



Fachbereich
Elektrotechnik und Informatik



FACH
HOCHSCHULE
LÜBECK

University of Applied Sciences

Bachelorarbeit

Eine gamifizierte Howto-App für Bachelorarbeiten

vorgelegt von

Tim-Pascal Lau

Ausgabedatum: 24. Mai 2018

Abgabedatum: 24. August 2018

Aufgabenstellung

Für Studierende im letzten Semester eines Bachelorstudiengangs umfasst **DIE** wesentliche Prüfungsleistung das Verfassen einer Bachelorarbeit. Die Auseinandersetzung mit komplexen Problemstellungen stellt jedoch erfahrungsgemäß für viele Studierende eine große Herausforderung dar, welche sich aus dem erstmaligen Zusammenspiel von selbständigem und eigenverantwortlichem Arbeiten sowie Problemlösen mittels erworbener Fach- und Methodenkenntnisse über einen längeren (dreimonatigen) Zeitraum ergibt.

Im Verlauf des Studiums sollten Studierende folgendes Wissen und folgende Fähigkeiten erworben haben und zielgerichtet anwenden können:

- Studiengang-spezifisches Grundlagenwissen
- Wissensansammlung über fachspezifische Methoden und deren Eigenschaften
- Die Fähigkeit, komplexe Probleme zu erkennen, zu strukturieren und systematisch mittels geeigneter Methoden zu bearbeiten

Es kommt im Kontext von Bachelorarbeiten dennoch oftmals zu Schwierigkeiten, das erworbene Wissen und Fähigkeiten zielgerichtet auf reale Probleme anzuwenden und deren Ergebnisse zusammenhängend zu dokumentieren.

Lösungsansatz

Ein möglicher Lösungsansatz wäre es, Studierenden eine Software bereitzustellen, welche unterstützend und wegweisend bei dem systematischen Vorgehen für komplexe Problemstellungen fungieren könnte. Hierbei soll es nicht darum gehen, dem Studierenden die eigentliche Arbeit abzunehmen, sondern vielmehr darum, Studierende hinsichtlich Vorgehen und Methodenauswahl zielgerichtet zu unterstützen.

Im Rahmen dieser Bachelorarbeit soll hierzu eine mobile Applikation entwickelt werden, die den Studierenden während der Dauer seiner Bachelorarbeit kontinuierlich “begleitet“. Dabei sollen Gamificationansätze realisiert werden, die motivierend im Rahmen einer Bachelorarbeit wirken sollen. Dies soll beispielhaft für den Studiengang Informatik/Softwareentwicklung erfolgen. Eine Erweiterbarkeit für andere Studiengänge ist hierbei jedoch von Anfang an konzeptionell vorzusehen.

Primäre Funktion der Software ist Studierenden bei den folgenden Aufgaben begleitend zu unterstützen und fortwährend zu motivieren “am Ball zu bleiben“:

- Brainstorming (zur Unterstützung der Ideenfindung für Bachelorarbeiten)
- Recherche und Literaturverwaltung
- Gliederung (unterschiedlicher Kategorien von Bachelorarbeiten, zum Beispiel mittels bewehrter Templates)
- Zeitplanung/Fortschrittsverfolgung sowie Erinnerungs- und Benachrichtigungsfunktion
- Problem-orientierte Anforderungsanalyse und deren Dokumentation
- Problem-orientierte Methoden- und Tool-/Framework-Selektion und deren Dokumentation
- Methoden-spezifische Aufbereitung von Ergebnissen
- Problem-orientierte Nachweisführung und deren Dokumentation

Die App soll mit Flutter für Android und iOS entwickelt werden. Dabei soll erhoben werden, inwiefern sich Flutter für die Entwicklung solcher Apps eignet (Lessons Learned).

Teilaufgaben

- Detaillierte Anforderungsanalyse oben angegebener Funktionen. Hierbei sind Studenten und Professoren des Studiengangs Informatik/Softwaretechnik geeignet einzubeziehen und relevante Literatur (insbesondere zu Gamification und Methoden der Informatik und des Softwareengineering) zu berücksichtigen.
- Architekturentwurf der Anwendung (Erweiterbarkeit für andere Studiengänge ist konzeptionell vorzusehen)
- Implementierung der Anwendung
- Die Funktionsfähigkeit der App soll mittels Softwaretests geeignet nachgewiesen werden.
- Die Nutzbarkeit der App soll systematisch evaluiert werden. Hierbei sind Studenten und Professoren des Studiengangs Informatik/Softwaretechnik geeignet einzubeziehen.
- Dokumentation der oben angegebenen Schritte inklusive Bewertung der Nutzbarkeit des Frameworks Flutter für solche Arten von Apps.

