Prestige zu erhalten. Deci & Ryan definierten vier Formen der extrinsischen Motivation: [8]

1. **Externale Regulation**
   Etwas wird nur durchgeführt, damit man eine Strafe vermeidet oder belohnt wird. Das Verhalten wird vollständig von außen gesteuert.
2. **Introjizierte Regulation**
   Durch „innere Kräfte“ werden Handlungen ausgeführt. Da diese Kräfte nicht willentlich gesteuert werden können, liegen sie außerhalb der eigenen Entscheidungsfähigkeit.
3. **Identifizierte Regulation**
   Die von externer Seite gegebenen Ziele werden nicht wie in der externalen Regulation einfach umgesetzt. Die Person kann sich mit den Zielen identifizieren und steht persönlich hinter diesen.
4. **Integrierte Regulation**
   Diese Form deckt sich bereits sehr stark mit der im nächsten Abschnitt beschriebenen intrinsischen Motivation. Durch ein selbstbestimmtes Handeln verfolgt man die gegebenen Ziele. Diese Ziele kommen hierbei jedoch noch von „außen“.

2.2.2 Intrinsische Motivation

Bei der intrinsischen Motivation beschäftigt man sich jedoch aus Eigeninteresse mit neuen Aufgaben und nicht, wie bereits bei der extrinsischen Motivation beschrieben, durch einen äußeren Einfluss. Wenn die gegebene Aufgabe erfolgreich gelöst wurde, belohnt man sich in weiterer Folge selbst und es entsteht ein Erfolgsgefühl. [8]

Um diesen Antrieb zu ermöglichen, muss eine positive Beziehung zwischen dem Lernenden und dem gegebenen Lernstoff bestehen. Mögliche intrinsische Motive können die Neugier und der Drang zu neuem Wissen, als auch der Drang zur Vollendung sein. Ein wichtiges intrinsisches Motiv ist der Aufforderungscharakter, der den Lernenden dazu auffordert, sich mit dem gewählten Wissensgebiet zu beschäftigen, obwohl er dabei keinen direkten Nutzen zieht. [7]

2.2.3 Die Lernumgebung

Bei der Lernumgebung ist es wichtig, sie so zu gestalten, dass entspanntes Lernen ermöglicht wird. Hierbei werden vier Aspekte der Gestaltung unterschieden:

2.2.3.1 Architektonische Gestaltung

Bei den Lernräumen sollte darauf geachtet werden, dass sie viel Freiraum bieten und die lernenden Personen sich in verschiedenen Lernzonen zurückziehen können. Wichtig ist, dass ein Bewegen im Raum nicht eingeschränkt wird. [9]

2.2.3.2 Strukturelle Gestaltung

Die Lernräume müssen so strukturiert sein, dass die Lernenden jederzeit einen Zugriff auf alle Materialen haben. Dadurch wird ein lehrerunabhängiges Lernen möglich. Einerseits sollen strukturierte und lernzielorientierte Materialien vorhanden sein, aber auch unstrukturierte Materialien, womit ein endeckendes Lernen ermöglicht wird. Eine digitale Lernplattform bietet genau diese Möglichkeiten. [9]

2.2.3.3 Organisatorische Gestaltung

Hierbei sollen die Lernenden so viel Freiraum wie nötig bekommen, in der sie sich Lernpartnerinnen und Lernpartner suchen können. Dadurch kann die aktive Lernzeit erhöht werden. [9]

2.2.3.4 Menschliche Gestaltung

Durch Verzicht auf Druck, aber mit einer klaren Planung und definierten Zielen kann eine angenehme Lernatmosphäre geschaffen werden, die ein Grundpfeiler für erfolgreiches Lernen ist. Ein Gespräch unter vier Augen ist ebenso wichtig. [9]

2.2.4 ARCS-Modell

Lernphasen, die über längere Zeit nur online abgehalten werden, benötigen ein durchdachtes motivationales Design, da es in diesem Setting noch herausfordernder ist,

die Lernenden motiviert zu halten. Eine Studie über sechs Jahre, welche vom Massachusetts Institute of Technology durchgeführt wurde, beobachtete die Dropout-Raten in mehr als 500 onlinebasierten Kursen auf verschiedenen Massive Open Online Course (MOOC) Plattformen. Bei den Teilnehmerinnen und Teilnehmern, welche vor dem Kursstart einen positiven Abschluss anvisierten, brachen im Laufe des Kurses mehr als 80% ab. Bei Einschreibgebühren lag die Anzahl der Abbrecherinnen und Abbrecher noch immer über 50%. Zur Schaffung einer motivierenden Lernumgebung lässt sich das ARCS-Modell von Keller anwenden, das, wie in Abbildung 1 dargestellt, vier motivationale Dimensionen aufweist: [10]

2.2.4.1 Aufmerksamkeit (Attention)

Die Aufmerksamkeit spielt eine große Rolle beim Lernen im Unterricht. In der Online-Lehre muss hier besonders darauf geachtet werden, da generell weniger Interaktion zwischen dem Lehrenden und den Lernenden herrscht. Um das Interesse bei den Kursteilnehmerinnen und Kursteilnehmer zu wecken, können widersprüchliche, provokante und unerwartete Einstiege mit Fragen oder Problemstellungen angewendet werden. Mit abwechslungsreichen didaktischen Methoden und verschiedenen Medien lässt sich die Aufmerksamkeit steigern, ebenso mit unterschiedlichen Interaktionsformen, wie dem Wechsel von Frontalunterricht zu einer Partnerarbeit. [10] Weitere Strategien zur Steigerung der Aufmerksamkeit sind nach Keller das Ermöglichen von Nachfragen, eine aktive Teilnahme im Unterricht, sowie das Zeigen von Humor. [11]

2.2.4.2 Relevanz (Relevance)

Für die Lernenden muss das vermittelte Wissen eine gewisse Relevanz aufweisen, damit eine Motivation für neues Lernen entsteht. Neben den Inhalten an sich, können verschiedene didaktische Methoden die gegebenen Lernziele als relevant erscheinend lassen. Wenn die Ziele und Motive der Kursteilnehmerinnen und Kursteilnehmer bekannt sind, lassen sich Aufgaben so gestalten, dass ein Bezug zu deren Lebenswelt geschaffen, eine Vertrautheit vermittelt, und dadurch als relevant wahrgenommen wird. [10]

2.2.4.3 Erfolgszuversicht (Confidence)

Klar mitgeteilte Anforderungen, die für das Erreichen der Ziele notwendig sind, geben den Lernenden eine gewisse Zuversicht und steigern das Selbstvertrauen. Durch Musterlösungen und Beispielen zu Aufgaben und Tests haben Teilnehmerinnen und Teilnehmer die Möglichkeit, sich selbst und ihre Erfolgsaussichten einzuschätzen.

Lernsituationen, in denen Fehler keinen Einfluss auf die Note haben, sowie generelle Erfolgserlebnisse im Lernen können die Erfolgszuversicht ebenso fördern. [10]

2.2.4.4 Zufriedenheit (Satisfaction)

Die vierte von Keller definierte Bedingung ist die Zufriedenheit. Ein informatives Feedback, transparente Bewertungskriterien und die Möglichkeit, das Gelernte auf andere Bereiche zu transferieren und anzuwenden, kann die Zufriedenheit in hohem Maße steigern. [10]

Bedingungen für Motivation			
Aufmerk-samkeit	**Relevanz**	**Erfolgs-zuversicht**	**Zufriedenheit**
Interesse wecken	Zielorientierung	Lern-anforderungen	Gelegenheiten zur Anwendung
Neugierde anregen	Motivpassung	Erfolgserlebnisse	Positive Folgen
Abwechslung	Vertrautheit	Selbstkontrolle	Fairness

Abbildung 1: ARCS-Modell Matrix
Quelle: [10, S. 43]

2.2.5 Die Motivation der Lehrperson

Neben einer vorbereitenden Gestaltung und Planung des Unterrichts hängt es auch von der Lehrperson selbst ab, ob und inwiefern die Schülerinnen und Schüler für das Fach und den zu vermittelnden Stoff motiviert werden. Nur wenn sich die Lehrperson sich mit dem Fach und dem zu unterrichtenden Stoff identifiziert, kann guter Unterricht entstehen. [12]

„Ganz allgemein gilt folgendes: Ein Mensch macht eine Sache gut, wenn die Sache im Freude macht [sic], er den Dingen aus eigener Motivation nachgeht, und er sich in und mit der Sache auskennt.“ (Spitzer 2007, S.413)

2.2.6 Relevanz für das Tutorium

Um die Studierenden über die gesamte Dauer des Tutoriums motiviert zu halten, kann ein motivationales Design bei der Umsetzung helfen. Das ARCS-Modell von Keller kann in der Erstellung des didaktischen Designs mit seinen vier Dimensionen unterstützend wirken und damit möglicherweise die Motivation bei den Studierenden steigern.

2.3 Weitere Unterrichtsstrategien

2.3.1 Medieneinsatz

Medien haben in der Lehre schon immer eine wichtige Rolle gespielt. Früher wurden die Begriffe *Unterrichtsmittel* und *Lehrmittel* verwendet, in den 1970er-Jahren wurden diese vom *Unterrichtsmedium* abgelöst. [13] Einerseits gibt es „personale Medien“ wie die Bewegung, Gestik und Mimik, als auch „nichtpersonale Medien“. Zu diesen zählen Objekte aus der Realität, Abbildungen, Modelle und Symbole. [6] Diese nichtpersonalen Medien werden in diesem Kapitel näher beleuchtet. Eine weitere Unterscheidung ist zwischen traditionellen und digitalen Medien möglich. [6] Mit digitalen Medien ist es Schülerinnen und Schülern möglich, selbstständig ohne Lehrperson zu lernen. [14] Medien können zu jeder Phase des Unterrichts eingesetzt werden, eine Anpassung an den Inhalt und das Ziel ist jedoch notwendig, um eine bestmögliche Wirkung zu erzielen.[6]

2.3.2 Veranschaulichung

Der Einsatz verschiedener Medien wurde in Kapitel 2.3.1 beschrieben. In diesem Abschnitt wird nochmals genauer auf die Veranschaulichung eingegangen. Wenn das erworbene Wissen veranschaulicht werden kann, bietet dies einen großen Mehrwert. Diese Erkenntnis ist schon lange in der Didaktik bekannt. Comenius beschrieb bereits 1657 die Wichtigkeit der Erfahrung und Wissensvermittlung über die Sinne des Menschen. Falls etwas nicht unmittelbar erfahrbar ist, sollte über Bilder und Modelle eine Erfahrung ermöglicht werden. [15] Mit der Veranschaulichung kann die die Motivation gesteigert und das allgemeine Verständnis verbessert werden. [16] In der Didaktik der Informatik ist die Visualisierung ebenfalls ein wesentlicher Aspekt. Durch das Veranschaulichen von Programmzuständen und der daraus resultierenden Veränderungen der Datenzustände können Personen das Programmieren leichter verstehen und dadurch auch erlernen. [17] Zur visuellen Veranschaulichung können neben realen Objekten und Präparaten auch Bilder, Bücher, Videos, Overheadfolien und Arbeitsblätter verwendet werden [6]

2.3.3 Sprache und Stimme

Neben dem fachlichen Wissen und pädagogisch didaktischen Kenntnissen ist die Vermittlung der Inhalte eine Grundvoraussetzung, um Lernen zu ermöglichen. Die Stimme ist hierbei das geläufigste Medium. Als Lehrperson hat man im Gegensatz zu anderen Berufen jedoch keine unterstützenden Hilfsmittel wie ein Mikrofon, sondern ist hierbei auf sich allein gestellt. An einem regulären Schultag werden die Stimmbänder und der Kehlkopf in einem nicht zu unterschätzenden Ausmaß beansprucht.

Neben dem alltäglichen Training können auch bewährte Strategien angewendet werden, um die Stimme zu schonen. Ebenfalls gibt es Stimm- und Sprechtechniken, die ein besseren Zuhören und Verständnis ermöglichen. [18]

Dass ein Stimmtraining auch einen Einfluss auf die langfristige Gesundheit von Lehrpersonen hat, wurde von Prof. Michael Fuchs durch eine Studie bestätigt. Bei Lehrerinnen und Lehrer, die während ihres Studiums keine Ausbildung zur Stimme und Sprache absolvierten, wurde ein 1,6faches Risiko einer berufsbedingten Dysphonie festgestellt. [19] Dr. Sigrid Lemke untersuchte bei angehenden Lehrpersonen die Stimm- und Atemleistung, wobei 37% der Probandinnen und Probanden deutliche Auffälligkeiten aufwiesen, die einer ärztlichen Untersuchung bedarf. [20]

Verschiedene Übungen zur Steigerung und Stärkung der Lautstärke trainieren das Zwerchfell, sowie die Bauch- und Brustmuskulatur. Diese müssen in regelmäßigen Abständen kontinuierlich durchgeführt werden, um einen hörbaren Erfolg zu erhalten. [18] Damit die Zuhörerinnen und Zuhörer beim Vortrag nicht gedanklich abschweifen, können rhetorische Stilmittel verwendet werden. Das Ändern der Sprechgeschwindigkeit und der Tonlage steigern die Aufmerksamkeit. Ebenso sollten während dem Sprechen bewusst gesetzte Pausen eingefügt werden, damit das Publikum die Informationen verarbeiten kann. [21] [14]

Wichtig ist auch die Beachtung der Unterrichtssprache je nach dem Sprachverständnis der Schülerinnen und Schüler. Fremdwörter und eine hochgestochene Wortwahl sollten vermieden werden, da dies eine Barriere beim Lernen darstellt. Generell sollten kurze Sätze und geläufige Wörter verwendet werden, und Fachbegriffe mit einer anschaulichen Sprache erklärt werden. [14]

2.3.4 Lehrverfahren

Ein Lernprozess kann durch unterschiedliche Verfahren gestartet werden. Kiper unterscheidet hier drei verschiedene Arten: [13]

- **Darbietendes Lehrverfahren**
  Dieses Verfahren dient in erster Linie der Vermittlung von Informationen und Fakten. Schülerinnen und Schüler erhalten sehr viele Informationen in einem kurzen Zeitraum. Zu dieser Art zählt das „Vortragen und Vorlesen“, das „Erklären und Beschreiben“, und das „Vorführen und Vormachen“.
- **Erarbeitendes Lehrverfahren**
  Hierbei ist die von der Lehrperson gestellte Frage das wichtigste Merkmal. Diese Frage sollte Denkprozesse insofern anstoßen, dass die Schülerinnen und Schüler am Ende das vermittelte Wissen verstanden haben.
- **Entdecken lassendes Lehrverfahren**
  Die Strukturierung ist beim endeckenden Lehrverfahren sehr gering. Die Lernenden sollen sich das Thema und Zusammenhänge selbst erarbeiten, die Lehrperson liefert hierbei nur Hilfestellungen.

2.3.5 Humor im Unterricht

Schülerinnen und Schüler bewerten Lehrpersonen nach ihrer Tüchtigkeit und der sozialen Beziehung zwischen ihnen. Dabei wird mit der Tüchtigkeit die fachliche und didaktische Kompetenz beschrieben, die der Lehrer oder die Lehrerin aufweist. Damit eine soziale Beziehung entstehen kann, werden von den Schülerinnen und Schülern eine Rücksichtnahme und allgemeine soziale Kompetenzen erwartet. Zudem wird hervorgehoben, dass Humor eine wichtige Rolle spielt. [22]

Aufgrund der Funktionsweise des menschlichen Gehirns werden Fakten längerfristig gespeichert, wenn der Hippocampus aktiv ist. Dieser Bereich wird bei Heiterkeit und einer positiven Stimmung angeregt. [12] Studien zeigen ebenfalls, dass durch den Einsatz von humoristischen Mitteln das Wissen besser abgespeichert wird. [23], [24]

2.4 Distanzlehre

2.4.1 Konzepte digitaler Medien

Für den Einsatz digitaler Medien in der Lehre gibt es drei mögliche Ansätze, die Schulmeister 2001 vorschlug. Diese sind das Anreicherungskonzept, das integrative Konzept sowie das virtuelle Konzept. Eine getrennte Betrachtung der einzelnen Ansätze ist möglich, in der Praxis werden sie zudem untereinander kombiniert. [10]

2.4.1.1 Das Anreicherungskonzept

Hierbei handelt es sich primär um das Abhalten einer Präsenzlehre mit der zusätzlichen Integration von digitalen Medien wie Bildern, Videos und Audiodateien. Ebenso können digitale Tools eingesetzt werden, die ein Einholen von Feedback und Input von Studierenden erleichtert. Mit Abstimmungssystemen kann die Meinung von den Teilnehmerinnen und Teilnehmern erhoben werden, Kleingruppen können in ihrem Arbeitsprozess Tools zum Brainstormen verwenden. Eine digitale Lernplattform bietet sich hier ebenso an und ermöglicht neben der Präsenzveranstaltung noch ein Bereitstellen von zusätzlichen Lernmaterialien, Vorlesungsskripten und weiteren Übungen und Selbsttests. Eine Abgabe und Bewertung von Hausübungen, als auch die Bekanntgabe von wichtigen Terminen kann mit einer digitalen Lernplattform einfach realisiert werden. [10]

2.4.1.2 Das integrative Konzept

Anders als beim Anreicherungskonzept, in dem die Online-Plattform ein zusätzliches Angebot darstellt, ist im integrativen Konzept, welches auch als *Blended Learning* bezeichnet wird, die digitale Lehre direkt mit den Präsenzterminen verzahnt. Die Onlinephasen haben dabei einen gleichwertigen Stellenwert und müssen mit den Präsenzeinheiten so abgestimmt werden, dass sich ein roter Faden durch beide Phasen zieht. Wie im Anreicherungskonzept lassen sich im integrativen Konzept auch digitale Medien in der Präsenzlehre integrieren. [10]

2.4.1.3 Das virtuelle Konzept

Beim virtuellen Konzept ist der Lehr- und Lernbetrieb vollständig onlinebasiert. Große amerikanische Universitäten wie die Harvard University und die Stanford University haben einen großen Teil dazu beigetragen, dass sogenannte MOOCs (Massive Open Online Courses) an Popularität gewinnen. Zusätzlich gibt es auch schon sehr viele Online-Anbieter, die kommerzielle Kurse für die Allgemeinheit anbieten. Diese bestehen meist aus Videos, Quizzes, verschiedenen Übungen und einer Möglichkeit zum Austausch zwischen den Lernenden. Bei reinen Online-Kursen ist es von großer Wichtigkeit, dass die Motivation hochgehalten wird, damit nicht zu große Drop-Out-Quoten entstehen. Diese ist bei rein virtuellen Kursen noch um einiges höher als bei Ansätzen, in denen Präsenzeinheiten enthalten sind. [10]

2.4.2 Didaktisches Design

Die verschiedenen Ebenen für das Gestalten der Lernprozesse sollten bei der Planung von Unterrichtseinheiten berücksichtigt werden. Dabei wird das Modell von Reinmann verwendet. Reinmann gliedert den Lehr-Lern-Prozess in drei Gestaltungsebenen: die Vermittlungsebene, die Aktivierungsebene und die Betreuungsebene. [10]

Vermittlung

Diese Komponente wird von Reinmann auch als materiale Seite in einem didaktischen Szenario bezeichnet. Dabei wird entschieden, welche Lernmaterialien in bestimmter Weise eingesetzt werden. Ob die Lerninhalte als Text oder Bild, Video oder Podcast vermittelt werden, ist dabei zu überlegen. In welchem Umfang die Medien eingesetzt werden und wie sie miteinander kombiniert werden, ist ebenfalls zu beachten. [10]

Aktivierung

Bei der Aktivierungsebene beschäftigt man sich mit der Gestaltung der Aufgaben. Die Kursteilnehmerinnen und Kursteilnehmer sollen sich mit den vermittelten Lerninhalten aktiv auseinandersetzen. Dabei kann je nach Aufgabengestaltung das bereits erhaltene Wissen gefestigt, als auch ein Transfer stattfinden, oder neues Wissen erlangt werden. [10]

Betreuung

Hierbei wird auf die tutorielle Betreuung und der Begleitung der Kursteilnehmerinnen und Kursteilnehmer geachtet. Neben der Gestaltung von Feedback an die Lernenden können Repetitorien gehalten und Lernräume bereitgestellt werden, damit ein Austausch zwischen den Personen ermöglicht wird. [10]

Abbildung 2: Gestaltungsebenen der Lehr-Lernziele nach Reinmann
Quelle: [10, S. 56]

2.4.3 Herausforderungen

Für eine gute Abhaltung der Distanzlehre sind, wie im oberen Abschnitt beschrieben, verschiedene Fertigkeiten notwendig. Als Tutoriums-Leiter und Tutoriums-Leiterin sollte man sich im Vorfeld über die notwendigen Skills bewusst sein, und die eigenen Schwächen und Stärken kennen. Die wichtigsten Aspekte werden in der folgenden Auflistung näher erläutert. [25]

- **Lehren im virtuellen Raum**
  Der natürliche Lehrraum ist für die unterrichtende Person nicht mehr gegeben, sondern findet virtuell statt. Die Teilnehmerinnen und Teilnehmer sind nicht miteinander in einem Raum, sondern können orts- und auch zeitungebunden teilnehmen. Durch diese Situation werden von beiden Seiten andere Kompetenzen als in der Präsenzlehre benötigt. [25]
- **Lehrkonzeption anpassen**
  In der Online-Lehre steht ein selbstgesteuertes Lernen im Vordergrund, wodurch die Lehrinhalte didaktisch angepasst werden müssen. Die Lernenden benötigen einen durchdachten Lernprozess, der verschiedene Unterrichtsformen wie kooperatives Lernen, Einzelarbeiten und Gruppenarbeiten beinhaltet. [25]

- **Onlinebasierte Lehrkompetenzen entwickeln**

  Für die Präsenzlehre benötigen Vortragende Kompetenzen, in der sie gruppendynamische Konflikte erkennen und beseitigen können. In der Online-Lehre können die Vortragenden diese alltäglichen Praxis-Kompetenzen nicht einsetzen. Daher ist eine Aneignung von onlinebasierten Lehrkompetenzen wichtig.

  Diese sind unter anderem ein durchdachter Gebrauch einer Lernplattform, der Einsatz verschiedener Medien, und synchrone (Virtual Classroom) als auch asynchrone Kommunikationsmittel (Foren). [25]
- **Diskurse zwischen Lehrenden und Lernenden initiieren**

  Für ein diskursives Lernen in der Online-Lehre ist eine angenehme Atmosphäre notwendig, die der Kursleiter bzw. die Kursleiterin schaffen muss. Die Person dient als Moderatorin oder Moderator, teilt den Lernenden klare Erwartungen und Anforderungen mit, und stellt zielführende Fragen zur Anregung einer Diskussion. Die technischen Rahmenbedingungen müssen dabei immer beachtet werden. [25]
- **Veränderte „Anwesenheits“-Zeiten**

  Bei der flexibleren Zeiteinteilung, die durch den Online-Unterricht ermöglicht wird, spricht Rolf Schulmeister von der „Virtualisierung der Zeit“. Die Lernenden können sich die Lernzeit beliebig einteilen, falls keine festen Termine gesetzt werden. [25]
- **Mehraufwand für die Lehrenden**

  Oft wird davon gesprochen, dass die Online-Lehre eine Zeitersparnis bringt, da man zeit- und ortsungebunden ist. Dies trifft jedoch nur für die Teilnehmerinnen und Teilnehmer zu. Da den Lernenden Kurs- und Lernmaterialien zur Verfügung gestellt werden sollte, müsse diese erstmal erstellt und in einer eingerichteten Lernplattform hochgeladen werden. Auch die Erstellung und Abhaltung von Prüfungen im Online-Setting erfordert einen höheren Aufwand für Lehrende. [25]

3 Gestaltungsprinzipien für Benutzeroberflächen

3.1.1 Einführung

Grundsätzlich sollten jede Software und jedes Gerät so funktionieren, dass sie jemand ohne Probleme benutzen kann und in schnellstmöglicher Zeit das gewünschte Ziel erreicht. Jedoch ist dies häufig nicht der Fall. Der Grund liegt an der fehlerhaften Interaktion zwischen dem Gerät und dem Anwender bzw. der Anwenderin. [26]

Interaktionsprobleme können weitreichende Folgen haben:

- Die Nutzer in einem Unternehmen können durch die schlechte Interaktion mit einem System frustriert und demotiviert werden. Die daraus folgende Demotivation kann zu Leistungseinbußen führen und senkt die zuvor berechnete Einsparung an Arbeitszeit und gewonnene Produktivität. [26]
- Bei einer unübersichtlichen Webseite verliert der Kunde bzw. die Kundin schnell das Interesse, die gewünschten Informationen zu suchen. Dies kann zum Kundenverlust, und einen daraus resultierenden Umsatzverlust führen. Übersichtlicher gestaltete Webseiten von Konkurrenz-Firmen können die Person ebenso abwerben. [26]
- Kundinnen und Kunden können bei der Bedienung eines Geräts überfordert und unzufrieden mit der Handhabung werden, wodurch beim nächsten Kauf ein Gerät einer anderen Marke ausgewählt wird. Bei großen Fehlern in der Handhabungsweise kann auch ein Imageverlust entstehen. [26]

Bei der Entwicklung einer Software oder eines Geräts ist es daher entscheidend, bereits ab der Planung Zeit und Ressourcen für die Erstellung einer sinnvollen Interaktion zwischen Bedienerin bzw. Bediener und dem System aufzuwenden. [26]

3.1.2 Historische Entwicklung

Überlegungen zu einer Schnittstelle zwischen „Mensch und Computer“ gab es bereits im Jahr 1945 von Vannevar Bush, der im Artikel „As We May Think“ ein fiktives System beschrieb, bei dem eine Sammlung von Informationen durch einen mechanischen Hebel durchsucht und verknüpft werden kann. Dieses System mit dem Namen Memex (Memory Extender) enthielt bereits Lösungsansätze von derzeitigen Programmfunktionen wie dem Durchblättern von Seiten, sowie dem Erstellen von Links und Notizen. [26]

John Licklider kritisierte in den 1960er Jahren die starre Benutzung von Programmen. Das Erstellen von Programmen, die danach von einem Batch-Prozess verarbeitet wurde und eine Liste als Ausgabe lieferte, war sehr inflexibel. Somit suchte er nach Lösungsansätze, die eine Kooperation zwischen dem Bediener bzw. der Bedienerin und dem Computer ermöglichen. [26]

1963 wurde von Ivan Sutherland das Sketchpad-System vorgestellt, mit dem eine grafische Interaktion mit dem Computer möglich war. Dabei konnte man mit einem Grafik-Tablet und einem Stift Skizzen zeichnen, die vom Computer automatisch analysiert und grafisch dargestellt wurden. [26]

Mit dem NLS-System (OnLine-System) wurde von Douglas Engelbart 1968 ein weiterer Schritt in Richtung der heutigen Mensch-Computer-Interaktion gesetzt. [27] Dieses System wies eine grafische Benutzeroberfläche auf, als Bedienung wurde jedoch kein Stift oder Pfeiltasten, sondern die heutige Maus verwendet. Der Vorteil lag hier einerseits bei der angenehmeren Bedienung über eine längere Dauer, und es kam zu keinem Verdecken der Bildschirmfläche bei der Navigation. [26]

Die Firma Xerox brachte mit dem XEROS Star ein System auf den Markt, das nicht von Programmiererinnen und Programmierern benutzt werden soll, sondern von Managerinnen und Managern, die nur gelegentlich den Computer verwenden. Hierbei wurden einige innovative Bedien-Möglichkeiten entwickelt, die noch in heutigen Betriebssystemen zu finden sind. Xerox Star ermöglichte das gleichzeitige Bedienen von mehreren Programmen, welche in eigenen Fenstern dargestellt wurden. Für ein schnelles Erlernen des Systems wurden generische Kommandos, wie *„Copy“* für das Kopieren, verwendet. Die Benutzeroberfläche glich einem Schreibtisch, auf dem sich verschiedene Dokumente befanden. Mehrere Dokumente wurden in Ordnern gesammelt und Anwendungen wie der Kalender waren als Symbol auf dem Schreibtisch sichtbar. Diese Symbole dienten als Link und führten beim Klick direkt zur Anwendung. [26]

Ein weiterer Meilenstein war der Apple Macintosh, der aufgrund seiner Benutzerfreundlichkeit kommerziell sehr erfolgreich war. Neuheiten dieses Systems waren einerseits die direkte Manipulation von Dateien und Ordnern mittels Drag-and-Drop und die Möglichkeit der Zwischenspeicherung von Texten, Dateien und Ordnern. Von Apple wurden ebenfalls Guidelines für die User-Interface-Entwicklung aufgestellt, die von Drittanbietern bei der Softwareentwicklung eingehalten wurden. Dadurch gab es in den verschiedenen Anwendungen einen einheitlichen Aufbau der Menüs, gleiche Tastaturkürzel für bestimmte Funktionen, und Anordnungen von Elementen im Fenster. Dies verkürzte die Einarbeitungszeit in die einzelnen Programme und steigerte infolgedessen die Benutzerfreundlichkeit. [26]

3.1.3 Richtlinien

3.1.3.1 Dialogprinzipien von Grice

Paul Grice stellte 1967 folgende Regeln für die Kommunikation auf: [28]

- **Be Informative**
  Vermittle nur Informationen, die für das Gegenüber sinnvoll sind. Bei einer Fehlermeldung des Druckers soll nicht der interne Fehlercode angezeigt werden, sondern ein für den User verständlicher Fehlertext.
- **Be Truthful**
  Gib nur Rückmeldungen an den Informationspartner, die auch wahr sind. Bei fehlendem Papier im Drucker soll nicht als Fehlermeldung „Der Drucker funktioniert nicht mehr“ angezeigt werden.
- **Be Relevant**
  Stelle nur Informationen dar, die für den User auch relevant sind. Überflüssige Informationen stellen nur einen Mehraufwand bei der Bedienung dar und sollte vermieden werden. Der Kontextmenüpunkt *Ausschneiden* sollte nur auswählbar sein, wenn zuvor ein Objekt ausgewählt wurde.
- **Be Perspicuous**
  Verwende eine einfache Sprache bei Meldungen an den User und verzichte auf Fachbegriffe der Informatik.

3.1.3.2 Die acht goldenen Regeln von Ben Shneiderman

Der Software-Ergonomie-Pionier Ben Shneiderman definiert aufgrund seiner Erfahrung folgende acht Regeln für eine gute Benutzerschnittstelle: [28]

- **Konsistenz**
  Verwende Styleguides, damit sich alle Programmiererinnen und Programmierer sowie Designerinnen und Designer auf eine einheitliche Gestaltung der einzelnen Bestandteile eines Systems festlegen und damit ein konsistentes Design gegeben ist. Hierbei ist nicht nur ein einheitliches Aussehen der Elemente gemeint, sondern auch die Wortwahl der Menüs und Texte, der Schriftart, Farben und Schriftgröße sowie die gleiche Abfolge von bestimmten Aktionen.
- **Berücksichtigung unterschiedlicher Erfahrungen**
  Bei der Anwendung sollten die Bedürfnisse von Anfängerinnen und Anfänger sowie von fortgeschrittenen Personen berücksichtigt werden. Für Personen, die das Programm zu ersten Mal bedienen, sollten Aktionen über klar verständliche Menüoptionen erreichbar sein. Fortgeschrittene möchten ihren Workflow verbessern, indem sie über Shortcuts Befehle schneller ausführen können.

- **Rückmeldungen auf Aktionen**
  Für den Benutzer und die Benutzerin ist es wichtig, dass auf eine durchgeführte Aktion eine Rückmeldung erscheint. Bei einem Klick auf einen Button kann als Rückmeldung eine Schattierung des Buttons, oder eine Änderung des Textes ausreichen. Falls jedoch ein Fehler auftritt, sollte der Benutzer mit einer eigenen Mitteilung darauf aufmerksam gemacht werden. Hierbei ist es wichtig, dass nicht nur der Fehler selbst mitgeteilt wird, sondern auch die mögliche Fehlerbehebung. Akustische Meldungen sollten so selten wie möglich eingesetzt werden, damit keine Reizüberladung auftritt und dadurch die Benutzerfreundlichkeit sinkt. Die taktile Rückmeldung gehört zum Bereich der Hardware-Ergonomie und wird beispielweise bei der Tastatur eingesetzt. Hierbei bekommt der User eine taktile Rückmeldung über die Tastatur, sobald die Taste eine bestimmte Drucktiefe erreicht, und das Signal an den Computer weitergeleitet wurde.
- **Abgeschlossene Operation**
  Wenn mehrere Aktionen eine zusammenhängende Operation bilden, sollte der User zu jedem Zeitpunkt über den aktuellen Stand informiert werden. Nach einem erfolgreichen Abschluss der Operation sollte der Bediener eine Zusammenfassung der einzelnen Schritte erhalten.
- **Fehlerverhinderung**
  Eine sinnvolle Fehlermeldung ist wichtig, der Bediener sollte jedoch im Idealfall keine Fehlermeldung erhalten. Eine Möglichkeit ist das Ausgrauen von Bedienelementen, die beim Auswählen zu einem Fehler und dadurch zu einer Fehlermeldung führen würden. Beim Eingeben eines Datums kann eine Darstellung des Kalenders und die visuelle Auswahl des Datums eine falsche Formateingabe verhindern. Beim Ausfüllen eines Formulars mit vielen Eingabefeldern sollte beim Auftreten eines Fehlers nur noch die fehlerhafte Eingabe leer sein und die bereits korrekt ausgefüllten Felder erhalten bleiben.
- **Einfache Rücksetzmöglichkeiten**
  Um die Selbstsicherheit zu erhöhen und auch das explorative Lernen zu fördern, sollte jede ausgeführte Aktion im Programm rückgängig gemacht werden können (UNDO). Dadurch fühlen sich Benutzer in der Bedienung sicherer und haben den Mut, Neues auszuprobieren. Ebenso sollte auch das Wiederherstellen der zurückgesetzten Aktion (REDO) möglich sein.
- **Benutzerbestimmte Eingaben**
  Bei der Anwendung sollte der User stets das Gefühl haben, die Software zu kontrollieren. Daher sollte man dem Benutzer und der Benutzerin jederzeit die Möglichkeit bieten, die Aktion abbrechen zu können.

- **Kurzzeitgedächtnis nur gering belasten**
  Der Anwender und die Anwenderin sollten in der Bearbeitung ihrer Aufgabe so wenig Informationen wie möglich in ihrem Kurzzeitgedächtnis speichern müssen. Daher ist es vorzuziehen, die Benutzeroberfläche mit wenigen Informationen zu befüllen. Menübäume werden für eine bessere Übersicht eher breit als tief angelegt, da dadurch die Navigation und das Merken der Standorte erleichtert wird.

3.1.3.3 Die zehn Usability-Heuristiken von Nielsen

Für eine Minimal-Gebrauchstauglichkeit einer Software definiert Jakob Nielsen zehn Prinzipien. Die Einhaltung dieser bei der Entwicklung wird auch *Discount Usability Engineering* genannt. [28]

- **Einfache und natürliche Dialoge**
  Bei Dialogen sollten nur relevante Informationen und Entscheidungsmöglichkeiten dargestellt werden. Die Anzahl an Informationen sollte so gering wie möglich gehalten werden.
- **Ausdrucksweisen des Anwenders**
  Der Anwender und die Anwenderin, und nicht die Softwareentwickler und Softwareentwicklerin, müssen die Bezeichnungen der Elemente kennen und einordnen können. Daher ist für die Benennung der Elemente ein Experte oder eine Expertin aus dem Fachgebiet heranzuziehen.
- **Minimale mentale Belastung des Benutzers**
  Da das Kurzzeitgedächtnis nur eine gewisse Kapazität aufweist, sollte die Anzahl an Informationen so gering wie möglich gehalten werden. Auch farbliche Hervorhebungen bringen hier nur bedingt eine Verbesserung, da jedes zusätzliche sichtbare Element das Risiko erhöht, dass der Anwender oder die Anwenderin für die gewünschte Aktion länger suchen muss oder etwas übersieht.
- **Konsistenz**
  Routinen und Darstellungen im Programm sollten immer der gleichen Logik folgen. Bei Dialogfenstern mit Auswahlmöglichkeiten ist darauf zu achten, dass die Anordnung der jeweiligen Reaktion, z.B. der Ok-Button, immer an der gleichen Stelle erfolgt.
- **Rückmeldungen**
  Bei jeder Aktion, die der Benutzer oder die Benutzerin ausführt, sollte eine Rückmeldung in Form einer veränderten Farbe, eines akustischen Signals oder einer taktilen Rückmeldung stattfinden. Bei längeren Aktionen sollte der User über den aktuellen Fortschritt sowie die noch notwendige Dauer informiert werden.

- **Klare Auswege**
  Bei einer Ausführung müssen der Bediener und die Bedienerin die Möglichkeit zum Ausstieg, bzw. zur Rückkehr zum Startpunkt haben. Ebenfalls müssen ausgelöste Aktionen, wie zum Beispiel dem Drucken von 100 Kopien, abgebrochen werden können.
- **Abkürzungen**
  Abkürzungen in Form von Shortcuts erleichtern fortgeschrittenen Usern den Ablauf und steigern die Produktivität. Eine Historie über die letzten Eingaben ist ebenfalls hilfreich und minimiert die Fehleranfälligkeit.
- **Gute Fehlermeldungen**
  Eine Fehlermeldung soll zusätzlich zur Art des Fehlers auch die mögliche Behebung anzeigen. Der Text sollte in einer für den User verständlichen Sprache gehalten werden, damit keine Verwirrung oder Überforderung auftritt. Eine Möglichkeit ist, durch Aufklappen der Meldung noch weitere, spezifische Informationen für den Administrator oder die Administratorin anzeigen zu lassen.
- **Fehlervermeidung**
  Bevor Fehler auftreten, sollten diese gänzlich vermieden werden. Dies ist durch verständliche Hinweise, Standardwerten oder Auswahllisten möglich.
- **Hilfe und Dokumentation**
  Eine Dokumentation und Hilfe ist für eine gute Software unerlässlich. Die Hilfe liefert dem Benutzer für die derzeitige Situation weitere Informationen. In der Dokumentation befindet sich eine vollständige Auflistung aller Funktionen und deren Erklärung. Die Dokumentation und Hilfe muss auch immer an den aktuellen Softwarestand angepasst werden.

3.1.3.4 Relevanz für das Tutorium

Die im oberen Abschnitt beschriebenen Richtlinien sollten bei der Entwicklung einer Software immer berücksichtigt werden. Die meisten Studierenden kommen mit diesen Richtlinien zum ersten Mal in Berührung. Die von den Studierenden entwickelten Apps werden auch hinsichtlich dieser Kriterien bewertet. Es ist daher wichtig, das Tutorium so zu planen, dass die Richtlinien von Grice, Shneiderman und Nielsen beachtet, und in den veranschaulichten Programmierbeispielen auch angewendet werden.

3.1.4 Gedächtnis und Erfahrung

Bei einem sinnvollen User-Interface ist zu beachten, dass die notwendigen Schritte von der benutzenden Person im Gedächtnis gespeichert werden müssen. Inwiefern das menschliche Gehirn Informationen speichert, ist daher ein relevanter Faktor in der Konzipierung des Interfaces. [26]

Generell gibt es drei unterschiedliche Speicher bei der menschlichen Informationsverarbeitung.

Sensorischer Speicher

Der sensorische Speicher kann eine große Menge an Informationen aufnehmen. Die Informationen sind sensorische Reize wie die Tonhöhe, Töne, Farben und Formen. Die visuellen Signale bleiben rund 0,2 Sekunden im Speicher, auditive für 1,5 Sekunden. Ein Auffrischen der Information ist nur durch den erneuten Stimulus möglich. [26]

Arbeitsgedächtnis

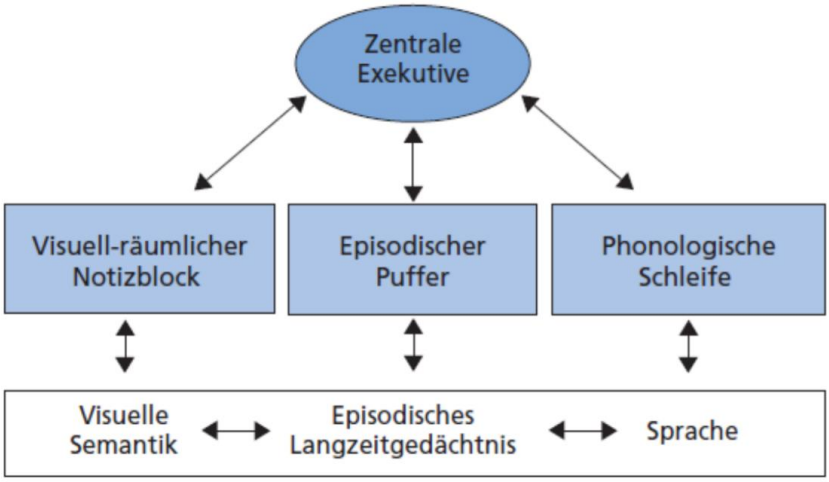

Abbildung 3: Modell des Arbeitsgedächtnis
Quelle: [29, S. 3]

In früheren Modellen über das menschliche Gedächtnis ging man von einem Kurzzeitgedächtnis aus, das separat vom Langzeitgedächtnis Informationen speichert. Dies wurde mittlerweile widerlegt. Der Bereich für das kurzzeitige Speichern wird seither Arbeitsgedächtnis genannt und teilt sich in drei Bereiche auf: Der visuell-räumliche Notizblock, der episodische Puffer und die phonologische Schleife. Der visuell-räumliche Notizblock speichert Informationen zu Farben, Formen und Anordnungen, währen die phonologische Schleife für die sprachliche Information zuständig ist. [26]

Der episodische Puffer ist in enger Kopplung mit dem episodischen Langzeitgedächtnis. Er ruft dort Informationen ab und vergleicht sie mit den Daten der phonologischen Schleife und dem visuell-räumlichen Notizblock. Für die Koordinierung der drei Bereiche ist, wie in Abbildung 3 ersichtlich, die zentrale Exekutive verantwortlich. [29] Das Arbeitsgedächtnis ist für die kurze Aufbewahrung von Informationen zuständig und kann nach neuesten Untersuchungen bei Erwachsenen drei bis vier Informationseinheiten speichern.

Dabei ist es möglich, dass einzelne Informationen zu komplexen Informationseinheiten, sogenannte *Superzeichenbildung* (*chunking*), gebündelt werden. Eselsbrücken sind ein gutes Beispiel solcher Superzeichen. [26]

In der Mensch-Computer-Interaktion ist die Leistung des Arbeitsgedächtnisses zu beachten, damit die Bedienerinnen und Bediener nicht mit zu vielen Aufgaben überfordert werden. Bei einer Überlastung des Arbeitsgedächtnisses treten Fehler in der Bedienung auf, wodurch dies zu vermeiden ist. Wenn eine Aufgabe erledigt wurde, soll der Benutzer bzw. die Benutzerin eine Information vom System erhalten. Dies schafft wieder neuen Platz im Arbeitsgedächtnis, da das Gehirn durch eine solche Meldung weiß, dass der Task abgeschlossen ist. [26]

Langzeitgedächtnis

Das Langzeitgedächtnis ist für die mittel- bis längerfristige Speicherung von Informationen zuständig, und kann diese einige Minuten bis mehrere Jahrzehnte speichern. Inwiefern die Information erhalten bleibt, hängt nicht wie nach früherer Meinung von der Wiederholung der Information ab, sondern von der Tiefe der Verarbeitung. Mit der Tiefe wird hierbei verstanden, wie sehr die neue Information mit bereits bekannten Informationen verknüpft werden kann. Wenn Informationen vergessen werden, wurden diese nicht gelöscht. Stattdessen bestehen sehr wenige bis keine Verbindungen zu den anderen Informationen, wodurch kein Zugriff mehr möglich ist. [26]

Beim Zugriff auf das Langzeitgedächtnis gibt es zwei Formen, das Erinnern und das Erkennen. Das Erinnern, an beispielsweise eine Telefonnummer, erfordert das vollständige Abrufen der gesamten Nummer aus dem Gedächtnis. Bei der Suche nach einer bestimmten Datei muss jedoch nur der Dateiname erkannt werden. Dies ist für den Benutzer und die Benutzerin um ein Vielfaches leichter. Daher sollte eine Software so gestaltet werden, dass die benutzende Person die gesuchten Informationen erkennen kann und sich nicht daran erinnern muss. [26]

Relevanz für das Tutorium

Die Speicherung von Informationen im menschlichen Gehirn spielt eine Rolle bei der Entwicklung von Apps. Wie auch bei den Richtlinien im vorherigen Kapitel ist es als vortragende Person im Tutorium wichtig, diesen Bereich zu beachten. Die Studierenden sollen bei den im Tutorium gezeigten Beispielen erfassen, an welchen Stellen ein Erkennen der Information sinnvoll ist, und wo ein Erinnern ausreicht.