Es wird empfohlen die Dokumentation von Anfang an begleitend zur Arbeit zu erstellen und sich an einem Dokumentations-Template - wie bspw. diesem - zu orientieren.

Erklärung zur Bachelorarbeit

Ich versichere, dass ich die Arbeit selbständig, ohne fremde Hilfe verfasst habe.

Bei der Abfassung der Arbeit sind nur die angegebenen Quellen benutzt worden.
Wörtlich oder dem Sinne nach entnommene Stellen sind als solche gekennzeichnet.

Ich bin damit einverstanden, dass meine Arbeit veröffentlicht wird, insbesondere dass die Arbeit Dritten zur Einsichtnahme vorgelegt oder Kopien der Arbeit zur Weitergabe an Dritte angefertigt werden.

(Datum)

Unterschrift

Danksagung

Hiermit möchte ich mich bei Herrn Prof. Dr. rer. nat. Dipl.-Inform. Nane Kratzke für die Betreuung meiner Bachelorarbeit und das präzise Feedback zu Arbeitsergebnissen sowie die grundsätzlich schnelle Beantwortung meiner angefallenen Fragen bedanken.

Weiterhin möchte ich mich bei Prof. Dr.-Ing. Stefan Krause bedanken, da er sich als Zweitprüfer für meine Bachelorarbeit bereit erklärt hat.

Einen weiteren persönlichen Dank möchte ich meiner Partnerin Melina Meyer aussprechen, die mir im Verlauf des Projekts sehr viel Verständnis sowie persönliche Unterstützung entgegengebracht und mich beim Korrekturlesen der Dokumentation unterstützt hat.

Einen abschließenden Dank möchte ich den Professoren und Studierenden der Fachhochschule Lübeck widmen, die durch die Beteiligung an zeitaufwändigen Interviews und weiteren Befragungen einen essenziellen Beitrag für diese Arbeit geleistet haben.

Inhaltsverzeichnis

1. Einleitung	1
1.1. Lösungsansatz	1
1.2. Zielstellung	1
1.3. Vorgehensweise	2
1.4. Voruntersuchung zur Ermittlung der Bedarfssituation	3
1.5. Aufbau der Dokumentation	5
2. Beschreibung der Grundlagen	6
2.1. Gamification	6
2.1.1. Was ist Gamification?	6
2.1.2. Bekannte Spiel-Design-Elemente	7
2.1.3. Nutzergruppen nach Bartle	9
2.1.4. Bedeutung für die Arbeit	10
2.2. Vorstellung des Frameworks Flutter	11
2.2.1. Was ist Flutter?	11
2.2.2. Plattformübergreifende Entwicklung von mobilen Applikationen	12
2.2.3. Flutter Widgets	13
2.2.4. Flutter UI Rendering	14
3. Untersuchung des Problemereiches	15
3.1. Hypothese	15
3.2. Identifikation der Interessengruppen	16
3.3. Wahl der Analysestrategie	17
3.3.1. Einbezug der Professoren	18
3.3.2. Einbezug der Studierenden	18
3.4. Ergebnisse der Professoren	19
3.4.1. Art der Arbeit	20
3.4.2. Erwartungen an den Bacheloranden	22
3.4.3. Häufig auftretende Probleme	24
3.4.4. Die Applikation – Wünsche und Anregungen	25
3.4.5. Die Applikation - Chancen und Risiken	26
3.5. Ergebnisse der Studierenden	28
3.5.1. Erwartungen an den Betreuer	28
3.5.2. Die größten Probleme der Studierenden	29
3.5.3. Wunsch nach besserer Vorbereitung auf die Bachelorarbeit	30
3.5.4. Einbezug der Studierenden in die Gestaltung der Applikation	31
3.6. Untersuchung der Ergebnisse	32
3.6.1. Identifikation des Problemereichs	32
3.6.2. Auswahl der Motivationsstrategie	33
3.6.3. Übersicht der abgeleiteten Produktfunktionen und Anforderungen	35