3.1.5 Wahrnehmung

Visuelles System

Für die visuelle Informationsaufnahme ist die Retina des menschlichen Auges zuständig, sie verarbeitet die Intensität und Wellenlänge des einfallenden Lichts. Diese Daten werden in der Folge im sensorischen Speicher zwischengelagert. Der Bereich für die visuelle Speicherung wird auch ikonischer Speicher genannt, und kann insgesamt zehn Einheiten von Ziffern und Buchstaben für rund 300 Millisekunden zwischenspeichern. Diese Informationen werden in weiterer Folge an das Gehirn weitergeleitet, das aus diesen ein mentales Bild erstellt. Dieses Bild ist kein objektives Abbild der Realität, sondern von den Zielen und Erwartungen der beobachtenden Person geprägt. Daher können sich auch Augenzeugenberichte grundlegend voneinander unterscheiden, da die Informationen von den Personen unterschiedlich interpretiert werden. [26]

Visuelle Suche

Häufig nimmt die Suche von bestimmten Bedienelementen und Informationen einen großen Teil der Zeit ein. Die Suchdauer, die für das Finden des gewünschten Ziels benötigt wird, hängt von mehreren Faktoren ab. Einerseits kann die Anzahl der anderen verfügbaren Objekte die Dauer der Suche verlängern. Weiters kann die visuelle Unterscheidung der einzelnen Objekte die Suchdauer beeinflussen. Falls das Zielobjekt ein auffälliges Merkmal aufweist, lässt sich dieses unabhängig von der Anzahl an anderen Objekten schneller finden. [26]

Präattentive Wahrnehmung

Als präattentive Wahrnehmung wird die Fähigkeit genannt, bestimmte Objekte aufgrund eines stark sichtbaren Unterscheidungsmerkmals sofort zu erkennen. In der Abbildung 4 (A) der ist rechte Winkel sehr schnell unter den vielen Kreuzen zu finden. In der Abbildung 4 (B) ist der Unterschied des rechten Winkels zum Buchstaben T nicht so gravierend, dadurch ist keine präattentive Wahrnehmung möglich und ein Durchsuchen der einzelnen Symbole erforderlich. In der Mensch-Computer-Interaktion wird ein solches Hervorheben häufig eingesetzt. [30]

Ein Beispiel findet sich in einem Textverarbeitungsprogramm, das Rechtschreibfehler mit einer gewellten roten Linie von den anderen Wörtern visuell hervorhebt. [26]

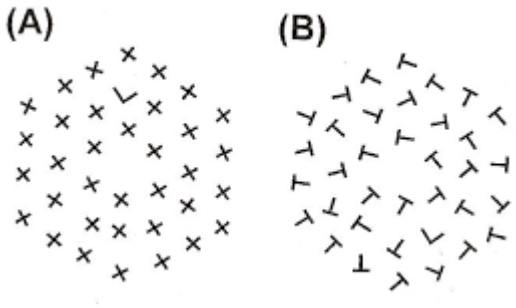

Abbildung 4: Präattentive Wahrnehmung
Quelle: [30, S. 4]

Farbwahrnehmung

Die Farbwahrnehmung ist ein wichtiger Aspekt der Benutzeroberflächenerstellung. Stark aufmerksamkeitslenkende Farben sollen sparsam eingesetzt werden. Der Kontrast zwischen dem Vorder- und Hintergrund kann die Wahrnehmung ebenfalls beeinflussen. Ein hoher Farbkontrast lässt das Objekt im Vordergrund kleiner erscheinen. Dunkle Farben werden vom Menschen entfernter wahrgenommen. Aufgrund der Anatomie des Auges, das eine zehnfach höhere Anzahl an Rezeptoren zur Erkennung von Grün- und Rottönen als von Blautönen aufweist, können Abstufungen der blauen Farbe geringer erkannt werden, wodurch diese Farbe in der Benutzeroberfläche nicht so gut geeignet ist. [26]

Wahrnehmungsfehler

Bei der Wahrnehmung von Farben und Formen kann es auch zu Fehlern kommen, wodurch sich das mentale Abbild von der Realität unterscheidet. Es wird vermutet, dass die wahrgenommenen Informationen hierbei nicht erst im Gehirn zusammengesetzt werden, sondern bereits auf der Netzhaut verarbeitet werden. Im folgenden Absatz werden einige solcher Effekte genauer beleuchtet. [28]

Simultankontrast

In der Abbildung 5 wirkt das linke Rechteck im grauen Hintergrund heller als das Rechteck im schwarzen Hintergrund. Dieser Effekt wird als Simultankontrast bezeichnet und entsteht durch eine sogenannte laterale Hemmung. Dabei wirkt sich der Reiz eines Pixels, der in einer Sehzelle entsteht, hemmend auf die benachbarten Zellen aus. [28]

Abbildung 5: Simultankontrast
Quelle: [31, S. 1]

Hermann-Gitter

Die laterale Hemmung kann auch im Hermann-Gitter beobachtet werden. In Abbildung 6 erscheinen die Kreuzungspunkte der weißen Linien dunkler, da dieser Bereich im Auge von den vier benachbarten schwarzen Quadraten gehemmt wird.[28] [32]

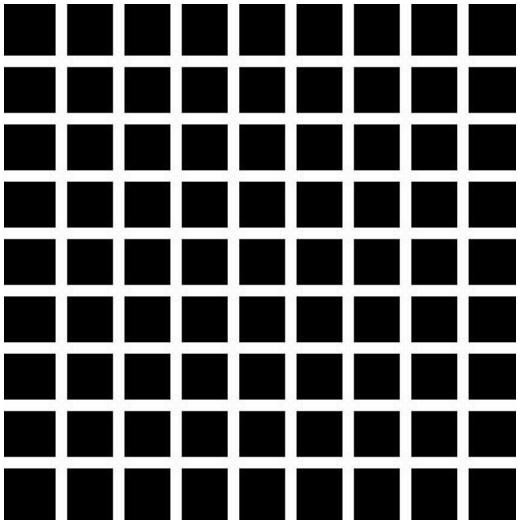

Abbildung 6: Hermann-Gitter
Quelle: [32, S. 1377]

Relevanz für das Tutorium

Das visuelle System spielt bei der Bedienung einer App eine große Rolle. Als Tutoriumsleiter bzw. Tutoriumsleiterin sollte man den Studierenden in den Einheiten den Grund für die Farbwahl und die Anordnung der Elemente erklären können. Das theoretische Wissen unterstützt einem dabei in der Vermittlung.

3.1.6 Usability Engineering

Bei der Entwicklung einer guten Software wird nicht direkt mit der Programmierung begonnen, sondern bereits zuvor ein Entwurf und eine Planung über die Software entwickelt.

Ein Einbinden der späteren Anwenderinnen und Anwender in die Entwicklungs- und Planungsphase ist sehr wichtig, da ansonsten an den Bedürfnissen und Anforderungen „vorbeiprogrammiert“ wird. Für diesen Entwicklungsprozess gibt es verschiedene Modelle. [28]

Wasserfall-Modell

Das Wasserfall-Modell lässt sich sehr einfach umsetzen und wird derzeit noch in vielen Projekten eingesetzt, da es klare Phasen und eine feste Aufgabenverteilung gibt. Eine neue Phase beginnt, sobald in der vorherigen Phase alle Aktivitäten abgeschlossen sind. Es sind auch Rücksprünge zu den vorherigen Phasen möglich. Die einzelnen Phasen in diesem Modell lauten: [33]

1. **Analyse und Anforderungserhebung**
   In enger Absprache mit den späteren Anwenderinnen und Anwendern werden die Ziele und detaillierten Anforderungen des Systems definiert.
2. **System- und Softwareentwurf**
   In dieser Phase wird die grundlegende Software-Architektur festgelegt und die Beziehungen und Abhängigkeiten der einzelnen Systeme beschrieben.
3. **Implementierung Komponententest**
   Der Softwareentwurf wird nun in einzelne Programmteile zerlegt und programmiert, weiters werden die einzelnen Komponenten mit Tests geprüft.
4. **Integration und Systemtest**
   Nachdem die einzelnen Komponenten getestet wurden, werden diese als Ganzes zusammengesetzt und das System auf deren Richtigkeit getestet. Nach dem erfolgreichen Systemtest wird die Software an den Kunden ausgeliefert.
5. **Betrieb und Wartung**
   Nach der Installation beim Kunden erfolgt die Wartung, bei der Fehler behoben werden und notwendige Änderungen in der Software beinhaltet.

Spiralmodell

Das Spiralmodell enthält 4 Schritte, die mindestens einmal durchlaufen werden. Es kann jedoch, je nach Komplexität und unbekannten Anforderungen, auch zu mehreren Durchläufen kommen. [33]

1. Für den Durchlauf werden Ziele und weitere Rahmenbedingungen für das System definiert, sowie mögliche Risiken aufgelistet.
2. Die im vorherigen Schritt erkannten Risiken werden genauestens analysiert und Möglichkeiten zur Risikoverminderung vollzogen.

3. Anhand der Risiken wird ein Entwicklungsmodell ausgewählt und das System erstellt.
4. Das Projekt wird auf dessen Vollständigkeit überprüft und entschieden, ob ein weiterer Durchlauf der Spirale notwendig ist.

Prototyping

Beim Prototyping werden die Anwenderinnen und Anwender bei der Entwicklung der Software direkt miteinbezogen. Dazu werden Vorabversionen erstellt, die der End-User testet und Feedback gibt. Der Prototyp kann einerseits die Software selbst in einem frühen Stadium sein, aber auch ein *Mock-Up*, das den Programmablauf nur simuliert. Bei der Prototypentwicklung gibt es drei Arten: [33]

1. **Explorativ**
   Hierbei werden die Eigenschaften und Anforderungen des Systems definiert und verschiedene Lösungsmöglichkeiten diskutiert.
2. **Experimentell**
   Bei dieser Art sind die Anforderungen schon bekannt und es werden verschiedene Lösungsmöglichkeiten erstellt.
3. **Evolutionär**
   Dabei wird das System immer wieder neu angepasst und es werden notwendige Änderungen durchgeführt.

Die explorative und experimentelle Art wird für eine Sammlung an verschiedenen Prototypen verwendet, wobei hier der Großteil an Prototypen wieder verworfen wird. Bei der evolutionären Prototypentwicklung wird ein System stets weiterentwickelt. [33] Bei den Prototypen gibt es verschiedene Designs, die je nach Verwendungszweck sinnvoll sind: [34]

1. **Horizontales Prototyping**
   Dabei werden nur spezielle Bereiche eines Systems als Prototyp implementiert. Häufig werden für die Benutzeroberfläche mehrere Prototypen entwickelt, die „dahinterliegenden Schichten“ werden simuliert.
2. **Vertikales Prototyping**
   Beim vertikalen Prototyping wird ein Teil des Gesamtsystems vollständig umgesetzt. Dabei sind auch Komponententests möglich. Ob hierbei jedoch das System als Ganzes funktioniert, ist nicht evaluierbar.

Vorteile des Prototyping

Die Vorteile des Prototyping liegen bei einer hohen software-ergonomischen Qualität des Systems. Da die Anwenderinnen und Anwender aktiv bei der Erstellung mitwirken, ist auch die Akzeptanz eine wesentlich höhere. Ebenso verringern sich die Schulungskosten, sowie die Wartungs- und Weiterentwicklungskosten. [34]

Nachteile des Prototyping

Als Nachteil werden die höheren Entwicklungskosten gesehen, da für die Erstellung von verschiedenen Prototypen deutlich mehr Zeit benötigt wird. [34]

Personas

Um die Fülle an Anforderungen für das System zu analysieren, hat Alan Cooper 1998 das Konzept der Personas vorgestellt. Dabei werden fiktive Personen mit bestimmten Eigenschaften erstellt und deren mögliche Anforderungen und Wünsche an das System erhoben. Die jeweiligen Persona werden kurz beschrieben, ein Profilbild verstärkt nochmals die Vorstellung des jeweiligen Charakters. Bei der Gestaltung des Systems kann sich der Entwickler und die Entwicklerin fragen: „Wie würde nun Frau X in diesem Menü navigieren?“. Die Kosten für die Erstellung von Personas ist sehr gering, wodurch sie sehr häufig eingesetzt werden. [28]

Evaluation

Für die Evaluation von Prototypen und Software-Vorabversionen bieten sich verschiedene Testarten an.

- **Expertentests**
  Hierbei wird das System von erfahrenen Experten getestet. Diese können durch ihre langjährige Erfahrung viele Einschränkungen in der Gebrauchstauglichkeit als auch Verbesserungsmöglichkeiten erkennen, die ungeübte Anwenderinnen und Anwender übersehen. [28]
- **Benutzertests**
  Bei den Benutzertests wird das System von Personengruppen geprüft, die diese in der realen Welt direkt nutzen. Fehler in der Handhabung oder Unklarheiten, die in der Arbeitswelt auftreten, können somit direkt erkannt werden. Bei den Testungen ist man jedoch nach der jeweiligen Tagesform, Motivation und Nervosität der Testerinnen und Tester abhängig. Zu dieser Art der Tests zählen ebenfalls Interviews und Fragebögen. [28]

- **Quantitative Tests**
  Mit dieser Testart werden quantitative Werte wie die Anzahl an Klicks pro Aufgabenstellung, die Zeit für eine Aufgabenstellung, Fehlklicks und andere messbare Werte erhoben. [28]
- **Qualitative Tests**
  Um Probleme in der Bedienbarkeit von den Testerinnen und Testern besser verstehen zu können, eignen sich qualitative Tests, in denen die Personen ihre Schwierigkeiten per Interview oder Fragebogen in eigenen Worten erklären. [28]

Relevanz zum Tutorium

Die Lehrveranstaltungsteilnehmerinnen und Lehrveranstaltungsteilnehmer müssen im Gruppenprojekt mehrere Entwürfe der App erstellen, Prototypen entwickeln und diese evaluieren. Dies kann in der Erstellung des Tutoriums berücksichtigt werden, indem die Vorteile von Flutter beim Erstellen von Prototypen hervorgehoben, und in einem Beispiel veranschaulicht werden.

4 Cross Platform Frameworks

Im Tutorium wird das Cross-Platform Framework Flutter sowie die dazugehörige Sprache Dart vorgestellt und deren Einsatzmöglichkeiten zur Erstellung von Apps vermittelt. Für die Lehre einer Programmiersprache wird ein notwendiges Basiswissen über die Semantik und Syntax benötigt. Damit man den Vorteil von Flutter gegenüber anderen Technologien argumentieren kann, hilft es, einen Überblick über die Geschichte von Cross-Platform Frameworks sowie die Funktionsweise konkurrierender Technologien zu haben.

4.1 Entstehung von Cross-Platform Frameworks

Durch die Weiterentwicklung von mobilen Telefonen zu sogenannten Smartphones, gab es die Möglichkeit, Softwarelösungen zu programmieren, die auf dem Endgerät ausgeführt werden. Diese sogenannten *Apps* wurden bei Benutzerinnen und Benutzern außerordentlich beliebt. [35] Durch diese große Zahl an neuen Apps nahmen zwei große Stores, die als zentrales Register der Apps für die jeweilige Plattform dienen, der Google Play Store für das Android-System, sowie der Apple App Store für das iOS-System, den Großteil der Sparte ein. Die Betriebssysteme hinter den beiden App-Stores waren 2018 auf über 99% der verkauften Smartphones installiert. Zu dieser Zeit waren über 5,8 Millionen verschiedene Apps verfügbar, die Zahl der Nutzer betrug über 3,8 Milliarden Personen weltweit. [35]

Aufgrund der Verschmelzung verschiedener Branchen wie dem Internet of Things (IoT), technischen Geräten mit smarter Steuerung, neuesten Autos und der Smartphone- und Tabletindustrie werden für eine Software mehrere Entwicklungen benötigt, damit diese auf den verschiedenen Systemen installiert und ausgeführt werden können. [35]

Ursprünglich gab es nur die Möglichkeit, für jedes Betriebssystem eine eigene Software zu erstellen. Diese Form der Software-Entwicklung wird als *native development approach* bezeichnet. Für die Entwicklung der beiden großen Betriebssysteme wurden daher Programmierinnen und Programmierer für Android- als auch für iOS-Apps benötigt. Weitere Probleme ergeben sich durch die verschiedenen Betriebssystem-Versionen. Bei einem Update des Betriebssystems benötigen auch die jeweiligen Apps in vielen Fällen ein Update, durch das ein weiterer Entwicklungsaufwand und somit Kosten entstehen. [35]

Durch diesen zusätzlichen Kostenaufwand entstanden viele neue Möglichkeiten zur Kostenminderung, die unter dem Namen *cross-platform mobile development* zusammengefasst werden. Dabei wird nur eine Codebasis erstellt, die zur Erstellung von Apps unterschiedlicher Betriebssysteme verwendet werden kann. Dies bietet einen großen Vorteil gegenüber dem *native development approach,* da sich die Entwicklungskosten verringern.

4.2 Derzeitige Technologien

Die folgende Auflistung liefert einen Überblick über die derzeit meistgenutzten Technologien. Wichtig hierbei ist, dass es noch weitere, meist unbekanntere, technologische Lösungen gibt, auf diese jedoch nicht näher eingegangen wird.

4.2.1 (Progressive) Web Apps

Hierbei wird mittels HTML, CSS und JavaScript eine mobile Webseite erstellt, die für die Darstellung auf dem Smartphone optimiert wurde. Grundsätzlich können Webseiten nicht lokal installiert werden, sondern werden über einen Browser aufgerufen und dargestellt. Damit die *User Experience* verbessert wird, wurden sogenannte Progressive Web Apps (PWA) eingeführt. Diese beinhalten im Gegensatz zu den bisherigen Webseiten zusätzliche Funktionen wie *service worker*, die eine Codeausführung im Hintergrund durch eigene Threads ermöglicht. Die Nutzungsmöglichkeiten wurden durch diese neue Technologie um ein Vielfaches erhöht, eine Limitierung ist jedoch noch immer durch den jeweiligen Browser und dessen Zugriffe auf das Gerät vorhanden. [35]

4.2.2 Hybrid Apps

Cordova, ein Open-Source Framework, war die erste Lösung, die diesen Ansatz verfolgte. Dabei wird eine WebView mit HTML, CSS und JavaScript erstellt, die einzelne Komponenten eines Browserfensters, wie der Adressleiste und der Lesezeichenleiste, ausblendet. Diese Webansicht hat zusätzlich die Möglichkeit, auf das Adressbuch, GPS, Bluetooth und andere Gerätedienste mittels einer eigenen API zuzugreifen. Mit dem Framework kann aus der WebView eine App erstellt werden, die für Nutzerinnen und Nutzer über einen App-Store verfügbar ist. Ein Nachteil dieses Lösungsansatzes ist das veränderte Nutzungserlebnis, und somit bei der Bedienung einen Unterschied zu nativen Apps erkennen lässt. [35]

4.2.3 Runtime-based and Interpreted Apps

Bei dieser Technologie wird mittels einer Programmiersprache wie C# oder JavaScript auf eine API zugegriffen, bei der im Hintergrund eine eigene Runtime-Schicht läuft und auf die einzelnen Komponenten des Smartphones zugreift. Der Vorteil hierbei ist, dass die visuelle Komponente je nach Betriebssystem anders dargestellt wird und das Nutzererlebnis sehr ähnlich zu einer nativen App ist. Die bekanntesten Frameworks dieser Technologie-Reihe sind React Native von Facebook, NativeScript der Firma „Telerik by Progress“ und Xamarin, das von Microsoft entwickelt wurde. Ein Wechsel zwischen den einzelnen Frameworks ist im späteren Entwicklungsverlauf nur noch mit einem großen Arbeitsaufwand möglich. [35]

4.2.4 Model-driven Software Development

Bei der modellgetriebenen Softwareentwicklung wird der Quellcode auf Basis eines erstellten Modells erzeugt. Dabei kann eine domänenspezifische Sprache (DSL) oder eine universelle Modellierungssprache wie die Unified Modeling Language (UML) verwendet werden. Für jedes Betriebssystem kann dazu ein eigener Sourcecode erstellt, und in weiterer Folge zu einer vollständigen Applikation kompiliert werden, die die gleichwertige Funktionalität einer nativen App aufweist. Im kommerziellen Bereich sind WebRatio Mobile, bubble und BiznessApps die bekanntesten Frameworks, im akademischen Bereich werden MD2, applause und MAML verwendet. Der Ansatz der modellgetriebenen Softwareentwicklung wird in der Praxis jedoch sehr selten verwendet. [35]

4.3 Dart

Dart wurde von Google entwickelt und sollte ursprünglich als Sprache für den Webbrowser Chrome dienen. Das weckte aber Befürchtungen bei den Entwicklerinnen und Entwicklern, da dies eine Konkurrenz zu Javascript darstellte und dies nicht im Einklang mit den Bemühungen zu einer Vereinheitlichung der Kompatibilität von Web-Browsern stand. [36]

Anfang 2018 wurde von Google das neue Framework Flutter vorgestellt, in welcher die Programmierung ausschließlich in Dart erfolgte. Die Programmiersprache wird ständig weiterentwickelt und liegt nun bereits in der Version 2 vor. Hierbei hat sich Dart von einer Art Javascript zu einer eigenen Sprache entwickelt, die starke Ähnlichkeiten zu C# und Java aufweist. [36]

Dart findet auf mobilen Geräten, Konsolen und im Web Verwendung. Bei mobilen Geräten wird Dart in einer virtuellen Maschine, die einen Just-in-Time Compiler enthält, ausgeführt.

Flutter verwendet diese Virtual Machine, wodurch bei der Entwicklung ein „Hot-Reloading“ möglich ist. Dabei werden Änderungen im Programmcode sofort in der laufenden App sichtbar. Für die Veröffentlichung der Applikation wird der Dart-Code mit einem Ahead-Of-Time-Compiler zu einem nativen ARM-Code kompiliert. [36]

In der Web-Umgebung wird der geschriebene Dart-Code in Javascript kompiliert, wodurch die App dann im Browser lauffähig ist. Dart kann ebenfalls als kompilierte oder auch als interpretierte Sprache am Server und am PC ausgeführt werden. [36]

Zuweisung und Initialisierung

Variablen und Funktionen können einem Typ zugeordnet werden, auch var und dynamic ist möglich. Der Variable *dynamic d = 10* kann in weiterer Folge ein ganz anderer Wert, wie *d = „Ten“* zugewiesen werden. Bei *var v = 10* kann der initialisierte Typ nicht mehr verändert werden. Diese Typinferenz wird auch in der Sprache C# verwendet. Wird eine Variable zusätzlich als final deklariert, kann ihr nur einmal ein Wert zugewiesen werden. [36]

Zahlen-Typen in Dart

In Dart wird für ganze Zahlen *int* und für Gleitkommazahlen *double* verwendet. Typen aus anderen Sprachen wie *short* und *float* werden nicht benötigt, da Dart eine optimierte Speicherverwaltung besitzt, die je nach Größe der Zahl den notwendigen Speicherplatz reserviert. Alle Variablen bekommen bei der Initalisierung den Wert *null*. [36]

Zeichenketten

Für Zeichenketten gibt es den Typ *String*, der auch aus anderen Sprachen bekannt ist. Strings können mit einfachen oder auch doppelten Hochkommapaaren geschrieben werden, dies ist dem Entwickler und der Entwicklerin frei überlassen. Innerhalb der Zeichenkette kann mit einem Dollarzeichen ($) auf Variablenwerte im Programm zugegriffen werden. [36]

Objektorientierung

Die objektorientierte Programmierung ist sehr ähnlich zu Java und C#. Eine Klasse wird mittels *class* beschrieben. Das Objekt der Klasse kann danach einfach mit dem Namen der Klasse instanziert werden. Bei der Initialisierung des Objekts können Werte direkt an den Konstruktor übergeben werden. Auf öffentliche Methoden kann über die Objektvariable zugegriffen werden. [36]

4.4 Flutter

Grundlagen

Mit Flutter lassen sich über eine Code-Plattform Apps für iOS und Android, aber auch Webentwicklungen erstellen. Die App selbst sieht sehr modern aus, da *Google Material Design* verwendet wird. Diese Design-Vorlage ist jedoch nicht verpflichtend zu verwenden, es können auch für iOS die nativen Symbole im *Cupertino Style* verwendet werden. [36]

Layouts

Das Layout wird in Dart nicht so wie in Android mit XML beschrieben, sondern befindet sich direkt in der Dart-Datenstruktur. Ein grafischer *WYSIWYG-Editor* (What you see is what your get) ist nicht gegeben, jedoch auch nicht notwendig. Die Bedienung von *Drag-And-Drop* Layout-Editoren wird mit der zunehmenden Anzahl an Elementen immer komplexer und unübersichtlicher, wodurch Google bei Flutter darauf verzichtete. [36]

Alles ist ein Widget

In Flutter ist jedes Element ein Widget. Widget ist ein zusammengesetztes Wort aus „*Window*“ und „*Gadget*“ und schließt jedes sichtbare als auch unsichtbare Element der App mit ein. Die Widgets können alle ineinander verschachtelt werden, wodurch ein *Widget-Tree* entsteht, der die gesamte App abbildet. Es gibt schon sehr viele vorgefertigte Widgets,

jedoch kann der Programmierer und die Programmiererin selbst auch Widgets erstellen, die in weiterer Folge an beliebigen Stellen verwendet werden können. Dadurch wird redundanter Code verhindert und die Software in einzelne kleine, in sich geschlossene, Komponenten zerlegt. [36]

5 Forschungsdesign

Im folgenden Kapitel wird zu Beginn die Forschungsfrage definiert, anschließend wird der didaktische Aufbau und die Durchführung des Tutoriums erläutert. In weiterer Folge werden die Forschungsmethode und die Art der Datenaufnahme, die für die Beantwortung der Forschungsfrage notwendig sind, beschrieben.

5.1 Forschungsfrage

„Bietet ein zusätzliches Tutorium zum Cross Platform Framework Flutter in der Lehrveranstaltung Mensch-Computer-Interaktion einen höheren Mehrwert für die Kompetenzentwicklung der Studierenden als die bestehenden Tutorien zu React Native und Android?“

5.2 Didaktisches Design des Tutoriums

Um die genannte Forschungsfrage zu beantworten, wurde ein neues Tutorium entwickelt und erprobt. Im folgenden Abschnitt wird der didaktische Aufbau näher erläutert. Das Design des Tutoriums wurde anhand der im Kapitel „Theoretischer Rahmen zur Didaktik“ vorliegenden Fachliteratur entwickelt und mit weiterführender Literatur aus der allgemeinen Didaktik sowie der Fachdidaktik der Informatik untermauert. Neben der Strukturierung wird auf den Ablauf der einzelnen Einheiten eingegangen.

5.2.1 Der didaktische Aufbau

Das Tutorium für das Framework Flutter wird in insgesamt fünf Einheiten unterteilt, wobei jede Einheit 90 Minuten dauert. Die einzelnen Einheiten werden aufgrund der Distanzlehre über die Software Zoom abgehalten. Dabei wird im Aufbau auf wechselnde Unterrichtsmethoden geachtet. Dies soll Monotonie verhindern, und die Aufmerksamkeit der Lernenden erhöhen. Keller erklärt im ARCS-Modell, zu finden in Kapitel 2.2.4, dass auf eine entstehende Monotonie besonders geachtet werden muss, da bei der Online-Lehre das Halten der Aufmerksamkeit deutlich erschwert ist. In der folgenden Abbildung werden die eingesetzten Medien und Unterrichtsmethoden dargestellt, die auch den Studierenden in dieser Form mitgeteilt werden.

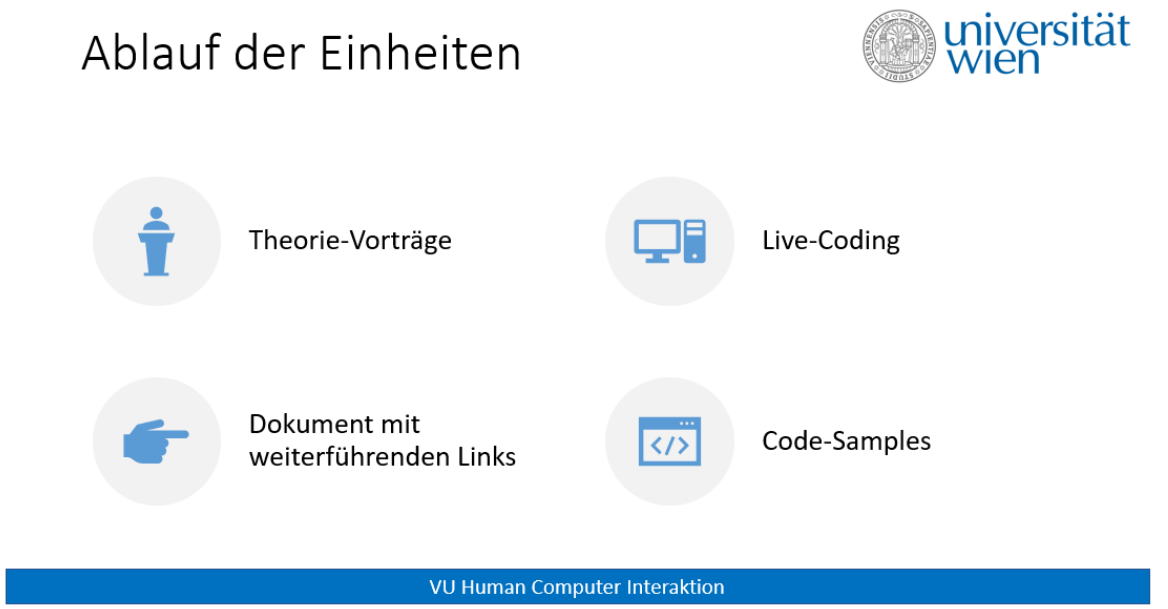

Abbildung 7: didaktischer Aufbau der Tutoriums-Einheiten
Quelle: eigene Darstellung

1. **Theorie-Vorträge**

   Mit Frontalunterricht wird den Studierenden theoretisches Wissen über das Framework Flutter, die Sprache Dart und verschiedene Konzepte der objektorientierten Programmierung vermittelt. Modrow und Strecker schreiben, dass ein Frontalunterricht zu Beginn eines Lernprozesses in vielen Fällen sinnvoll sein kann, um das nötige Wissen zu vermitteln. [37]

   Die Theorie-Vorträge zu den Wissensbereichen werden in einzelnen Themenblöcken vorgetragen. Zwischen den Theorie-Vorträgen wird ein Live-Coding eingebaut, damit die Aufmerksamkeit nicht sinkt. Die Vermittlung des Wissens darf maximal 10 bis 15 Minuten dauern, denn bereits nach fünf Minuten beginnt die Aufmerksamkeit zu sinken. [38]

2. **Live-Coding**

   Im Anschluss an die theoretischen Grundlagen des jeweiligen Themenbereichs wird ein Praxisbeispiel zur Veranschaulichung gezeigt. Der Programmcode wird je nach Komplexität und Aufwand entweder direkt in Echtzeit geschrieben, oder bereits im Vorfeld erstellt und mit den Teilnehmerinnen und Teilnehmer besprochen. Dabei werden dann kleine Änderungen im Code vorgenommen, und deren Auswirkungen auf das Programm verdeutlicht. Der Code wird in Visual Studio Code geschrieben, parallel läuft über einen Emulator die aktuelle App, die die Änderungen durch das Hot-Reloading sofort visualisiert.

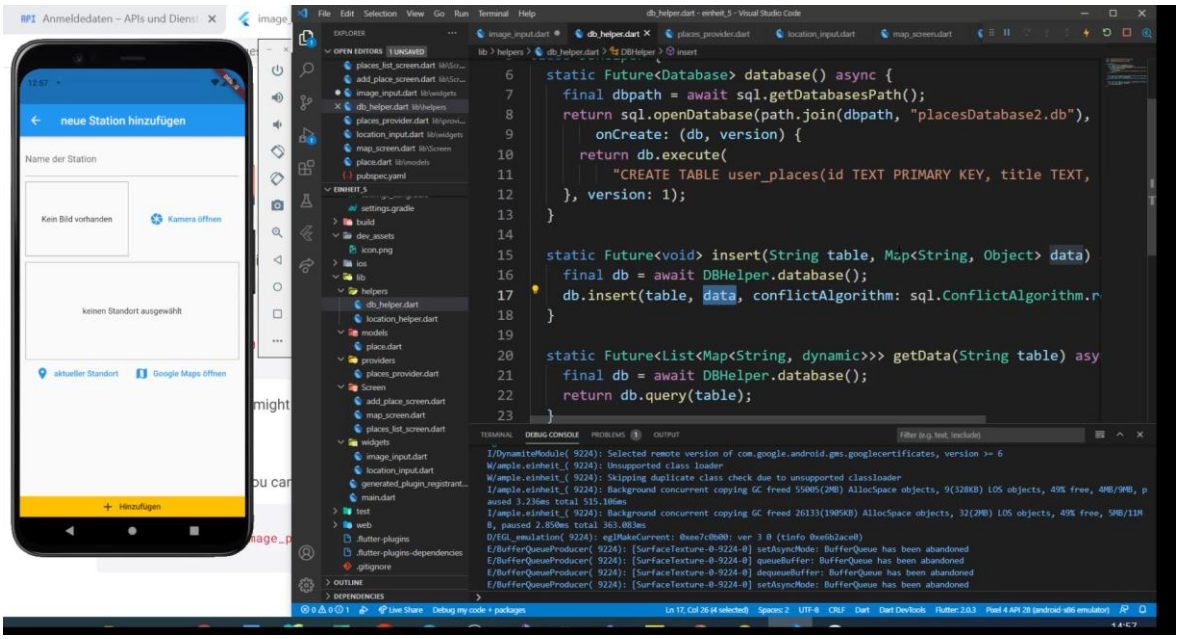

Abbildung 8: Screenshot einer Live-Programmierung
Quelle: eigene Darstellung

3. **Dokument mit weiterführenden Links**

   Nach dem Tutorium wird den Lehrveranstaltungsteilnehmerinnen und Lehrveranstaltungsteilnehmern ein Dokument auf Moodle zur Verfügung gestellt, das weiterführende Links zu den gezeigten Themen enthält. Unter anderem werden Code-Beispiele zu ähnlichen Projekten und Links zu einem noch detaillierten Einblick in die theoretischen Inhalte eingefügt.

4. **Code-Samples**

   Zusätzlich zum Tutoriumsvideo und dem Dokument mit den weiteren Lern-Angeboten wird der Quellcode der in der Einheit präsentierten App hochgeladen.

Die strukturelle Gestaltung ist nach Zylka bei der Lernumgebung ein wichtiger Aspekt. Durch das Bereitstellen des Tutoriumvideos, der Powerpoint-Folien, einem Dokument mit Links zu einer genaueren Einarbeitung in die Themen und den Code-Beispielen wird der Aspekt der strukturellen Gestaltung eingehalten. (Kapitel 2.2.3.2)

5.3 Teilnahme und Durchführung des Tutoriums

Zu Beginn der Lehrveranstaltung meldeten sich 162 Studierende für die Lehrveranstaltung „Mensch-Computer-Interaktion“ an. Die Lehrveranstaltung begann am 4. März 2021 und endete am 24. Juni 2021. In diesem Zeitraum wurden insgesamt 17 Termine abgehalten.

Aufgrund der großen Anzahl an Teilnehmerinnen und Teilnehmer gab es mehrere Übungsgruppen, bei denen zu unterschiedlichen Zeiten die Übungseinheit stattfand. Dies führte zu Überschneidungen des Flutter-Tutoriums und manchen Gruppen, wodurch die Uhrzeit flexibel angepasst wurde.

Das Flutter-Tutorium fand dabei an folgenden Tagen statt:

1. 1. Einheit: 18.03.2021 von 13:15-14:45
2. 2. Einheit: 25.03.2021 von 11:00-12:30
3. 3. Einheit: 15.04.2021 von 20:00-21:30
4. 4. Einheit: 22.04.2021 von 11:00-12:30
5. 5. Einheit: 29.04.2021 von 11:00-12:30

Dabei wurde im Moodle-Forum am Vortag eine Nachricht mit dem Beginn der jeweiligen Tutoriums-Einheit und dem Themenbereich geschrieben.

Die aktiv in der Zoom-Konferenz teilgenommenen Studierenden konnte nicht exakt erhoben werden, da sich immer wieder neue Personen ein- und ausloggten. Es nahmen jedoch zu jedem Zeitpunkt mindestens 10 Personen an der Konferenz teil.

Bei den einzelnen Terminen kam es zu keinen Verbindungsproblemen, womit das Tutorium so wie geplant abgehalten werden konnte.

5.3.1 Ablauf der einzelnen Einheiten des Tutoriums

Es wurden insgesamt die geplanten fünf Einheiten abgehalten, die sich in der Komplexität steigerten und die Studierenden auf die notwendigen Abgaben zeitgerecht vorbereiteten.

Die primären fachlichen Ziele der Lehrveranstaltung liegen, wie bereits im Einführungskapitel erläutert, bei der Umsetzung der Usability Richtlinien, der Erstellung eines benutzerfreundlichen Interfaces und der Beachtung der kognitiven Limitationen, die bei der Verwendung einer neuen Software auftreten. Daher wurde den Teilnehmerinnen und Teilnehmern mit der Live-Programmierung und den Code-Beispielen das Erlernen erleichtert. Ebenso sollte damit die Erfolgszuversicht gesteigert werden, die zu einer höheren Motivation in der Bearbeitung der Aufgaben führt. (Kapitel 2.2)

Die Einbindung der im Stream anwesenden Personen wurde ebenso in allen Einheiten berücksichtigt. Über ein Chatfenster konnten Fragen gestellt werden, die entweder direkt oder in der folgenden Einheit zu Beginn beantwortet wurden. Dies führte auch zu einigen Diskussionen, wodurch sich die Anwesenden aktiv eingebunden fühlten.

Nach jeder Tutoriumseinheit gab es einen Überblick über die Themen der folgenden Einheit und eine Zusammenfassung des bisher Gelernten. Für eine zusätzliche Aktivierung wurde in der nächsten Einheit wieder auf das vorherige Wissen verwiesen und hierbei direkt angeknüpft. Dieses Anknüpfen an Vorwissen ermöglichte ein leichteres Lernen von neuen Inhalten. [39]

1. Einheit

In der ersten Einheit wurde der Ablauf des gesamten Tutoriums vorgestellt. Dabei wurde den Studierenden mitgeteilt, wie die Einheiten generell aufgebaut sind, an welchen Tagen die einzelnen Termine stattfinden, und welche Themenblöcke zu diesen Einheiten behandelt werden. Mit diesem Überblick wurde der Aspekt der menschlichen Gestaltung von Zylka für eine gute Lernumgebung beachtet. Eine klare Planung bietet den Teilnehmerinnen und Teilnehmern Sicherheit und schafft eine angenehme Atmosphäre. (Kapitel 2.2.3.4)

Nach dem organisatorischen Teil wurde auf die historische Entwicklung von cross-platform-Frameworks und die Vorteile von Flutter zu anderen Frameworks eingegangen. Den Hauptteil bildete die Installation und Konfiguration der einzelnen Programme. Hierbei wurde *Git*, die *Flutter SDK*, *Android Studio* und *Visual Studio Code* live in der Konferenz installiert und die Einstellungen am Betriebssystem, wie zum Beispiel dem Hinzufügen der Systemumgebungsvariable der *Flutter SDK* in Windows, vorgenommen.

2. Einheit

Die zweite Einheit begann mit einem Einblick in die Programmiersprache Dart. Nach dem Theorievortrag wurden die einzelnen Konzepte der Typisierung, Verzweigungen, Schleifen und die Objektorientierung mit kleinen Beispielen auf der Webseite www.dartpad.com vorgezeigt. Als Grundlage für die Beispiele wurde das Buch „Moderne App-Entwicklung mit Dart und Flutter: eine umfassende Einführung.“ von Dieter Meiller [36] sowie die offizielle Dokumentation der Flutter Foundation [40] verwendet.

Auf der Webseite www.dartpad.com konnte der Code direkt ausgeführt werden und das Resultat des ausgeführten Codes war auf der linken Fensterhälfte ersichtlich. Das im vorherigen Theorievortrag vermittelte Wissen wurde hierbei praktisch umgesetzt und das Resultat begutachtet. An die Teilnehmerinnen und Teilnehmer wurden dabei Fragen gestellt, wie nun die Verzweigung im praktischen Beispiel auszusehen hat. Dabei wurde die Aktivierungsebene von Reinmann genutzt, die bereits in Kapitel 2.4.2 beschrieben wurde.

Widgets, welche das Kernkonzept von Flutter darstellen, wurden in einem Frontalunterricht mit Beispielen und Modellen erklärt. Darauf folgte eine kurze Pause von 5 Minuten, damit die Studierenden für den kommenden Praxisteil wieder ihre volle Aufmerksamkeit auf die Umsetzung der Widgets richten konnten. Anhand einer Quiz-App wurden nach der Pause die wichtigsten Widgets live programmiert und deren Funktion erklärt.

3. Einheit

Die dritte Einheit hatte im didaktischen Aufbau zu den vorherigen Einheiten einen anderen Zugang. Da die Themen „Navigation und Seitenwechsel in der App“ und „Strukturierung und Designen von Widgets“ sehr codelastig waren, wurde im Vorfeld eine größere App programmiert, die alle zu präsentierenden Inhalte darstellte. In der Einheit wechselte die Präsentationsform zwischen einem Theorievortrag und dem Zeigen des dazugehörigen Codes in der Beispiel-App.

Für die Strukturierung der App wurden Listen und Tabellen, sowie die verschiedenen Möglichkeiten zum Bestimmen des Abstandes zueinander näher erläutert. Um von einer Seite zur nächsten zu gelangen, wurde das Grundkonzept des Routings erklärt und die beiden vorgefertigten Widgets „Drawer“ und „BottomNavigationBar“ im Detail besprochen.

Dabei wurde nicht nur auf die Anwendung Wert gelegt, sondern auch grundlegende Richtlinien der Mensch-Computer-Interaktion angewandt. Bei der Strukturierung der Widgets wurde darauf geachtet, dass immer wiederkehrende Buttons mit derselben Logik an der gleichen Stelle platziert werden sollten. Nielsen definierte diese Strukturierung als vierte Richtlinie seiner zehn Usability-Heuristiken. (Kapitel 3.1.3)

Mit dem „Drawer“ und der „BottomNavigationBar“ konnte eine weitere Richtlinie einer guten Bedienbarkeit als praktisches Beispiel vermittelt werden. Durch diese Widgets wurden mehrere „Seiten“ in der App erstellt, auf welche die gesamten Informationen und Interaktionsmöglichkeiten aufgeteilt wurden. Durch dieses Ausblenden aller Informationen und Auswahlmöglichkeiten, die der Benutzer oder die Benutzerin nicht benötigen, sinkt die mentale Belastung des Users und es kommt zu weniger Fehlern in der Bedienung, als auch einer Verbesserung der User Experience. (Kapitel 3.1.4)

4. Einheit

Für die 4. Einheit wurde eine App vorgefertigt, die Daten von einer API empfing und zurücksandte. Als API wurde eine Firebase Datenbank erstellt, die Einrichtung dieser wurde im Tutorium ebenfalls behandelt. Der zweite Themenkomplex war das State Management und die verschiedenen Ansätze dazu in Flutter. Hierbei wurden vier mögliche State-Management Systeme vorgestellt, die zwei bekanntesten wurden anhand einer im Vorfeld erstellten Beispiel-App in der Umsetzung näher erklärt.

Die Vermittlung der Firebase-API hatte mehrere Gründe. Einerseits können mit dieser API sehr rasch Server-Mockups nachgestellt werden, wodurch das Prototyping vereinfacht wird. Das Prototyping ist wie in der Einleitung beschrieben, ein Ziel der Lehrveranstaltung. Die Firebase-API bietet eine enorme Palette unterschiedlicher Funktionen des Cloud

Computing. Interessierte Teilnehmerinnen und Teilnehmer wird mit diesem Einblick und der beiliegenden Link-Sammlung ein entdeckendes Lernen ermöglicht, und dadurch die Lernumgebung verbessert. (Kapitel 2.2.3)

Die verschiedenen Möglichkeiten des State Managements sollten bei den Studierenden die intrinsische Motivation steigern. Die Ansätze hatten einen unterschiedlichen Schwierigkeitsgrad, die Möglichkeiten der Zustandsspeicherung stiegen jedoch auch mit der Komplexität. Dadurch sollte der Drang nach neuem Wissen und die Neugier angeregt werden, als auch die Erfolgszuversicht, die eine Dimension des ARCS-Modells von Keller darstellt. (Kapitel 2.2.4)

5. Einheit

Die letzte Einheit behandelte verschiedene Zugriffsmöglichkeiten auf Funktionen des Smartphones und dessen vorgefertigten Plugins mit dem Flutter-Framework. Dazu gehörte die Kamera, das interne Dateisystem, das Speichern und Laden einer SQLite-Datenbank, der Zugriff auf den Standort und die Anzeige von Google Maps. Eine im Vorfeld erstellte Demo-App wurde hier wieder für den praktischen Teil verwendet, um alle Funktionen in der gegebenen Zeit zu vermitteln. Der verbliebene Teil der Zeit wurde für das Generieren der App als APK, die Konfiguration des Starticons und anderen Einstellungsmöglichkeiten in der APK-Erstellung verwendet.

Die Studierenden erhielten in dieser Einheit ein großes Repertoire an Funktionen, die sie für die Bearbeitung der notwendigen Abgaben benötigen. Das Ziel hierbei war wie bereits in der allgemeinen Vorgehensweise darin, den Teilnehmerinnen und Teilnehmern schnellstmöglich die Anwendung beizubringen, damit sie ihre vorhandene Zeit vermehrt in die Umsetzung der Design Prinzipien stecken können.

5.3.2 Aufnahme des Tutoriums

Die Aufnahme des Bildschirms wurde mit dem Programm OBS Studio, eine professionelle Software zur Aufzeichnung von Streams, durchgeführt. Über das Meeting-Tool Zoom konnte die Einheit nicht aufgezeichnet werden, da dafür ein Premium-Account nötig gewesen wäre. Die Zoom-Einheiten wurden über Moodle erstellt, damit alle ordentlichen Studierenden teilnehmen konnten. Im Hintergrund organisierte Moodle auch, welche Person bei der Einheit die Rolle des Präsentators bzw. der Präsentatorin besitzen darf und weitere Rechte wie die Erlaubnis der Aufnahmefunktion. Die Aufnahmefunktion war nur Personen zugeteilt, welche im universitätsinternen System eine LektorInnen-Rolle aufwiesen. Für die Leitung des Flutter-Tutoriums wurde diese Rolle nicht automatisch zugeteilt, eine Zuweisung über die IT-Administration der Universität hätte nicht zeitgerecht erfolgen können.

Vor der ersten Tutoriums-Einheit wurde die Aufnahmefunktion überprüft. Eine Flutter-App wurde mit Visual Studio Code erstellt und auf dem Emulator simuliert. Das aufgenommene Video wurde auf ihre Qualität vom Tutoriumleiter angesehen und wies keine Probleme auf. In den ersten drei Einheiten des Tutoriums wurde das Video auch kurz auf ihre Vollständigkeit und Sprachqualität geprüft, es wurden keine Mängel erkannt.

Bei der vierten Einheit gab es ein Problem mit der Aufnahme des Tutoriums. Die Einheit wurde im Anschluss nochmals offline aufgenommen und hochgeladen. Einige Tage vor der letzten Tutoriums-Einheit schrieb ein Kursteilnehmer im Moodle-Forum, dass die ersten drei Videoaufnahmen nach den ersten Minuten zu stocken beginnen und nur die vierte Einheit einwandfrei läuft. Einige andere Studierende antworteten darauf, dass dies bei ihnen ebenso der Fall sei. Die letzte Einheit wurde wieder über Zoom gestreamt, hierbei kam es wieder zu Aufnahmeproblemen, die sich im Video in einer sehr niedrigen Bildrate zeigte.

5.4 Forschungsmethoden

Um die Qualität und Wirkweise des Tutoriums zu untersuchen, und damit die Forschungsfrage zu beantworten, wurde ein Fragebogen entwickelt, der die Studierenden über das von ihnen besuchte Tutorium befragte. Der Fragebogen teilte sich in folgende fünf Abschnitte auf:

1. Erhebung des verwendeten Frameworks und des besuchten Tutoriums

Die erste Fragestellung lautete: „Ich habe für die Einzelaufgabe 2 folgendes Framework verwendet“.

Dabei gab es folgende Antwortmöglichkeiten:

1. Android
2. React Native
3. Flutter
4. Aufgabe für Nicht-InformatikerInnen

Die zweite Fragestellung lautete: „Ich habe in der LV folgendes Tutorium besucht“.

Dabei gab es folgende Antwortmöglichkeiten:

- Android
- React Native
- Flutter

Mit diesen beiden Fragestellungen konnte einerseits die Anzahl der Teilnehmerinnen und Teilnehmer in den einzelnen Tutorien erhoben werden, als auch die für die Einzelaufgabe 2 verwendeten Frameworks.

2. Erhebung der Qualität des Tutoriums

Der zweite Abschnitt erhob die von den Studierenden subjektiv empfundene Qualität des Tutoriums hinsichtlich mehrerer Kriterien. Die fünf geschlossenen Fragestellungen wurden mit einer 5-stufigen Likert-Skala gestaltet.

Der Skalenbereich lautete:

1. Stimme überhaupt nicht zu
2. Stimme eher nicht zu
3. Stimme weder zu noch lehne ich ab
4. Stimme zu
5. Stimme voll und ganz zu

Folgende Fragestellungen wurden den Studierenden in diesem Abschnitt gestellt:

- Die Inhalte des Tutoriums wurden verständlich vermittelt
- Die Inhalte wurden strukturiert aufgebaut
- Die bereitgestellten Lernmaterialien waren hilfreich
- Die im Tutorium eingesetzten Medien waren sinnvoll
- Die Fragen der Studierenden wurden im Tutorium zufriedenstellend beantwortet

Mithilfe dieser Fragestellungen sollte die Forschungsfrage bezüglich der Qualität des Tutoriums beantwortet werden. Als Qualitätskriterien werden hierbei folgende Merkmale guten Unterrichts von Hilbert Meyer aus dem Kapitel 2.1 herangezogen:

- Klare Strukturierung des Unterrichts
- Inhaltliche Klarheit
- Sinnstiftendes Kommunizieren
- Methodenvielfalt
- Individuelles Fördern

3. Erhebung der Kompetenzentwicklung hinsichtlich der Ziele für die Einzelaufgabe 2

Im dritten Abschnitt wurden Daten zur Beantwortung der zweiten Frage hinsichtlich des Kompetenzerwerbs der Studierenden erhoben. Die beiden geschlossenen Fragestellungen wurden mit einer 5-stufigen Likert-Skala gestaltet. Der Skalenbereich lautete:

1. Stimme überhaupt nicht zu
2. Stimme eher nicht zu
3. Stimme weder zu noch lehne ich ab
4. Stimme zu
5. Stimme voll und ganz zu

Die gestellten Fragestellungen lauteten:

- **Das Tutorium war für die Implementierung der Funktionen hilfreich**
  Dabei wurde ermittelt, ob mit dem Tutorium die technische Umsetzung für die Einzelaufgabe 2 erleichtert wurde. Das Verständnis für die Syntax und Semantik der Programmiersprache Dart sowie die korrekte Anwendung des Frameworks Dart waren einige Ziele des Tutoriums.
- **Das Tutorium war für das Erstellen des Interfaces hilfreich**
  Neben der technischen Umsetzung musste man in der Einzelaufgabe 2 auch die Richtlinien der Usability berücksichtigen. Im Flutter-Tutorium wurde dies besprochen. Mit der Fragestellung sollte ermittelt werden, ob das Tutorium in diesem Bereich ebenfalls hilfreich war.

4. Erhebung der subjektiv empfundenen allgemeinen Kompetenzentwicklung

Neben der Kompetenzentwicklung für die Einzelaufgabe 2 sollte auch das subjektive Empfinden der allgemeinen Kompetenzverbesserung zu den Themen der Lehrveranstaltung „Mensch-Computer-Interaktion“ abgefragt werden, um die Forschungsfrage zu der Kompetenzentwicklung vollständig beantworten zu können.

Dazu wurden zwei geschlossenen Fragestellungen mit der bereits bekannten 5-stufigen Likert-Skala gestaltet. Der Skalenbereich lautete wieder:

1. Stimme überhaupt nicht zu
2. Stimme eher nicht zu
3. Stimme weder zu noch lehne ich ab
4. Stimme zu
5. Stimme voll und ganz zu

5. Offene Fragestellungen für positives und negatives Feedback

Im letzten Abschnitt des Fragebogens wurden zwei offene Fragestellungen eingefügt, die mit einer unbegrenzten Anzahl an Zeichen beantwortet werden konnten. Die Fragestellungen lauteten:

- Besonders gut gefallen hat mir im Tutorium
- Folgendes könnte im Tutorium verbessert werden

Die damit erhobenen Daten dienten der qualitativen Analyse des Tutoriums und trugen zur Beantwortung der Forschungsfragen bei.

5.5 Erhebung der Daten

Für die Lehrveranstaltung „Mensch-Computer-Interaktion“ wurde eine Moodle-Seite eingerichtet, in der neben den Einladungslinks für die Tutoriumseinheiten auch ein Forum für den Austausch der Studierenden untereinander, als auch mit dem Team der Lehrveranstaltungsleitung vorhanden war. Diese Moodle-Plattform ermöglichte auch eine Umfrage, mit der alle Teilnehmerinnen und Teilnehmer erreicht werden konnten. Das von Moodle bereitgestellte Plugin „Reaktionsblatt“ wurde für die Umfrage verwendet und damit der Fragebogen, wie im vorherigen Abschnitt „Forschungsmethoden“ beschrieben, erstellt.

Den Studierenden wurde bereits im Vorfeld mitgeteilt, dass sie für die Teilnahme an der Umfrage Mitarbeitspunkte bekamen, und die Umfrage anonymisiert wurde. Der Fragebogen wurde freigeschaltet, nachdem die Note für die „Einzelaufgabe 2“ eingetragen wurde. 14 Tage hatten die Teilnehmerinnen und Teilnehmer für die Beantwortung des Fragebogens Zeit, danach wurde die Umfrage beendet. Insgesamt nahmen, von den zu Beginn 162 eingeschriebenen Studierenden, 92 Personen an der Umfrage teil.

6 Ergebnisse

In diesem Abschnitt werden die mit dem Fragebogen gewonnenen Daten ausgewertet und die Ergebnisse präsentiert. Zuerst werden die quantitativen Ergebnisse der Umfrage beschrieben und mit einer grafischen Gegenüberstellung der einzelnen Tutorien visualisiert. Neben den quantitativen Daten des Fragebogens wurden auch die Daten der Lehrveranstaltungsabgaben analysiert und ausgewertet. Dies ist im zweiten Unterkapitel zu finden. Die qualitativen Daten werden in Kapitel 6.3 ausgewertet und beschrieben. Dazu wurde eine qualitative Inhaltsanalyse durchgeführt und Kategorien induktiv erstellt. Alle Daten wurden aus dem Moodle-Forum via .csv-Datei exportiert und mit der Tabellenkalkulationssoftware Excel formatiert und aufbereitet.

6.1 Auswertung und Ergebnisse der quantitativen Daten

Die Abbildungen 9 bis 26 wurden mit Excel erstellt als Grafiken exportiert. Dazu wurden die Datensätze einzelnen Arbeitsblätter aufgeteilt und für die weitere Berechnung formatiert. Zur Darstellung der Daten wird ein Box-Plot verwendet. Mit diesem Diagramm kann sehr schnell ein erster Eindruck über die Daten und deren Streuung vermittelt werden. In der Abbildung 9 ist der Aufbau eines Boxplots ersichtlich. Die rechteckige Box entspricht dem Bereich, in dem 50% aller Daten liegen. Das X innerhalb des Rechtecks kennzeichnet den Mittelwert aller Daten. Die „Antennen“ an den Enden des Rechtecks stellen den Streuungsbereich der restlichen Werte dar. Datenpunkte, die sehr weit außerhalb der restlichen Werte liegen, werden als einzelne Punkte dargestellt und „Ausreißer“ genannt. [41] Eine sehr enge obenliegende Box deutet in den folgenden Abbildungen auf eine hohe Zustimmung zu, je weiter unten die Box liegt, desto niedriger ist die Zustimmung.

Für die Boxplot-Darstellung wurden die Daten mit dem Diagrammtyp „Kastengrafik“ dargestellt. Die statistischen Kenngrößen wurden über die Funktion „Datenanalyse → Populationskenngrößen“ berechnet. Im folgenden Abschnitt werden die statistischen Kenngrößen und die Boxplot-Darstellung zu jeder Forschungsfrage dargestellt, gefolgt von einer Analyse der Daten.

Die Boxplots sind in der Y-Achse folgendermaßen zu lesen:

5 → Stimme voll und ganz zu

4 → Stimme zu

3 → Stimme weder zu noch lehne ich ab

2 → Stimme eher nicht zu

1 → Stimme überhaupt nicht zu

Die Inhalte des Tutoriums wurden verständlich vermittelt

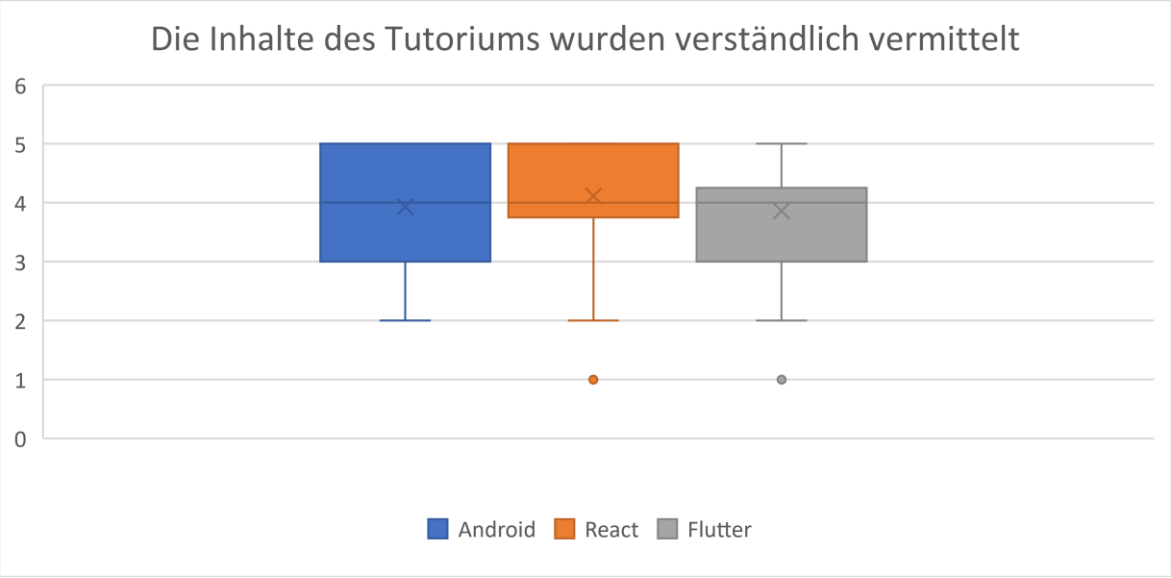

Abbildung 9: visuelle Auswertung zur Fragestellung "Die Inhalte des Tutoriums wurden verständlich vermittelt"
Quelle: eigene Darstellung

Android			*React*			*Flutter*	
Mittelwert	3,933333333		Mittelwert	4,11538462		Mittelwert	3,86
Standardfehler	0,283962886		Standardfehler	0,20995914		Standardfehler	0,13404233
Median	4		Median	4		Median	4
Modus	5		Modus	5		Modus	4
Standardabweichung	1,099783528		Standardabweichung	1,07058575		Standardabweichung	0,94782242
Minimum	2		Minimum	1		Minimum	1
Maximum	5		Maximum	5		Maximum	5
Summe	59		Summe	107		Summe	193
Anzahl	15		Anzahl	26		Anzahl	50

Abbildung 10: quantitative Daten zur Fragestellung "Die Inhalte des Tutoriums wurden verständlich vermittelt"
Quelle: eigene Darstellung

Analyse

Am verständlichsten wurde das React-Tutorium mit einem Mittelwert von ~4,12 bewertet. Android und Flutter wurden mit einem Mittelwert von ~3,93 und 3,86 bewertet, und schneiden daher schlechter ab. Bei der Berücksichtigung des Standardfehlers liegen alle Tutorien im selben Intervall. Der Minimal-Wert beträgt bei Android 2, bei den anderen beiden Tutorien 1.

Die Inhalte wurden strukturiert aufgebaut

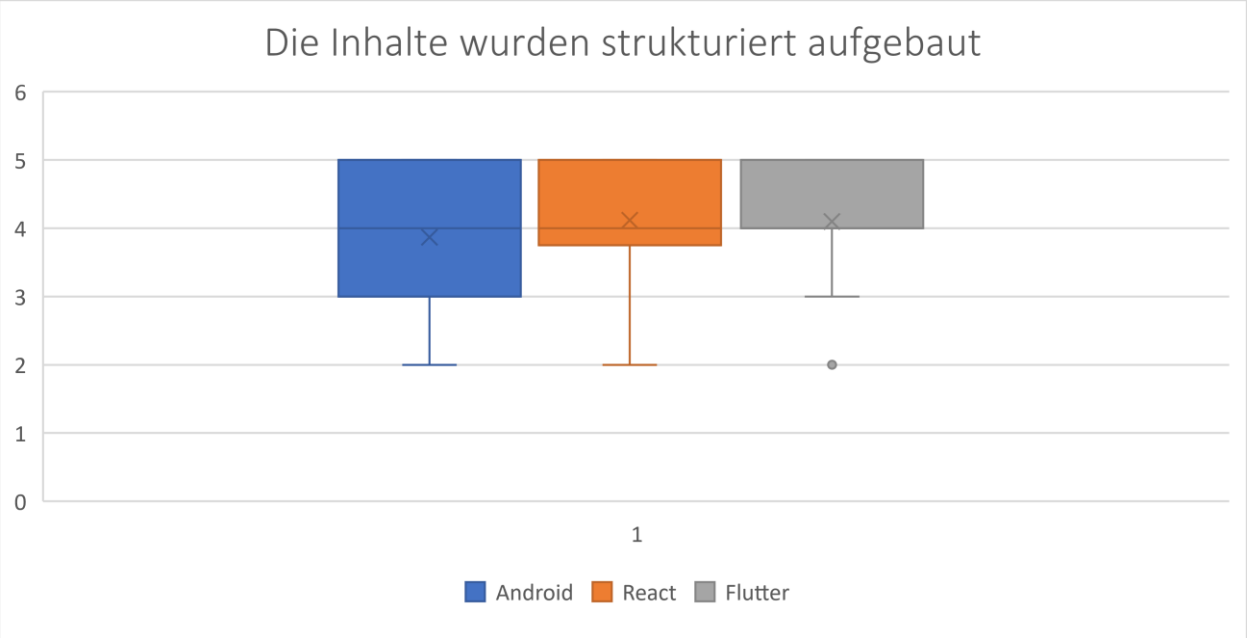

Abbildung 11:visuelle Auswertung zur Fragestellung "Die Inhalte wurden strukturiert aufgebaut "
Quelle: eigene Darstellung

Android			*React*			*Flutter*	
Mittelwert	3,866666667		Mittelwert	4,11538462		Mittelwert	4,1
Standardfehler	0,290593263		Standardfehler	0,1694055		Standardfehler	0,09583148
Median	4		Median	4		Median	4
Modus	4		Modus	4		Modus	4
Standardabweichung	1,125462868		Standardabweichung	0,86380197		Standardabweichung	0,67763093
Minimum	2		Minimum	2		Minimum	2
Maximum	5		Maximum	5		Maximum	5
Summe	58		Summe	107		Summe	205
Anzahl	15		Anzahl	26		Anzahl	50

Abbildung 12: quantitative Daten zur Fragestellung "Die Inhalte wurden strukturiert aufgebaut "
Quelle: eigene Darstellung

Analyse

Bei der Strukturierung der Inhalte schneiden das React-Tutorium und das Flutter-Tutorium mit jeweils ~4,1 Punkten ähnlich ab. Das Android-Tutorium liegt mit ~3,86 Punkten hier weiter hinten, jedoch ist bei der Berücksichtigung des Standardfehlers kein signifikanter Unterschied zu erkennen. Das Flutter-Tutorium hat eine geringere Standardabweichung als das React-Tutorium, wodurch eine geringere Streuung zum Mittelwert erkennbar ist.

Die bereitgestellten Lernmaterialien waren hilfreich

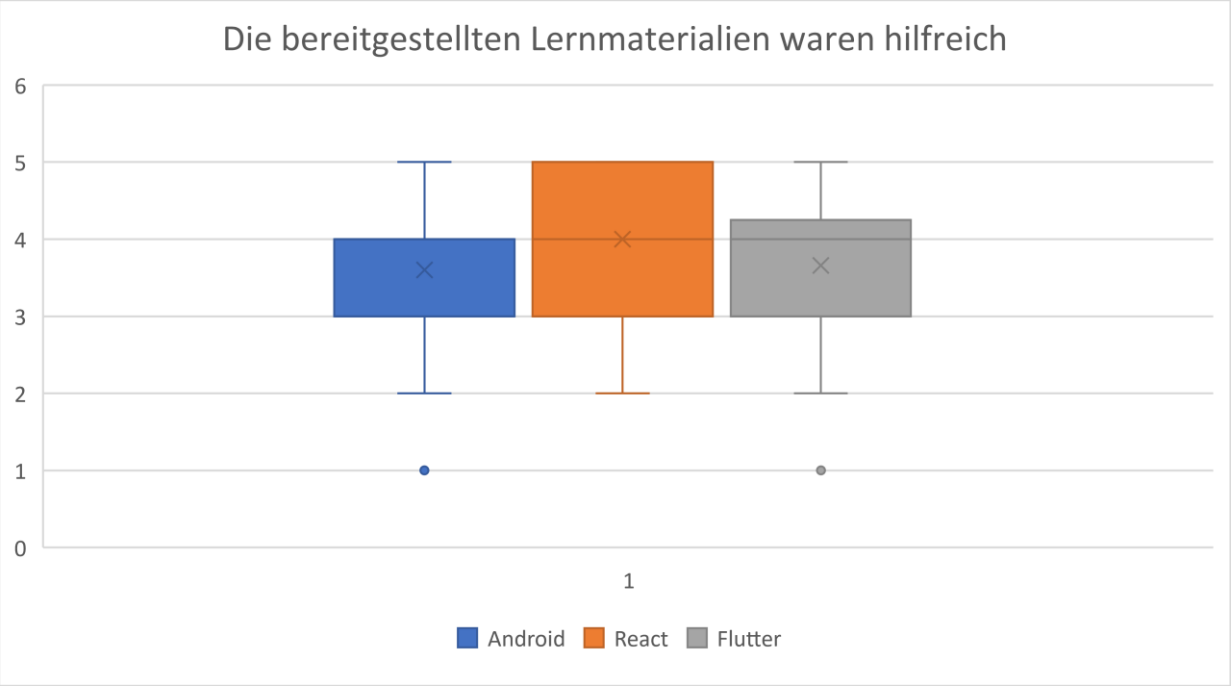

Abbildung 13: visuelle Auswertung zur Fragestellung "Die bereitgestellten Lernmaterialien waren hilfreich"
Quelle: eigene Darstellung

Android			*React*			*Flutter*	
Mittelwert	3,6		Mittelwert	4		Mittelwert	3,66
Standardfehler	0,305505046		Standardfehler	0,19215378		Standardfehler	0,15282376
Median	4		Median	4		Median	4
Modus	4		Modus	5		Modus	4
Standardabweichung	1,183215957		Standardabweichung	0,9797959		Standardabweichung	1,08062718
Minimum	1		Minimum	2		Minimum	1
Maximum	5		Maximum	5		Maximum	5
Summe	54		Summe	104		Summe	183
Anzahl	15		Anzahl	26		Anzahl	50

Abbildung 14: quantitative Daten zur Fragestellung "Die bereitgestellten Lernmaterialien waren hilfreich"
Quelle: eigene Darstellung

Analyse

Bei den bereitgestellten Materialien wurde das React-Tutorium mit einem Mittelwert von 4 am besten bewertet. In React wurde die Antwortmöglichkeit „stimme voll und ganz zu“ am häufigsten gewählt, in den anderen Tutorien war dies nur „stimme zu“.

Die im Tutorium eingesetzten Medien waren sinnvoll

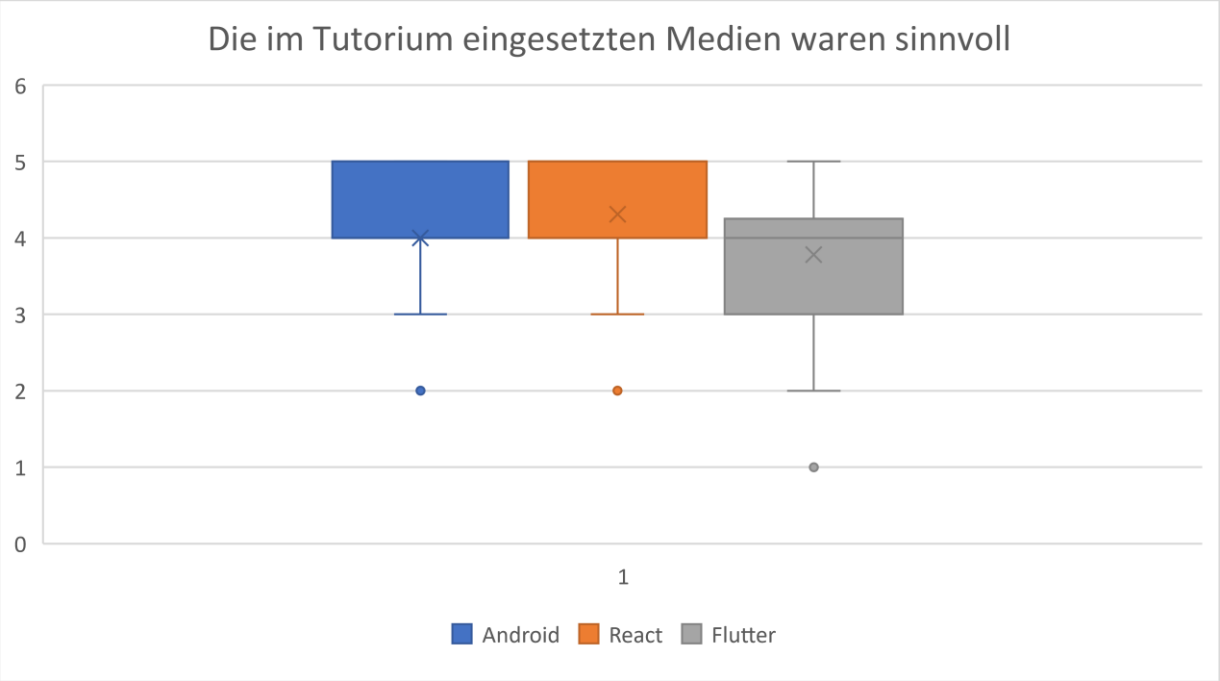

Abbildung 15: visuelle Auswertung zur Fragestellung "Die im Tutorium eingesetzten Medien waren sinnvoll"
Quelle: eigene Darstellung

Android			*React*			*Flutter*	
Mittelwert	4		Mittelwert	4,30769231		Mittelwert	3,78
Standardfehler	0,21821789		Standardfehler	0,17337581		Standardfehler	0,1464965
Median	4		Median	5		Median	4
Modus	4		Modus	5		Modus	4
Standardabweichung	0,845154255		Standardabweichung	0,88404664		Standardabweichung	1,03588669
Minimum	2		Minimum	2		Minimum	1
Maximum	5		Maximum	5		Maximum	5
Summe	60		Summe	112		Summe	189
Anzahl	15		Anzahl	26		Anzahl	50

Abbildung 16: quantitative Daten zur Fragestellung "Die im Tutorium eingesetzten Medien waren sinnvoll"
Quelle: eigene Darstellung

Analyse

Bei den eingesetzten Medien unterscheiden sich die Mittelwerte stärker als in den vorherigen Fragen. Dabei erreicht das React-Tutorium 4,3 Punkte, Android 4 Punkte, und Flutter 3,78. Bei der Berücksichtigung des Standardfehlers ist noch ein Unterschied zwischen React und Flutter zu erkennen.

Die Fragen der Studierenden wurden im Tutorium zufriedenstellend beantwortet

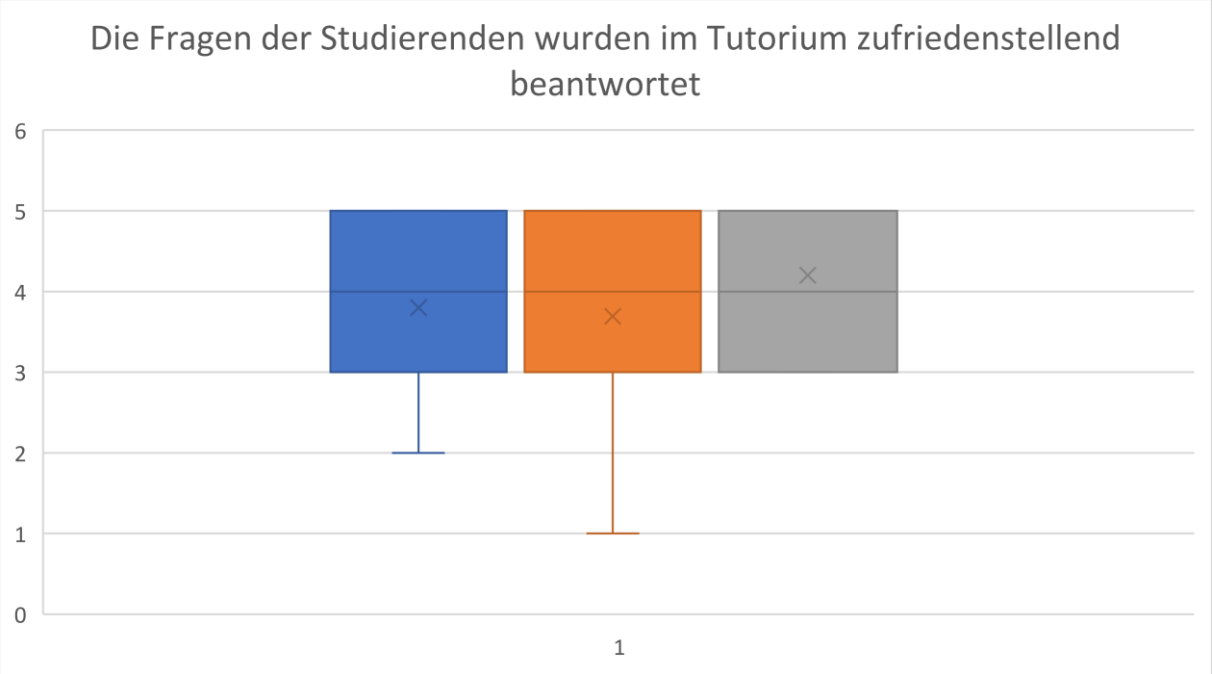

Abbildung 17: visuelle Auswertung zur Fragestellung "Die Fragen der Studierenden wurden im Tutorium zufriedenstellend beantwortet"
Quelle: eigene Darstellung

Android			*React*			*Flutter*	
Mittelwert	3,8		Mittelwert	3,69230769		Mittelwert	4,2
Standardfehler	0,242997159		Standardfehler	0,24031537		Standardfehler	0,11780302
Median	4		Median	4		Median	4
Modus	3		Modus	4		Modus	5
Standardabweichung	0,941123948		Standardabweichung	1,22537278		Standardabweichung	0,83299313
Minimum	2		Minimum	1		Minimum	3
Maximum	5		Maximum	5		Maximum	5
Summe	57		Summe	96		Summe	210
Anzahl	15		Anzahl	26		Anzahl	50

Abbildung 18: quantitative Daten zur Fragestellung "Die Fragen der Studierenden wurden im Tutorium zufriedenstellend beantwortet"
Quelle: eigene Darstellung

Analyse

Das Flutter-Tutorium erreicht hier einen Mittelwert von 4,2 und liegt damit vor den anderen beiden Tutorien. Mit Berücksichtigung des Standardfehlers ist ein Unterschied zwischen Flutter und den anderen Tutorien erkennbar. Auffallend ist der Minimum-Wert von Flutter, welcher bei 3 liegt. Ein weiterer Unterschied ist im Modus erkennbar. Android erreicht hier den Wert 3, React 4 und das Flutter-Tutorium wurde hinsichtlich der Beantwortung von Fragen am häufigsten mit „stimme voll und ganz zu“ bewertet.

Das Tutorium war für die Implementierung der Funktionen hilfreich

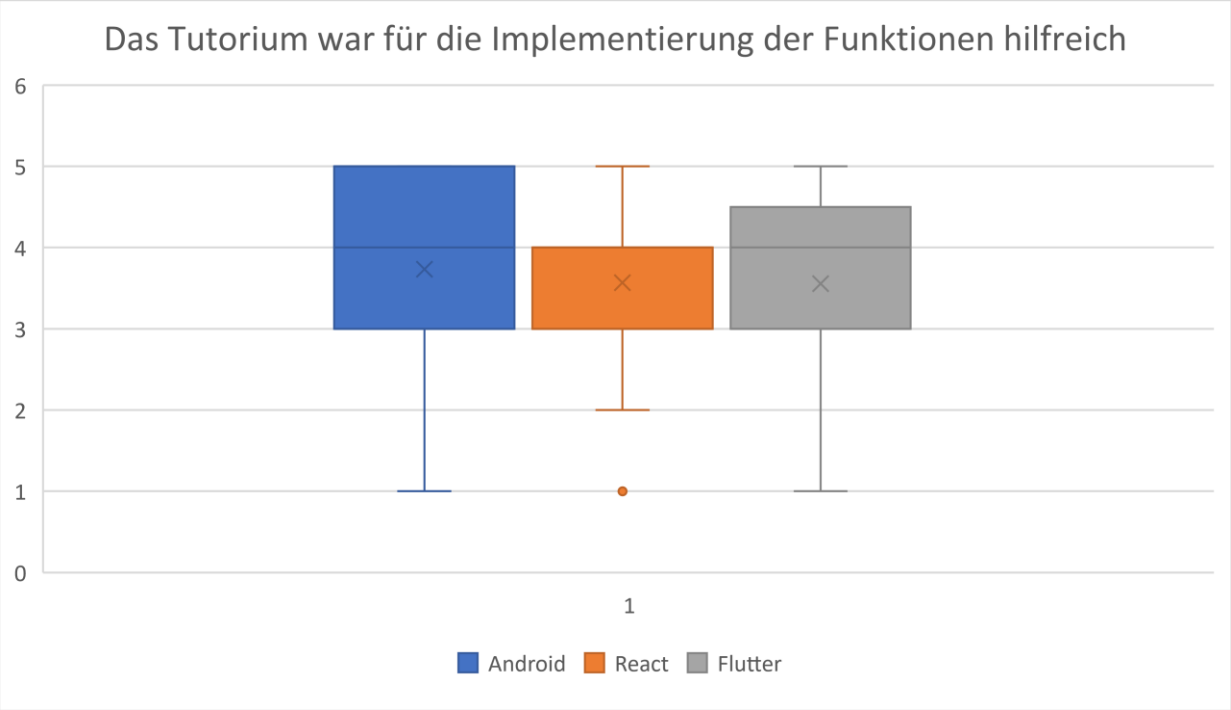

Abbildung 19: visuelle Auswertung zur Fragestellung "Das Tutorium war für die Implementierung der Funktionen hilfreich"
Quelle: eigene Darstellung

Android			*React*			*Flutter*	
Mittelwert	3,733333333		Mittelwert	3,56521739		Mittelwert	3,55555556
Standardfehler	0,358125632		Standardfehler	0,2578881		Standardfehler	0,16683493
Median	4		Median	4		Median	4
Modus	4		Modus	4		Modus	4
Standardabweichung	1,387014608		Standardabweichung	1,23678788		Standardabweichung	1,11916275
Minimum	1		Minimum	1		Minimum	1
Maximum	5		Maximum	5		Maximum	5
Summe	56		Summe	82		Summe	160
Anzahl	15		Anzahl	23		Anzahl	45

Abbildung 20: quantitative Daten zur Fragestellung "Das Tutorium war für die Implementierung der Funktionen hilfreich"
Quelle: eigene Darstellung

Analyse

Die Tutorien wurden hier bei Beachtung des Mittelwerts sehr ähnlich bewertet. Auch der Median, Modus, Minimum und Maximum-Wert sind in allen drei Tutorien identisch.

Das Tutorium war für das Erstellen des Interfaces hilfreich

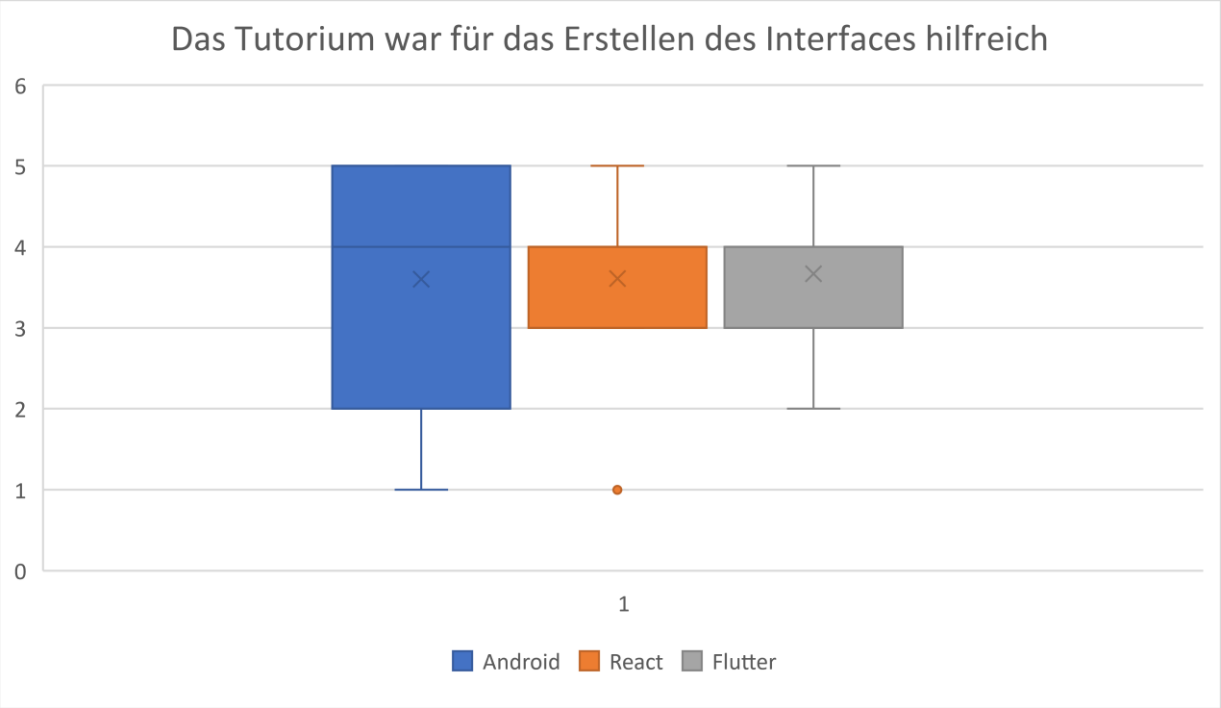

Abbildung 21: visuelle Auswertung zur Fragestellung "Das Tutorium war für das Erstellen des Interfaces hilfreich"
Quelle: eigene Darstellung

Android			*React*			*Flutter*	
Mittelwert	3,6		Mittelwert	3,60869565		Mittelwert	3,66666667
Standardfehler	0,375436254		Standardfehler	0,19620037		Standardfehler	0,14213381
Median	4		Median	4		Median	4
Modus	5		Modus	4		Modus	4
Standardabweichung	1,45405836		Standardabweichung	0,94094394		Standardabweichung	0,95346259
Minimum	1		Minimum	1		Minimum	2
Maximum	5		Maximum	5		Maximum	5
Summe	54		Summe	83		Summe	165
Anzahl	15		Anzahl	23		Anzahl	45

Abbildung 22: quantitative Daten zur Fragestellung "Das Tutorium war für das Erstellen des Interfaces hilfreich"
Quelle: eigene Darstellung

Analyse

Der Mittelwert liegt bei allen drei Tutorien bei rund 3,6, das Flutter-Tutorium wurde hier mit 3,6 ein wenig besser bewertet als die anderen beiden mit jeweils 3,6. Der Median und das Maximum sind ebenso ident. Bei Android liegt der Modus bei 5, React und Flutter wurden am häufigsten mit 4 bewertet.

Meine Programmierkenntnisse haben sich durch das Tutorium insgesamt verbessert

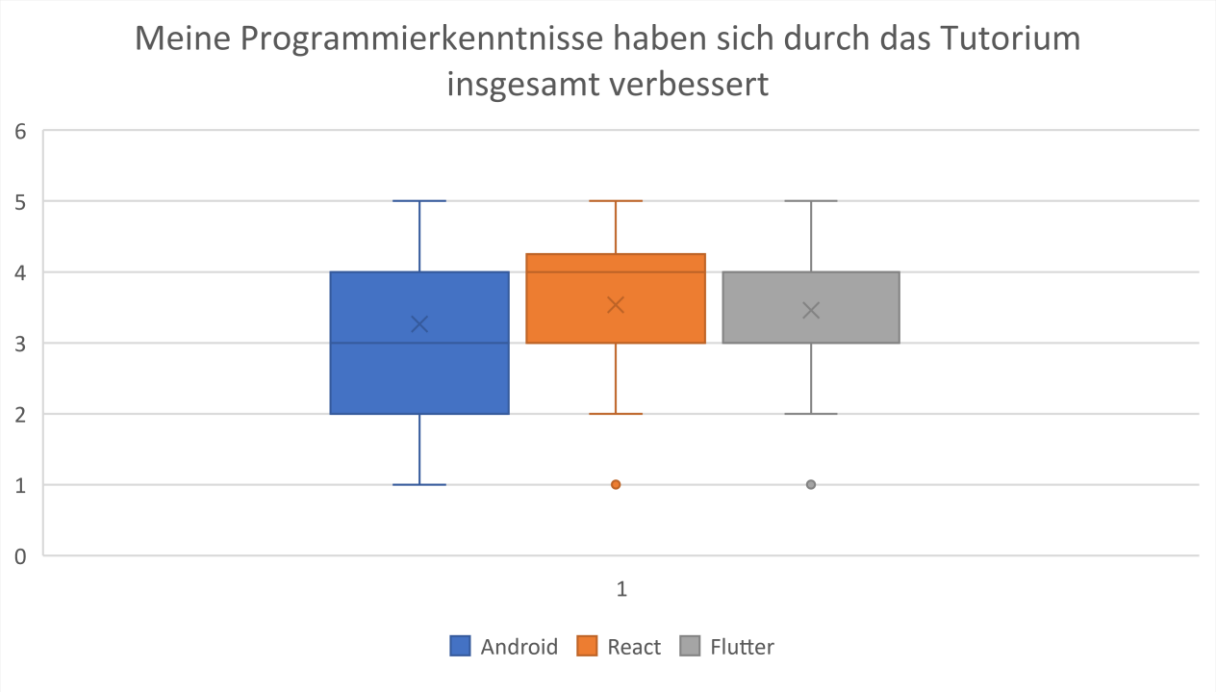

Abbildung 23: visuelle Auswertung zur Fragestellung "Meine Programmierkenntnisse haben sich durch das Tutorium insgesamt verbessert"
Quelle: eigene Darstellung

Android			*React*			*Flutter*	
Mittelwert	3,266666667		Mittelwert	3,53846154		Mittelwert	3,46
Standardfehler	0,300264434		Standardfehler	0,24949653		Standardfehler	0,16465177
Median	3		Median	4		Median	4
Modus	4		Modus	4		Modus	4
Standardabweichung	1,162919151		Standardabweichung	1,2721877		Standardabweichung	1,1642638
Minimum	1		Minimum	1		Minimum	1
Maximum	5		Maximum	5		Maximum	5
Summe	49		Summe	92		Summe	173
Anzahl	15		Anzahl	26		Anzahl	50

Abbildung 24: quantitative Daten zur Fragestellung "Meine Programmierkenntnisse haben sich durch das Tutorium insgesamt verbessert"
Quelle: eigene Darstellung

Analyse

Mit Beachtung des Standardfehlers gibt es keine Unterschiede zwischen den Tutorien hinsichtlich des Mittelwerts. Alle drei Tutorien haben den gleichen Modus, das Minimum und Maximum sind auch ident. Der Median liegt mit 3 bei Android unter dem erreichten Wert von Flutter und React.

Ich fühle mich nun für die App Entwicklung im Teamprojekt (Meilenstein 3) gut vorbereitet

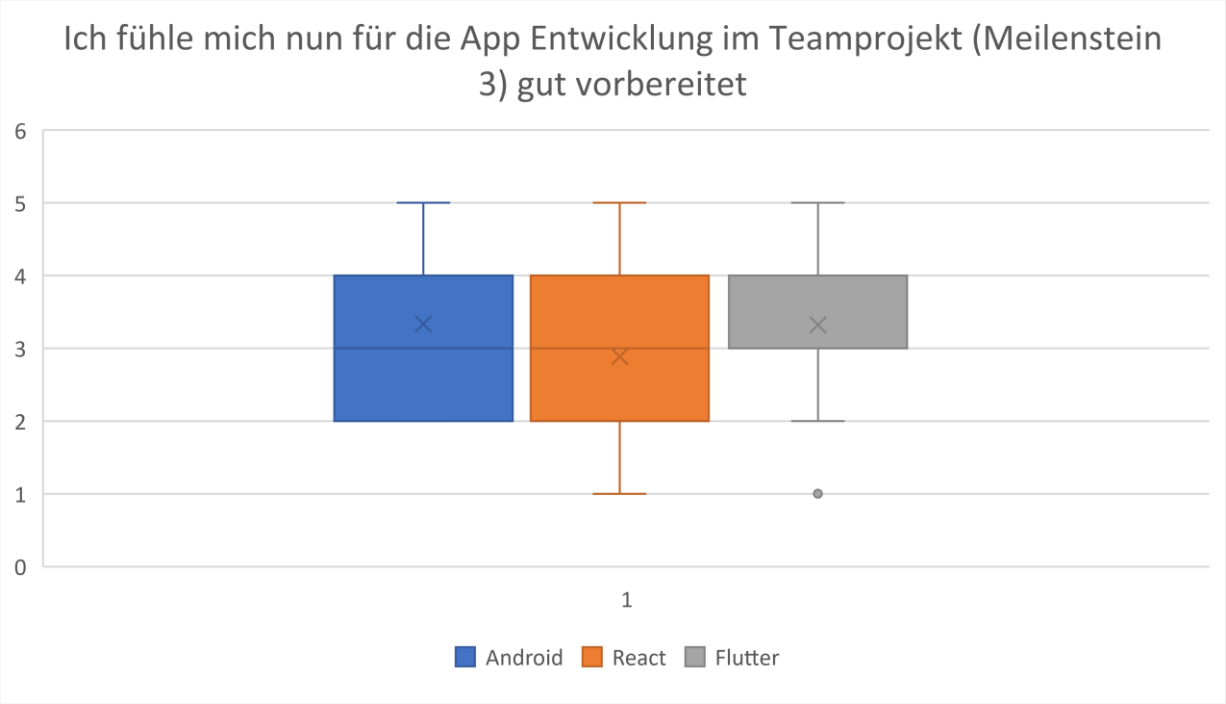

Abbildung 25: visuelle Auswertung zur Fragestellung "Ich fühle mich nun für die App Entwicklung im Teamprojekt (Meilenstein 3) gut vorbereitet
Quelle: eigene Darstellung

Android			*React*			*Flutter*	
Mittelwert	3,333333333		Mittelwert	2,88461538		Mittelwert	3,32
Standardfehler	0,270214494		Standardfehler	0,21716299		Standardfehler	0,13832821
Median	3		Median	3		Median	3
Modus	4		Modus	2		Modus	4
Standardabweichung	1,046536237		Standardabweichung	1,10731831		Standardabweichung	0,97812816
Minimum	2		Minimum	1		Minimum	1
Maximum	5		Maximum	5		Maximum	5
Summe	50		Summe	75		Summe	166
Anzahl	15		Anzahl	26		Anzahl	50

Abbildung 26: quantitative Daten zur Fragestellung "Ich fühle mich nun für die App Entwicklung im Teamprojekt (Meilenstein 3) gut vorbereitet
Quelle: eigene Darstellung

Analyse

Bei dieser Fragestellung liegt React, auch unter Berücksichtigung des Standardfehlers, hinter Flutter. Android erreicht hier einen Mittelwert von ~3,33 und liegt damit knapp vor Flutter, das einen Mittelwert von 3,32 erreichte. Der Standardfehler ist bei Android aufgrund der geringeren Anzahl an Werten fast doppelt so hoch wie bei Flutter.

6.2 Quantitative Daten der Lehrveranstaltungsabgaben

Neben den erhaltenen Daten aus dem Fragebogen bieten auch die quantitativen Daten der Lehrveranstaltung eine weitere Möglichkeit, die Forschungsfrage zu beantworten. Dazu wurden die Daten der Abgaben aus dem Moodle-Kurs als .csv-Datei runtergeladen, in Excel importiert und, wie in Abbildung 27 ersichtlich, mittels Funktionen die absolute und relative Häufigkeit ermittelt.

Einzelaufgabe 2

Verwendetes Framework	Anzahl	relative Häufigkeit
React Native	33	20,75%
Flutter	54	33,96%
Android	34	21,38%
HTML	16	10,06%
nichts abgegeben	22	13,84%
	159	100,00%

Abbildung 27: Häufigkeit der verwendeten Frameworks für die Einzelaufgabe 2
Quelle: eigene Darstellung

Für die Einzelaufgabe wurde Flutter am häufigsten ausgewählt. Dieses Framework verwendeten insgesamt 54 Teilnehmerinnen und Teilnehmer, damit liegt es klar vor den anderen beiden Frameworks. React Native und Android liegen sehr knapp zusammen. HTML konnte nur von Studierenden gewählt werden, die für kein Studium an der Fakultät Informatik inskribiert waren. Diese Sprache liegt mit 16 Abgaben auf dem letzten Platz. 22 Personen haben die Einzelaufgabe 2 nicht fertiggestellt und daher auch nichts abgegeben.

Meilenstein 3 der Teamaufgabe

Verwendetes Framework	Anzahl	relative Häufigkeit
React Native	10	25,64%
Flutter	21	53,85%
Android	8	20,51%
nichts abgegeben	0	0,00%
	39	100,00%

Abbildung 28: Häufigkeit der verwendeten Frameworks für die Teamaufgabe
Quelle: eigene Darstellung

Der Meilenstein 3 war die Abgabe der App, die in der Gruppe erstellt wurde. Hierbei reichten insgesamt 39 Teams ihr Projekt ein, davon verwendeten mit insgesamt 21 Abgaben mehr als die Hälfte Flutter für ihre App. 10 Gruppen nahmen React Native als Framework, und 8 Gruppen nutzten Android.

6.3 Auswertung und Ergebnisse der qualitativen Daten

Die erhaltenen Daten der zwei offenen Fragestellungen wurden mit der qualitativen Inhaltsanalyse ausgewertet, die Kategorien wurden induktiv erstellt. Im folgenden Kapitel werden zu den beiden Fragestellungen die Kategorien mitsamt der Unterkategorien veranschaulicht.

6.3.1 Fragestellung „Besonders gut gefallen hat mir im Tutorium:“

Zu dieser Fragestellung haben sich vier Kategorien gebildet. Jede Kategorie wird mit einer Definition erklärt und enthält eine Tabelle mit den dazugehörigen Unterkategorien, Ankerbeispielen und der Kodierregel zu den Unterkategorien.

6.3.1.1 Medieneinsatz

Diese Kategorie enthält alle Antworten, die zu den „nichtpersonalen Medien“ gezählt werden, die im Tutorium verwendet wurden. In der Tabelle sind die drei Unterkategorien „Praxisbeispiele“, „Online-Videos“ und „Bereitgestellte Materialen“ mit den Ankerbeispielen und den Kodierregeln ersichtlich.

Unterkategorie	**Ankerbeispiel**	**Kodierregel**
Praxisbeispiele	„Das im Tutorium praktische Beispiele gezeigt wurden und nicht nur Theorie unterrichtet wurde“	Aussagen über Praxis-Beispiele und live programmierte Beispiele im Tutorium
Online-Videos	„Weiters positiv hervorzuheben ist dass die Tutorials im Anschluss Online gestellt wurden, so konnte ich mehrmals die Videos ansehen und besser verstehen.“	Aussagen über Online-Videos im Moodle-Kurs
Bereitgestellte Materialien	„…waren sehr hilfreich, ebenso wie die Verweise auf andere Seiten die Hilfestellung geben können“	Erwähnungen zu den zusätzlichen Materialien wie Powerpoint-Präsentation und den PDFs mit weiterführenden Links

Tabelle 1: Kategorie "Medieneinsatz" zur Fragestellung „Besonders gut gefallen hat mir im Tutorium:"

6.3.1.2 Interaktivität

In diese Kategorie fallen Antworten über die Kommunikation zwischen den Studierenden und dem Vortragenden. Dabei werden die Unterkategorien „Auf Fragen eingegangen“, „Themenwünsche“ und „Hilfsbereitschaft“ gebildet.

Unterkategorie	Ankerbeispiel	Kodierregel
Auf Fragen eingegangen	„Während der Einheiten finde ich muss man hervorheben, dass wirklich immer und ausführlich auf Fragen eingegangen wurde.“	Wenn erwähnt wurde, dass Fragen beantwortet wurden.
Themenwünsche	„Das Eingehen auf Themenwünsche der Studenten.“	Aussagen über Themenwünsche
Hilfsbereitschaft	„Man hat gemerkt der Tutor will wirklich helfen.“	Aussagen über die Hilfsbereitschaft des Tutors.

Tabelle 2: Kategorie "Interaktivität" zur Fragestellung „Besonders gut gefallen hat mir im Tutorium:"

6.3.1.3 Lehrkompetenz

Unter Lehrkompetenz wird die fachliche und didaktische Kompetenz verstanden, die von den Studierenden wahrgenommen wurde. Die aus den Aussagen gebildeten Untergruppen lauten „Tempo des Vortrags“, „Erklärung“, „Wissen des Leiters“, „Atmosphäre“ „Motivation“.

Unterkategorie	Ankerbeispiel	Kodierregel
Tempo des Vortrags	„Positiv war …sowie die Sprechgeschwindigkeit des Vortragenden.“	Aussagen über die Geschwindigkeit der Erklärungen und der Sprechgeschwindigkeit
Erklärung	„Es wurde sehr verständlich und gut nachvollziehbar erklärt.“	Erwähnungen über das Erklären der Inhalte
Wissen des Leiters	„Der Leiter kennt sich gut mit den Themen aus.“	Wenn über das Wissen und über die Kenntnisse des Tutors geschrieben wurde

Atmosphäre	„…lockere und teilweise lustige Atmosphäre.“	Wenn über die Atmosphäre im Tutorium geschrieben wurde
Motivation	„Enthusiasmus [sic] des Lehrenden für das Framework war gut erkennbar.“	Aussagen über die Motivation des Tutors

Tabelle 3: Kategorie "Lehrkompetenz" zur Fragestellung „Besonders gut gefallen hat mir im Tutorium:"

6.3.1.4 Ablauf

Diese Kategorie umfasst alle Meldungen zur Strukturierung und dem Ablauf der einzelnen Einheiten. Die Unterkategorien mit den Beispielen und der Kodierregel sind in der folgenden Tabelle ersichtlich.

Unterkategorie	Ankerbeispiel	Kodierregel
Didaktische Aufbereitung	„Es war didaktisch gut aufbereitet, also es wurde wirklich von 0 gestartet mit Installationseinführung usw.“ „Der Aufbau und die Strukturierung waren meiner Meinung sehr gut“	Erwähnungen über die Aufbereitung und Strukturierung des Kurses.
Umfang und Tiefe	„Es wurden fast alle Themen, welche für A2 benötigt wurden, abgedeckt.“ „ The tutorials were quite informative and extensive. I also learned about some additional features and tools of the language.”	Positive Aussagen über den Umfang und die Menge an Informationen.
Pausen in der Einheit	„kurze Pausen wurden bei Bedarf gemacht“	Wenn über eine Pause in der Einheit geschrieben wurde

Environment-Einrichtung	„Schritt für schritt erklärung [sic] des entwicklungs-environment“	Bei Aussagen über die Einrichtung des Environments

Tabelle 4: Kategorie "Ablauf" zur Fragestellung „Besonders gut gefallen hat mir im Tutorium:"

6.3.2 Fragestellung „Folgendes könnte im Tutorium verbessert werden:“

Für die Auswertung der zweiten Fragstellung werden zwei Hauptkategorien gebildet, die verschiedene Unterkategorien aufweisen. Diese sind mit Beispielen und Kodierregeln in einer Tabelle ersichtlich.

6.3.2.1 Ablauf des Tutoriums

Diese Kategorie beinhaltet Aussagen über den Ablauf und die Strukturierung des Tutoriums. Die dafür erstellten Unterkategorien lauten „Intensivierung bestimmter Themen“, „Schwierigkeit“, „Terminplanung“ und „Anzahl der Einheiten“.

Unterkategorie	Ankerbeispiel	Kodierregel
Intensivierung bestimmter Themen	„Man könnte noch etwas mehr auf die Interfacegestaltungsmöglichkeiten eingehen.“	Wenn geschrieben wurde, dass etwas mehr unterrichtet hätte werden sollen, oder gewisse Themen nicht erklärt wurden.
Schwierigkeit	„Die Geschwindigkeit von der Vorlesung war als Nicht-Informatiker ein bisschen schwer zum verstehen.“	Aussagen über die hohe Geschwindigkeit oder die Komplexität der Themen.
Terminplanung	„ Es waere meiner Meinung nach auch sinnvoll das Tutorium in kurzeren Abschnitten abzuhalten..“ „Das Zeitmanagement bzw. die Terminplanungen hätten etwas verbessert werden können.“	Aussagen über die Terminplanung der Einheiten.

Anzahl der Einheiten	„ Es wurde sehr viel Stoff behandelt in wenig Zeit behandelt, aber das lag wahrscheinlich an der knappen Stundenanzahl.“	Erwähnungen der Dauer des Tutoriums.

Tabelle 5: Kategorie "Ablauf des Tutoriums" zur Fragestellung „Folgendes könnte im Tutorium verbessert werden:“

6.3.2.2 Medieneinsatz

In der Kategorie *Medieneinsatz* werden Erwähnungen hinsichtlich der personalen als auch der nichtpersonalen Medien aufgelistet. In der folgenden Tabelle sind die Unterkategorien mit den Ankerbeispielen und den Regeln dargestellt.

Unterkategorie	Ankerbeispiel	Kodierregel
Video-Aufzeichnung	„ The stream quality of the Tutorium was sometimes poor, hence it was difficult to follow it live, but this was purely a technical issue.”	Erwähnungen über die Qualität der Tutorium-Aufzeichnung
Qualität der zusätzlich bereitgestellten Materialien	„ Die Materialien koennten ein bisschen besser aufbereitet werden bzw. hin und da eine best practice mit einbauen.“	Aussagen über die bereitgestellten Materialien mit Ausnahme der Aufzeichnung
Praxisbeispiele	„ Eventuell mehr live-Programmieren,“	Wenn die Praxisbeispiele und das live-Coding erwähnt wurde
Erklärung	„Flutter-Funktionen nicht gut erklärt beziehungsweise auch mehr erläutern, warum gerade dieser Ansatz gewählt wird / dies so gemacht wurde.“	Aussagen über die Vermittlungsfähigkeit des Lehrenden.

Tabelle 6: Kategorie "Medieneinsatz" zur Fragestellung „Folgendes könnte im Tutorium verbessert werden:“

7 Diskussion

Die im vorherigen Kapitel ausgewerteten Daten werden nun zur Diskussion herangezogen, um die Forschungsfrage zu beantworten. Im ersten Abschnitt wird das Tutorium vom Leiter reflektiert. Danach erfolgt eine Interpretation der quantitativen und qualitativen Ergebnisse. Dabei wird immer wieder auf die Fachliteratur zurückgegriffen und mit den Ergebnissen verglichen. Abschließend wird in der Zusammenfassung anhand der interpretierten Daten die Forschungsfrage beantwortet.

7.1 Didaktische Reflexion des Tutoriums

Um die Forschungsfrage ausreichend zu beantworten, bedarf es einer didaktischen Reflexion des Tutoriums. Dabei werden die Planung und Durchführung des Tutoriums anhand der eigenen Einschätzung des Tutoriumleiters analysiert und bewertet.

7.1.1 Positiv hervorzuheben

Die Strukturierung der einzelnen Einheiten, welche eine zeitliche Begrenzung von 90 Minuten aufwiesen, funktionierte sehr gut. Die Einheiten wurden so geplant, dass immer ein in sich geschlossener Themenblock vorgetragen und in einem Praxisbeispiel präsentiert wurde. In der Unterrichtsplanung wurde für die einzelnen Teilsequenzen die dafür benötigte Zeit zugewiesen. Ein Zeitpuffer zwischen den einzelnen Sequenzen wurde für Fragen der Studierenden oder technische Probleme reserviert. In der Durchführung konnten die zeitlichen Vorgaben sehr gut eingehalten werden. Dadurch wurde jeder Themenkomplex in der vorhergesehenen Zeit abgehandelt als auch Fragen der Teilnehmerinnen und Teilnehmer ausreichend beantwortet.

Alle Fragen zur Programmiersprache Dart und dem Framework Flutter konnten im Tutorium beantwortet werden. Die Fragen wurden einerseits direkt beantwortet, oder zu Beginn der nächsten Einheit. Dies war manchmal erforderlich, da die Beantwortung einige Zeit in Anspruch nahm und für eine bessere Vermittlung eine Powerpoint-Folie zur Veranschaulichung erstellt wurde.

Das Vermitteln der einzelnen Konzepte des Frameworks und der Anwendung der Programmiersprache Dart lief aufgrund des vorherigen Übens sehr gut. Um gewisse Konzepte einfach zu erklären, bedarf es einer Sprache, die verständlich für alle anwesenden Teilnehmerinnen und Teilnehmer ist und zu keiner Verwirrung führt. Durch das vorherige Üben konnte schon erprobt werden, welche Wortwahl für die Vermittlung passend ist.

Aufgrund der Distanzlehre wurde das Tutorium über die Software Zoom übertragen. Dies funktionierte einwandfrei, wodurch alle Interessierten den Stream sahen, über ein Chat-Fenster Fragen stellten, und über die Moodle-Plattform die zur Verfügung gestellten Materialien downloaden konnten. Ebenso konnte beim Halten des Tutoriums das jeweilige Fenster für den Stream ausgewählt, und etwaige Fragen über das Chat-Fenster im Blick gehalten werden. Zoom eignete sich hierzu als sehr gute Software für die Online-Lehre.

7.1.2 Verbesserungsmöglichkeiten

7.1.2.1 Aufnahme des Tutoriums

Wie bereits in Kapitel 5.3.2 beschrieben, gab es Probleme bei der Aufnahme des Tutoriums. Das Problem bei der Aufzeichnung lag an der übermäßigen Verwendung der Hardware-Ressourcen des aufzeichnenden Notebooks. Es ist nicht eindeutig zu identifizieren, welche Komponente den „Flaschenhals“ in der Aufzeichnung darstellte. Der mit 16 Gigabyte zur Verfügung stehende Arbeitsspeicher reichte für die verwendeten Anwendungen. Vermutlich wurde die GPU des AMD Ryzen 5 Pro 2500 überlastet. In der Tutoriums-Einheit belasteten folgende Anwendungen die Grafikkarte:

- Zoom mit der Übertragung des Monitors und der Videokamera
- Visual Studio Code mit dem Ausführen der erstellten App
- Powerpoint
- Android Virtual Device Emulator, der ein ganzes Android-System mit der programmierten App simulierte
- OBS Studio mit der Aufnahme des Bildschirms

Um alle Anwendungen und den Zoom-Chat im Überblick zu behalten, wurden insgesamt drei aktive Monitore verwendet, die die Grafikkarte zusätzlich belasteten. Die Kombination aus all diesen Faktoren könnte die niedrige Bildrate in der Aufnahme erklären.

Damit dieses Problem bei zukünftigen Projekten vermieden wird, muss die Auslastung des Systems zuvor simuliert und das Videoergebnis analysiert werden. Falls die Überlastung erneut entstehen sollte, kann eine Premium-Version von Zoom verwendet werden, die eine automatische Aufnahmefunktion enthält, oder die Aufnahme wird auf einem leistungsstärkeren System durchgeführt. Eine Reduzierung der Monitore oder der Anwendung würde das Tutorium in seiner Qualität vermutlich negativ beeinflussen, wovon abzuraten ist.

7.1.2.2 Terminfindung

Da das Flutter-Tutorium an mehreren Terminen stattfand, waren passende Termine notwendig, die nicht mit den Zeiten anderer Lehrveranstaltungen kollidierten. Dabei wurden zwei Termine kurzfristig verschoben, und die Termine fanden nicht regelmäßig zu den gleichen Zeiten statt.

Hilfreicher wäre es, bereits vor der ersten Einheit zu erheben, zu welcher Zeit das Tutorium besucht werden kann, und danach die einzelnen Einheiten in regelmäßigen Abständen zur jeweils gleichen Zeit abzuhalten.

7.1.2.3 Übungsaufgaben bereitstellen

Um das vermittelte Wissen zu festigen, wären spezifische Übungsaufgaben zu dem jeweiligen Themenkomplex eine sinnvolle Ergänzung. Die Studierenden haben in der Lehrveranstaltung bereits sehr viele Abgaben und der Workload ist dadurch dementsprechend hoch. Wenn die Übungsaufgaben so gestaltet werden, dass die Studierenden die erstellte Lösung in den weiteren Abgaben wiederverwenden können, wäre der Ansporn der Studierenden zum Lösen der Beispiele dementsprechend erhöht.

7.1.2.4 Mehr Einheiten

In den gehaltenen fünf Einheiten wurde ein großer Teil des Frameworks und der verschiedenen Funktionen vorgestellt. Verschiedene Best-Practices, Design-Pattern und Implementierungsmöglichkeiten mit deren Vor- und Nachteile fanden aufgrund der limitierten Zeit keinen Platz. Ein Umfang von zehn Einheiten zu jeweils 90 Minuten würde den notwendigen Platz schaffen. Ein weiterer Vorteil würde sich in der Methodenvielfalt ergeben. Verschiedene Umsetzungsmöglichkeiten könnten in Gruppenarbeiten von den Teilnehmerinnen und Teilnehmern erarbeitet und im Anschluss diskutiert werden, ebenso wäre Platz für Einzelaufgaben während der Einheit.

7.2 Interpretation der quantitativen Ergebnisse

Im folgenden Abschnitt werden die quantitativen Ergebnisse aus Kapitel 6.1 und 6.3 interpretiert.

Bei der Auswertung der neun gestellten quantitativen Fragestellungen wurde festgestellt, dass die Frameworks sehr ähnlich bewertet wurden. Die Vermittlung der Inhalte, die Strukturierung des Tutoriums und die bereitgestellten Lernmaterialien wurden in allen drei Tutorien ähnlich bewertet. Ebenso waren diese für die Erstellung des Interfaces und der Funktionen in gleichem Maße hilfreich. Bei der Auswertung der subjektiv empfundenen Verbesserung der Programmierkenntnisse konnte ebenfalls kein signifikanter Unterschied festgestellt werden.

Das Flutter-Tutorium hatte mit insgesamt vier Einheiten zu je 90 Minuten, und einer Fragestunde, einen größeren Umfang als die anderen beiden Tutorien, welche zwei Einheiten von je 90 Minuten umfassten, in der in einer Einheit das Framework und die Einrichtung erklärt wurde, und die zweite Einheit als Fragestunde diente. Anhand des Fragebogens brachte der größere Umfang des Tutoriums keine signifikante Verbesserung für die Studierenden, wurde jedoch auch nicht schlechter bewertet. Möglicherweise informierten sich die Studierenden selbst ausreichend über die Frameworks und konnten sich mit verschiedenen im Internet frei verfügbaren Materialien die Programmiersprache ausreichend aneignen.

Die Studierenden bewerteten die drei Tutorien hinsichtlich der Unterstützung für die Bearbeitung der Aufgaben sehr ähnlich, wie in den Abbildungen 19, 21 und 25 ersichtlich ist. Wenn man die quantitativen Daten der Lehrveranstaltungsabgaben hierbei miteinbezieht, kann jedoch ein Trend hinsichtlich der Unterstützung erkannt werden. Wie in der Abbildung 27 ersichtlich, verwendeten 54 Teilnehmerinnen und Teilnehmer für die Einzelaufgabe 2 das Framework Flutter. Diese Anzahl ist deutlich höher als bei den anderen beiden Tutorien. Für den Meilenstein 3 im Teamprojekt, der die Entwicklung der App kennzeichnet, entschieden sich laut Abbildung 28 insgesamt 21 Gruppen, und somit mehr als die Hälfte aller Teams, für den Einsatz des Frameworks Flutter. Hieraus kann geschlossen werden, dass noch einige Studierende nach der Einzelaufgabe 2 das Framework wechselten und Flutter zusätzlich erlernten. Der große Umfang des Tutoriums wird hier entscheidend gewesen sein. Wenn die Personen im Team unterschiedliche Frameworks für die Einzelaufgabe 2 verwendeten und sich jetzt im Team auf ein Framework einigen mussten, wurde vermutlich das mit dem größeren Angebot gewählt.

In der Auswertung des Fragebogens konnte in der Fragestellung „Die Fragen der Studierenden wurden im Tutorium zufriedenstellend beantwortet“ ein signifikanter Unterschied zwischen den Tutorien erkannt werden. Da das Flutter Tutorium häufiger abgehalten wurde, konnten dementsprechend mehr Fragen beantwortet werden. Einerseits stand mehr Zeit zur Beantwortung von Fragen zur Verfügung, durch das regelmäßige Halten ergaben sich jedoch auch in der Zeit zwischen den Einheiten Fragen, die dann in der nächsten Einheit oder im Moodle-Forum gestellt wurden. Ebenfalls konnte auf spezielle Fragen näher eingegangen werden, indem diese bei der folgenden Einheit grafisch aufbereitet und im Detail erklärt wurden.

Das Flutter Tutorium schnitt bei den eingesetzten Medien nicht so gut ab wie die anderen beiden Tutorien. Dies ist in der Abbildung 15 zu erkennen. In Kapitel 7.1.2.1 wurde die Ursache bereits näher erläutert. Aufgrund dieser Probleme kam es hier zu einer schlechteren Bewertung. In Anbetracht der Abbildungen 27 und 28 wurde Flutter dennoch am häufigsten verwendet.

7.3 Interpretation der qualitativen Ergebnisse

Für die Beantwortung der Forschungsfrage sind die qualitativen Ergebnisse von entscheidender Bedeutung. In diesem Abschnitt werden Teile der Ergebnisse mit der in Kapitel 2 verwendeten Literatur verglichen und interpretiert. Auszüge aus den Daten liefern einen zusätzlichen Eindruck, den die Studierenden vom Tutorium hatten.

Eine Hauptkategorie der Fragestellung über die positiven Aspekte lautet „Interaktivität“ und beinhaltet Aussagen über die Kommunikation zwischen dem Lehrenden und den Lernenden. Sehr häufig empfanden die Kursteilnehmerinnen und Kursteilnehmer das Beantworten und Eingehen von Fragen während dem Tutorium als positiv.

„Es wurde immer umgehend auf Fragen der Studierenden eingegangen.“

„Die offene Haltung des Tutoriumsleiters, auch in Bezug auf die Beantwortung etwaiger Fragen im Chat, ist mir jedoch positiv aufgefallen. Das lief im allgemeinen relativ flüssig, im Vergleich zu anderen Vortragenden, bei denen man schon gute 20 Minuten auf seine Antwort warten darf.“

„Das individuell auf Fragen einegangen wurde und diese in der nächsten Einheit besprochen wurden.“

Zu den Kernaufgaben des Tutors und der Tutorin gehört die Beantwortung von Fragen. [42] Für einen guten Unterricht ist es auch wichtig, dass die Lernenden Fragen stellen dürfen. In Kapitel 2.1 spricht Hilbert Meyer über die wichtigen Merkmale, wobei hier das sinnstiftende Kommunizierte ein Merkmal darstellt. Im ARCS-Modell von Keller (Kapitel 2.2.4) steigert das Ermöglichen von Nachfragen die Motivation der Schülerinnen und Schüler. Diese Punkte wurden in Anbetracht der qualitativen Daten berücksichtigt.

Die Hauptkategorie „Medieneinsatz“ wurde bei den positiven als auch negativen Erwähnungen gebildet. Die dazugehörigen Unterkategorien sind in Tabelle 1 und Tabelle 6 zu finden. Insgesamt sieben positive Erwähnungen gab es zu den gewählten Praxis-Beispielen im Tutorium, dreimal wurde die Live-Programmierung dieser Beispiele angeführt.

„Das im Tutorium praktische Beispiele gezeigt wurden und nicht nur Theorie unterrichtet wurde“

„Flutterinho etc. waren nette Beispiele.“

„Dass anhand von vorprogrammierten Beispielen auf die Funktionen von Flutter eingegangen wurde…“

„Die Praxisnahe Herangehensweise“

„dass der Lehrer manchmal live programmiert hat, hilft es uns Schülern sehr, nicht nur den Code zu erklären, sondern den Code gleichzeitig zu schreiben und zu erklären.“

In Kapitel 2.3.2 wurde die Veranschaulichung im Unterricht und deren Vorteile beschrieben. Durch die Live-Programmierung und den Beispielen war es für die Studierenden leichter, die Funktionen von Flutter zu verstehen. Einige Studierende empfanden diese Live-Programmierung nicht sinnvoll, und hätten lieber einen groben Überblick über die Möglichkeiten bekommen.

„Ich denke, das "Live mitprogrammieren" ist weniger sinnvoll als einfach nur einen Überblick geben über die Möglichkeiten und grob, wie man häufige Problempatterns angeht (zb Provider)“

Insgesamt gab es jedoch nur zwei Meldungen in der Unterkategorie „Praxisbeispiele“ (Tabelle 6), die weniger Live-Coding sinnvoller finden würden. Hingegen teilten sechs Personen mit, dass zu wenig live programmiert wurde.

Der größte Kritikpunkt im Tutorium war die die schlechte Qualität der Aufzeichnung. Insgesamt 28-Mal wurde dies von den Studierenden erwähnt. Das Problem und die mögliche Ursache wurden bereits in Kapitel 7.1.2.1 beschrieben. Da manche Studierenden nicht bei der Präsenz-Einheit teilnehmen konnten, und dadurch von der Aufzeichnung abhängig waren, hatte diese einen sehr hohen Stellenwert.

„Die zur Verfügung gestellten Videos hängen sehr hinterher, das heißt es wird z.B.: gesagt "Wir drücken hier installieren" aber das Video ist noch gar nicht an diesem Punkt. Das macht das lernen von den Tutorium Aufnahmen eher schwierig.“

„Das einzige, was ich vebessern [sic] wuerde ist die Aufzeichnung der Tutoriums. Das Bild hat immer gehaengt, deshalb konnte man es gar nicht verstehen und folgen. Aus diesem Grund waren mir die Tutoriums nicht hilfreich. Weil waehrend des Tutorium hoert man mehr zu und kann nicht gleichzeitig programmieren. Und die Aufzeichnungen sollten dabei helfen, alles in Ruhe nach dem Tutorium anzuschauen, was leider nicht moeglich war.“

Die Aufnahme des Tutoriums war jedoch nur ein zusätzliches Angebot, falls der Studierende nicht beim Termin teilnehmen konnte. Sieben Personen beschrieben die Online-Videos auch positiv.

Die zusätzlich bereitgestellten Materialien wurden insgesamt siebenmal positiv und zweimal negativ bewertet. Dabei wurde auf die Powerpoint-Folien, die weiterführenden Links und den Code-Beispielen Bezug genommen.

„Die Codebeispiele waren sehr hilfreich, ebenso wie die Verweise auf andere Seiten die Hilfestellung geben können.“

„Der Leiter des Tutorium [sic] hat sehr gute Präsentationsfolien bereitgestellt…“

„Die Folien sind oft kaum aussagekräftig außer man schaut sich dazu parallel die Einheit/das Video an, ich habe deshalb auch bei späteren Rückfragen eigentlich nur mit den Videos gearbeitet.“

Die Lehrkompetenz wurde von den Studierenden überwiegend positiv gewertet. Es gab in dieser Kategorie insgesamt 19 positive Erwähnungen bezüglich der Fähigkeiten zum Lehren. Eine Person empfand die Vortragsweise nicht *„flüssig“.* Zu dieser Kategorie gehören die in Tabelle 3 aufgelisteten Unterkategorien. Damit die Studierenden dem Vortrag folgen können, und die gelehrten Themen verstehen, bedarf es einer einfachen und nachvollziehbaren Sprache, die Stimme und Artikulation muss ebenfalls an die Situation angepasst werden. (Kapitel 2.3.3) Die in den Unterkategorien „Tempo des Vortrags“ und „Erklärung“ aufgelisteten Aussagen beziehen sich auf diese Faktoren.

„Es wurde sehr verständlich und gut nachvollziehbar erklärt.“

„Die Inhalte wurden verständlich und klar erklärt

Insgesamt 13 positive Erwähnungen gab es zu diesen beiden Unterkategorien. Die vom Tutoriumsleiter im Studium gesammelte theoretische und praktische Erfahrung trägt natürlich einen Teil bei. Um Anderen etwas scheinbar Komplexes zu lehren, bedarf es einer einfachen Sprache. Der Tutor versuchte dies auch bei der Funktionsweise hinter Flutter und bei der Erklärung gewisser Pattern. Um etwas verständlich zu erklären, muss auch das nötige Wissen über das Thema vorhanden sein. Dies wurde von den Studierenden zweimal erwähnt.

„Gute Kenntnisse in der Materie“

„Der Leiter kennt sich gut mit den Themen aus…“

Die Atmosphäre spielt im Unterricht ebenfalls eine Rolle, und wurde von den Teilnehmerinnen und Teilnehmern erwähnt. In einem Online-Kurs wird diese vom Organisator in großem Maße beeinflusst. Diese Person kann bestimmen, ob die Atmosphäre zum erfolgreichen Lernen anregt. Wie in Kapitel 2.1 und 2.2.3 beschrieben, wird eine angenehme Atmosphäre für einen guten Unterricht benötigt.

„Sehr sympathischer Vortragender, der auch über kleine Ungereimtheiten lachen konnte.“

„…lockere und teilweise lustige Atmosphäre…“

Zweimal wurde von den Studierenden die Motivation des Tutors positiv bewertet.

„Motivierter Tutor,..“

„Enthusiasmus des Lehrenden für das Framework war gut erkennbar.“

Wie in Kapitel 2.2 bereits ausführlich beschrieben, hat die Lernmotivation vor allem im Distance-Learning einen erheblichen Einfluss auf den Lernerfolg. In einer Studie wurde ein Zusammenhang zwischen der Motivation der Lehrperson und der folgenden Motivation der Schülerinnen und Schüler zum Unterrichtsthema untersucht. Es wurde herausgefunden, dass eine sehr motivierte Lehrperson einen positiven Einfluss auf die Lernenden hat. [43]

Über den Ablauf der Lehrveranstaltung gab es sehr viel Feedback. Insgesamt gab es zu den in Tabelle 4 und Tabelle 6 abgebildeten Unterkategorien 26 positive Meldungen und 29 Verbesserungsvorschläge. Insgesamt 14-Mal wurde dabei der große Umfang des Tutoriums gelobt. Die Studierenden gaben an, dass sie für das kommende Projekt sehr gut vorbereitet wurden und die relevanten Funktionen präsentiert bekamen.

„Die Tiefe des Tutoriums und die zur Verfügung gestellten Code-Materialien waren sehr hilfreich. Andere kostenlose Tutorien auf Youtube sind sehr oberflächlich.“

„Die vorgestellten Flutter- Package und Funktionen sind sehr relevant für unsere Projekte.“

Manchen Studierenden fehlten jedoch gewisse Themen im Vortrag oder wurden nicht ausreichend behandelt. Man hätte mehr auf die Interfacegestaltung, auf mögliche Fehlermeldungen, auf die verschiedenen States der Widgets in Flutter, und auf die „*Pattern wie bloc oder Provider“* eingehen sollen. Nicht optimal funktionierten die Terminplanung und das Zeitmanagement. Die Einheiten sollten generell in kürzeren Abständen abgehalten werden, damit mehr Zeit für die Entwicklung der Apps vorhanden ist. Die größten Schwierigkeiten hatten manche Studierenden mit der Komplexität des Themas und der dadurch hohen Geschwindigkeit in den Einheiten. Bei acht Personen steigerte sich die Komplexität zu schnell, wodurch sie überfordert waren. An der Lehrveranstaltung nahmen auch Studierende teil, die noch nie programmiert hatten. Fünf der acht Meldungen beschriebenen in ihren Aussagen, dass sie „Nicht-InformatikerInnen“ waren. Der Kurs wurde jedoch für „InformatikerInnen“ geplant, daher wurden die Grundlagen der Programmierung nicht erklärt.

Die didaktische Aufbereitung wurde achtmal positiv hervorgehoben. Die Studierenden teilten mit, dass die Inhalte gut strukturiert und aufbereitet wurden.

„Die Aufteilung in sinnvolle Themenkomplexe war sehr angenehm…“

„Der Aufbau und die Strukturierung waren meiner Meinung sehr gut…“

Für Hilbert Meyer ist die klare Strukturierung, und der „rote Faden“, ein wesentliches Merkmal guten Unterrichts (Kapitel 2.1).

7.4 Zusammenfassung

Zusammenfassend kann anhand der Interpretation der quantitativen und qualitativen Ergebnisse geschlussfolgert werden, dass das Tutorium zum Cross Platform Framework Flutter einen zusätzlichen Mehrwert für die Kompetenzentwicklung der Studierenden bietet.

Die quantitativen Ergebnisse zeigen, dass das Flutter-Tutorium von den Studierenden sehr ähnlich bewertet wurde wie die bereits existierenden Tutorien. Aufgrund des höheren Umfangs konnten Fragen der Studierenden besser beantwortet und insgesamt mehr Themen behandelt werden. Die quantitativen Daten zu den Abgaben in der Lehrveranstaltung bieten einen noch besseren Einblick. Da sich mehr als die Hälfte der Teams für das Framework Flutter entschied, wurde das Tutorium für diese Abgabe dann zusätzlich von Personen verwendet, die bereits ein anderes Framework erlernten.

Ein noch tieferer Einblick wird bei den qualitativen Ergebnissen ermöglicht. Die vier Kategorien zur ersten Fragestellung zeigen, dass das Tutorium insgesamt als sehr hilfreich empfunden wurde. Die Praxisbeispiele unterstützten mitsamt den Online-Videos und den bereitgestellten Materialien die Studierenden beim Erlernen von Flutter. Der Tutoriumsleiter wurde in seiner Rolle als Vermittler als kompetent eingestuft und konnte das Thema gut erklären, Fragen beantworten und eine angenehme Atmosphäre schaffen. Positiv wurde auch der ganze Ablauf und die didaktische Aufbereitung des Tutoriums empfunden.

Probleme gab es bei der Aufnahme des Tutoriums und der Terminplanung. Die niedrige Qualität der Aufzeichnung spiegelte sich in den quantitativen und qualitativen Daten deutlich wider. Für Studierende, die in der Einheit nicht „in Präsenz“ teilnehmen konnten, stellte das Lernen mit der Aufzeichnung eine gewisse Hürde dar. Wie bereits beschrieben gibt es für dieses Problem jedoch Lösungsmöglichkeiten.

Abschließend müssen bei der Beantwortung der Forschungsfrage auch einige Limitationen berücksichtigt werden. Der quantitativ-qualitative Fragebogen wurde nur einmal durchgeführt, wodurch die Reliabilität nicht gegeben ist. Die Fragestellungen wurden im Vorfeld auch nicht nach den wissenschaftlichen Standards validiert. [44] Ebenfalls wurde die tatsächliche Kompetenzentwicklung der Studierenden nicht erhoben. Dafür wäre der vorherige Leistungsstand sowie der Leistungsstand nach der Lehrveranstaltung und für eine längerfristige Einschätzung der Leistungsstand nach Studienabschluss notwendig. Diese Daten lagen jedoch nicht vor und eine Auswertung dieser hätte den Workload der Arbeit erheblich überschritten.

8 Fazit und Ausblick

Diese Arbeit beschäftigte sich mit der Erstellung eines didaktisch aufbereitenden Tutoriums, das in der der Lehrveranstaltung Mensch-Computer-Interkation abgehalten wurde. Zu Beginn wurde die eigene Motivation hinsichtlich dieser Arbeit und der grundsätzliche Aufbau der Arbeit erläutert. In der theoretischen Rahmung sind wichtige Werke der Pädagogik und Didaktik enthalten. Die verwendete Fachliteratur behandelt die Merkmale eines guten Unterrichts, einen theoretischen Einblick in die Lernmotivation, in die Distanzlehre und weitere Unterrichtsstrategien. Ein fachlicher Einblick in die Mensch-Computer-Interkation und den Cross-Platform Frameworks wird in Kapitel 3 und 4 ermöglicht. Anschließend wurde ein Tutorium für das Cross-Platform Framework Flutter erstellt, dass die didaktischen Merkmale aus der theoretischen Rahmung berücksichtigt. Das Tutorium wurde im Zuge der Lehrveranstaltung Mensch-Computer-Interaktion an fünf Terminen abgehalten. Moodle diente als unterstützende Lernplattform. Über diese Plattform wurde nach den Einheiten allen Teilnehmern der Lehrveranstaltung ein qualitativ-quantitativer Fragebogen ausgehändigt. Die quantitativen Daten wurden mit Excel ausgewertet, bei den qualitativen Daten wurden mittels induktiver Kodierung insgesamt sechs Kategorien gebildet. Die gewonnenen Daten wurden im Diskussionsteil analysiert und interpretiert. Eine eigene Reflexion ist in diesem Teil ebenfalls zu finden. Die Daten lieferten als Ergebnis, dass das Tutorium einen Mehrwert für die Studierenden bot, somit konnte die Forschungsfrage beantwortet werden.

Weiterführende Forschungsprojekte können an diese Arbeit anknüpfen und neue Ergebnisse liefern. Die Überprüfung der fachlichen Kompetenzentwicklung der am Tutorium teilnehmenden Studierenden wäre ein sehr interessantes Forschungsthema. Dafür müssen jedoch, wie im vorherigen Kapitel beschrieben, der bisherige Leistungsstand und der Leistungsstand zu Studienabschluss erhoben werden. Ein weiteres interessantes Forschungsfeld könnte die gewonnenen Programmierkenntnisse von fachfremden Studierenden, die in der Lehrveranstaltung Mensch-Computer-Interaktion teilnehmen und das Tutorium besuchen, untersuchen. Die in dieser Arbeit erhobenen Daten zeigen, dass die zu erreichenden Ziele für Programmieranfängerinnen und Programmieranfänger sehr schwer zu erreichen waren, eine genaue Datenerhebung fehlt jedoch. Die Weiterentwicklung des Flutter-Tutoriums würde sich ebenfalls als Forschungsarbeit anbieten. Hierbei könnte der Kurs für eine Präsenzlehre an der Universität angepasst werden. Anstatt dem darbietenden Lehrverfahren (Kapitel 2.3.4) könnten auch alternative Lehrverfahren und deren Auswirkung auf die Kompetenzentwicklung der teilnehmenden Studierenden untersucht werden. Die auf diese Arbeit möglichen aufbauenden Forschungsarbeiten sind vielfältig und würden die Lehrveranstaltungen an der Fakultät für Informatik Wien stetig verbessern und neue Erkenntnisse liefern.

9 Abbildungsverzeichnis

Abbildung 1: ARCS-Modell Matrix 9
Abbildung 2: Gestaltungsebenen der Lehr-Lernziele nach Reinmann 14
Abbildung 3: Modell des Arbeitsgedächtnis 22
Abbildung 4: Präattentive Wahrnehmung 25
Abbildung 5: Simultankontrast 26
Abbildung 6: Hermann-Gitter 26
Abbildung 7: didaktischer Aufbau der Tutoriums-Einheiten 36
Abbildung 8: Screenshot einer Live-Programmierung 37
Abbildung 9: visuelle Auswertung zur Fragestellung "Die Inhalte des Tutoriums wurden verständlich vermittelt" 47
Abbildung 10: quantitative Daten zur Fragestellung "Die Inhalte des Tutoriums wurden verständlich vermittelt" 47
Abbildung 11:visuelle Auswertung zur Fragestellung "Die Inhalte wurden strukturiert aufgebaut " 48
Abbildung 12: quantitative Daten zur Fragestellung "Die Inhalte wurden strukturiert aufgebaut " 48
Abbildung 13: visuelle Auswertung zur Fragestellung "Die bereitgestellten Lernmaterialien waren hilfreich" . 49
Abbildung 14: quantitative Daten zur Fragestellung "Die bereitgestellten Lernmaterialien waren hilfreich" ... 49
Abbildung 15: visuelle Auswertung zur Fragestellung "Die im Tutorium eingesetzten Medien waren sinnvoll" 50
Abbildung 16: quantitative Daten zur Fragestellung "Die im Tutorium eingesetzten Medien waren sinnvoll" ... 50
Abbildung 17: visuelle Auswertung zur Fragestellung "Die Fragen der Studierenden wurden im Tutorium zufriedenstellend beantwortet" 51
Abbildung 18: quantitative Daten zur Fragestellung "Die Fragen der Studierenden wurden im Tutorium zufriedenstellend beantwortet" 51
Abbildung 19: visuelle Auswertung zur Fragestellung "Das Tutorium war für die Implementierung der Funktionen hilfreich" 52
Abbildung 20: quantitative Daten zur Fragestellung "Das Tutorium war für die Implementierung der Funktionen hilfreich" 52
Abbildung 21: visuelle Auswertung zur Fragestellung "Das Tutorium war für das Erstellen des Interfaces hilfreich" 53
Abbildung 22: quantitative Daten zur Fragestellung "Das Tutorium war für das Erstellen des Interfaces hilfreich" 53
Abbildung 23: visuelle Auswertung zur Fragestellung "Meine Programmierkenntnisse haben sich durch das Tutorium insgesamt verbessert" 54
Abbildung 24: quantitative Daten zur Fragestellung "Meine Programmierkenntnisse haben sich durch das Tutorium insgesamt verbessert" 54
Abbildung 25: visuelle Auswertung zur Fragestellung "Ich fühle mich nun für die App Entwicklung im Teamprojekt (Meilenstein 3) gut vorbereitet 55

Abbildung 26: quantitative Daten zur Fragestellung "Ich fühle mich nun für die App Entwicklung im Teamprojekt (Meilenstein 3) gut vorbereitet 55
Abbildung 27: Häufigkeit der verwendeten Frameworks für die Einzelaufgabe 2 56
Abbildung 28: Häufigkeit der verwendeten Frameworks für die Teamaufgabe 56

10 Tabellenverzeichnis

Tabelle 1: Kategorie "Medieneinsatz" zur Fragestellung „Besonders gut gefallen hat mir im Tutorium:" 57
Tabelle 2: Kategorie "Interaktivität" zur Fragestellung „Besonders gut gefallen hat mir im Tutorium:" 58
Tabelle 3: Kategorie "Lehrkompetenz" zur Fragestellung „Besonders gut gefallen hat mir im Tutorium:" 59
Tabelle 4: Kategorie "Ablauf" zur Fragestellung „Besonders gut gefallen hat mir im Tutorium:" 60
Tabelle 5: Kategorie "Ablauf des Tutoriums" zur Fragestellung „Folgendes könnte im Tutorium verbessert werden:“ 61
Tabelle 6: Kategorie "Medieneinsatz" zur Fragestellung „Folgendes könnte im Tutorium verbessert werden:“ . 61

11 Literaturverzeichnis

[1] H. Bittermann und G. Cavallar, „Schule: Mehr digitale Inkompetenz?“, 17. September 2021. [Online]. Verfügbar unter: https://www.derstandard.at/story/2000129711538/schule-mehr-digitale-inkompetenz
[2] Universität Wien, „Teilcurriculum für das Unterrichtsfach Informatik im Rahmen des Masterstudiums zur Erlangung eines Lehramts im Bereich der Sekundarstufe (Allgemeinbildung) im Verbund Nord-Ost“. 29. Juni 2017. [Online]. Verfügbar unter: https://senat.univie.ac.at/fileadmin/user_upload/s_senat/konsolidiert_Lehramt/Teilcurriculum_Informatik_MA_Lehramt.pdf
[3] Universität Wien, Hrsg., „Teilcurriculum für das Unterrichtsfach Informatik im Rahmen des Bachelorstudiums zur Erlangung eines Lehramts im Bereich der Sekundarstufe (Allgemeinbildung) im Verbund Nord-Ost“. 27. Juni 2016. [Online]. Verfügbar unter: https://senat.univie.ac.at/fileadmin/user_upload/s_senat/konsolidiert_Lehramt/Teilcurriculum_Informatik_BA_Lehramt.pdf
[4] Universität Wien, „Human-Computer-Interaction Vorbesprechung und Einführung“, 2021. [Online]. Verfügbar unter: http://vda.univie.ac.at/Teaching/HCI/21s/LectureNotesOpen/01_VorbesprechungHCI.pdf
[5] H. Meyer, *Was ist guter Unterricht?*, 15. Auflage. Berlin: Cornelsen, 2021.
[6] W. Wiater, *Unterrichten und Lernen in der Schule: eine Einführung in die Didaktik*, 3. Aufl. Donauwörth: Auer, 2015.
[7] W. Stangl, „Lernmotive und Lernmotivation“. 2020. [Online]. Verfügbar unter: https://arbeitsblaetter.stangl-taller.at/MOTIVATION/Lernmotivation.shtml
[8] S. Theobald, *Lernmotivation: Stationsarbeit im Vergleich zu Frontalunterricht*. Hamburg: Diplomica Verl, 2012.
[9] J. Zylka, Hrsg., *Schule auf dem Weg zur personalisierten Lernumgebung: Modelle neuen Lehrens und Lernens*, 1. Auflage. Weinheim Basel: Beltz, 2017.
[10] A. Wipper und A. Schulz, *Digitale Lehre an der Hochschule: vom Einsatz digitaler Tools bis zum Blended-Learning-Konzept*. Opladen Toronto: Verlag Barbara Budrich, 2021.
[11] L. Haag, S. Rahm, H. J. Apel, und W. Sacher, Hrsg., *Studienbuch Schulpädagogik*, 5., Vollst. überarb. Aufl. Bad Heilbrunn: Klinkhardt, 2013.

[12] M. Spitzer, *Lernen: Gehirnforschung und die Schule des Lebens*, 1. Aufl. München: Spektrum Akademischer Verlag, 2007.
[13] H. Kiper, *Einführung in die Schulpädagogik*. Weinheim Basel: Beltz Verlag, 2001.
[14] H. Glöckel, *Vom Unterricht: Lehrbuch der allgemeinen Didaktik*, 4., Durchges. und erg. Aufl. Bad Heilbrunn/Obb: Klinkhardt, 2003.
[15] J. A. Comenius, K. Schaller, und J. A. Comenius, *Große Didaktik*, 10., Aufl. Stuttgart: Klett-Cotta, 2007.
[16] E. Kiel, Hrsg., *Unterricht sehen, analysieren, gestalten*, 3., Überarbeitete Auflage. Bad Heilbrunn: Verlag Julius Klinkhardt, 2018.
[17] J. Sorva, *Visual program simulation in introductory programming education*. Espoo: Aalto Univ. School of Science, 2012.
[18] S. Eberhart und M. Hinderer, *Stimm- und Sprechtraining für den Unterricht: ein Übungsbuch*, 3., Überarbeitete Auflage. Stuttgart: Ferdinand Schöningh, 2020.
[19] Deutscher Bundesverband der akademischen Sprachtherapeuten, Hrsg., „Logos - Die Fachzeitschrift für akademische Sprachtherapie und Logopädie“, Juni 2016.
[20] K.-H. Stier, „‚Prävention von Stimmstörungen bei Lehramtsstudentinnen, Referendarinnen und Lehrerinnen nach der Akzentmethode‘ - eine Evaluationsstudie.“, pädagogische Hochschule Ludwigsburg, Blaubeuren, 2013. [Online]. Verfügbar unter: https://d-nb.info/1117812499/34
[21] P. Flume und M. Schmettkamp, *Präsentieren*, 1. Aufl. Freiburg München: Haufe-Gruppe, 2015.
[22] K. Piepenbrink, *Humor und Schule: eine Einführung in die Soziologie des Humors für den Unterricht*. Hamburg: Diplomica-Verl, 2013.
[23] T. Sambrani, S. Mani, M. Almeida, und E. Jakubovski, „The Effect of Humour on Learning in an Educational Setting“. International Journal of Education and Psychological Research, 3. September 2014. [Online]. Verfügbar unter: https://www.researchgate.net/profile/Tanvi-Sambrani-2/publication/290428251_The_effect_of_humour_on_learning_in_an_educational_setting/links/5b97b59e299bf14ad4cc4ce6/The-effect-of-humour-on-learning-in-an-educational-setting.pdf
[24] A. Ziv, „Teaching and Learning with Humor: Experiment and Replication“, *J. Exp. Educ.*, Bd. 57, Nr. 1, S. 4–15, Sep. 1988, doi: 10.1080/00220973.1988.10806492.
[25] J. Nielsen, R. Budiu, und J. Nielsen, *Mobile usability: für iPhone, iPad, Android und Kindle*, 1. Aufl. Heidelberg München Landsberg Frechen Hamburg: mitp, 2013.
[26] B. Preim und R. Dachselt, *Interaktive Systeme / Preim, Bernhard. 1: Grundlagen, Graphical User Interfaces, Informationsvisualisierung*, 2. Aufl. Berlin Heidelberg: Springer, 2010. doi: 10.1007/978-3-642-05402-0.
[27] S. B. Barnes, „Douglas Carl Engelbart: developing the underlying concepts for contemporary computing“, *IEEE Ann. Hist. Comput.*, Bd. 19, Nr. 3, S. 16–26, Sep. 1997, doi: 10.1109/85.601730.
[28] M. Dahm, *Grundlagen der Mensch-Computer-Interaktion*. München: Pearson-Studium, 2006.
[29] Hasselhorn, Marcus, „Arbeitsgedächtnis und Sprachentwicklungsstörungen. Bestandsaufnahme und Möglichkeiten der Diagnostik“, Jan. 2019, doi: 10.25656/01:16123.
[30] U. Maiter, „Präattentive und attentive Informationsverarbeitung bei Depressiven“, Universitätsklinikum Ulm, Ulm, 2012. [Online]. Verfügbar unter: https://d-nb.info/1023729067/34
[31] M. Hahn, „Die 8 Farbkontraste: Bedeutung, Wirkungen & Praxis-Beispiele aus Print- & Webdesign“. 3. Mai 2019. [Online]. Verfügbar unter: https://www.webdesign-journal.de/farbkontraste-wirkung/#Simultankontrast
[32] P. H. Schiller und C. E. Carvey, „The Hermann Grid Illusion Revisited“, *Perception*, Bd. 34, Nr. 11, S. 1375–1397, Nov. 2005, doi: 10.1068/p5447.
[33] M. Scharch, „Vorgehensmodelle in der Software-Entwicklung“. April 2016. [Online]. Verfügbar unter: https://ilias.uni-giessen.de/ilias/goto.php?target=file_183936_download&client_id=JLUG

[34] C. Steinbauer und T. Aichholzer, „Prototyping“. 7. Mai 2003. [Online]. Verfügbar unter: http://webcache.googleusercontent.com/search?q=cache:ghqBbf27DE4J:wwwu.edu.uni-klu.ac.at/taichhol/_neu_/%3Fshow%3Dload%26id%3D56+&cd=1&hl=de&ct=clnk&gl=at
[35] A. Biørn-Hansen, C. Rieger, T.-M. Grønli, T. A. Majchrzak, und G. Ghinea, „An empirical investigation of performance overhead in cross-platform mobile development frameworks“, *Empir. Softw. Eng.*, Bd. 25, Nr. 4, S. 2997–3040, Juli 2020, doi: 10.1007/s10664-020-09827-6.
[36] D. Meiller, *Moderne App-Entwicklung mit Dart und Flutter: Eine umfassende Einführung*. Berlin ; Boston: De Gruyter Oldenbourg, 2020.
[37] E. Modrow und K. Strecker, *Didaktik der Informatik*. Boston: De Gruyter, 2016.
[38] J. Wiechmann und S. Wildhirt, Hrsg., *Zwölf Unterrichtsmethoden: Vielfalt für die Praxis*, 6., Vollständig überarbeitete Auflage. Weinheim Basel: Beltz, 2016.
[39] C. Kopf, „AKTIVIERUNGSAUFFORDERUNGEN IM WIRTSCHAFTSBERUFLICHEN UNTERRICHT – IDENTIFIZIERUNG INSTRUKTIONALER MUSTER“, Universität Mannheim, Mannheim, 2017. [Online]. Verfügbar unter: https://www.mangold-international.com/_Resources/Persistent/c/1/d/1/c1d1e1c3b76ea0f142f729a8dad412ac3da98e48/Dissertation_Claudia_Kopf.pdf
[40] Flutter Foundation, Hrsg., „Flutter documentation“. 8. September 2021. [Online]. Verfügbar unter: https://flutter.dev/docs
[41] F. Krummenauer, „Boxplots – die flexible Alternative zum ‚Antennen-Bildchen‘“. Deutscher Ärzte-Verlag Köln, 2007.
[42] I. Rahms, „Tutorienprogramm“. 15. Oktober 2021. [Online]. Verfügbar unter: https://www.studienbuero-bildungswissenschaften.zfl.uni-mainz.de/tutorienprogramm/
[43] GC University Faisalabad, M. N. Kalyar, B. Ahmad, GC University Faisalabad, H. Kalyar, und MPhil Education, „Does Teacher Motivation Lead to Student Motivation? The Mediating Role of Teaching Behavior“, *Vopr. Obraz. Educ. Stud. Mosc.*, Nr. 3, S. 91–119, 2018, doi: 10.17323/1814-9545-2018-3-91-119.
[44] B. Rammstedt, „Zur Bestimmung der Güte von Multi-Item-Skalen: Eine Einfürhung“. Zetnrum für Umfragen, Methoden und Analysen, 2004.