3.6.4.	Übersicht der Risiken	37
3.6.5.	Übersicht der Chancen	38
4.	Konzeptvorstellung der Applikation	39
4.1.	Beschreibung der Software	39
4.1.1.	Meilensteinverwaltung	40
4.1.2.	Bachelorarbeit Guide	41
4.1.3.	Herausforderungen	43
4.1.4.	Achievements	44
4.1.5.	Sonstige Softwareinhalte	45
4.1.6.	Dashboard	46
4.1.7.	Übersicht der Produktfunktionen	48
4.1.8.	Umgang mit Risiken	49
4.2.	Beschreibung der Gamification-Elemente	50
5.	Architektur und prototypische Implementierungsdetails	52
5.1.	Was ist Redux?	52
5.1.1.	Actions	53
5.1.2.	Reducers	53
5.1.3.	Store	53
5.1.4.	Flutter Redux Library	54
5.2.	Architekturbeschreibung	55
5.2.1.	Übersicht des Datenflusses	55
5.2.2.	Verwendet externe Bibliotheken	56
5.2.3.	Architekturübersicht der Model-Komponenten	57
5.3.	Beispielhafter Ablauf einer Nutzeraktion	59
6.	Evaluation	64
6.1.	Ergebnisse der Softwaretests	64
6.2.	Ergebnisse der Nutzerevaluation	65
6.2.1.	Zusammenfassung der Testdurchführung mit den Studierenden	66
6.2.2.	Zusammenfassung der Testdurchführung mit den Professoren	68
6.2.3.	Ergebnis der Nutzerevaluation	70
6.3.	Bewertung des Frameworks Flutter zur Entwicklung von mobilen Applikationen	71
6.3.1.	Einführung und Dokumentation	71
6.3.2.	Verwendete Programmiersprache	71
6.3.3.	Handhabung und Anwendung	71
6.3.4.	UI-Performance	72
6.3.5.	Support und Community	72
6.3.6.	Framework-spezifische Features	73
6.3.7.	Zusammenfassung der Bewertung	73
7.	Zusammenfassung	74
7.1.	Die wichtigsten Erkenntnisse der Arbeit	74
7.2.	Schwierigkeiten bei der Bearbeitung	75
7.3.	Erfüllung der Aufgabenstellung	75
7.3.1.	Unterstützung bei Bearbeitung der Bachelorarbeit	75

7.3.2.	Unterstützung bei Zeitplanung	76
7.3.3.	Motivation der Studierenden	76
7.4.	Fortführung des Projekts	77
7.4.1.	Ausblick auf weitere Untersuchungen	77
7.4.2.	Ausblick auf Softwarefunktionen	77
7.5.	Fazit	78
Abbildungsverzeichnis		79
Tabellenverzeichnis		80
Listings		81
A. Anhang		83
A.1.	Befragung der Beteiligten	84
A.1.1.	Transkripte der Interviews mit den Professoren (Interview I)	84
A.1.2.	Transkripte der Interviews mit den Professoren (Interview II)	91
A.1.3.	Transkripte der Interviews mit den Professoren (Interview III)	95
A.1.4.	Transkripte der Interviews mit den Professoren (Interview IV)	102
A.1.5.	Transkripte der Interviews mit den Professoren (Interview V)	106
A.1.6.	Notizen von Interviews mit Professoren (Interview VI)	113
A.1.7.	E-Mail-Befragung der Professoren I	115
A.1.8.	Zusammenfassung der schriftlichen Befragung der Studierenden	116
A.1.9.	Ergebnisse der Online-Umfrage für die Studierenden	122
A.2.	Übersicht der Herausforderungen und Achievements	133
A.3.	Oberflächenprototyp	135
A.4.	Detaillierte Beschreibung der Model Komponenten	138
A.4.1.	AppState-Klasse (@immutable)	138
A.4.2.	Milestone-Klasse (@immutable)	140
A.4.3.	Task-Klasse (@immutable)	141
A.4.4.	Challenge-Klasse (@immutable)	141
A.4.5.	Achievement-Klasse (@immutable)	141
A.4.6.	Property-Klasse (@immutable)	142
A.4.7.	Content-Klasse (@immutable)	143
A.4.8.	AppContentLoader-Klasse	144
A.4.9.	JSONAppContentFile-Klasse	144
A.4.10.	Achievement-Notification Funktionen	144
A.5.	Dokumente der Evaluation	145
A.5.1.	Aufgabenkatalog	145
A.5.2.	Ergebnisse der Professoren	147
A.5.3.	Ergebnisse der Studierende	149
Literatur		157

1. Einleitung

Für Studierende im letzten Semester eines Bachelorstudiengangs stellt das Verfassen einer Bachelorarbeit die wesentliche Prüfungsleistung dar. Die Auseinandersetzung mit komplexen Problemstellungen stellt jedoch erfahrungsgemäß für viele Studierende eine große Herausforderung dar, die sich aus dem erstmaligen Zusammenspiel von selbständigem und eigenverantwortlichem Arbeiten sowie Problemlösen mittels erworbener Fach- und Methodenkenntnisse über einen längeren (dreimonatigen) Zeitraum ergibt.

Im Verlauf des Studiums sollten Studierende folgendes Wissen und folgende Fähigkeiten erworben haben und zielgerichtet anwenden können:

- Studiengang-spezifisches Grundlagenwissen
- Wissensansammlung über fachspezifische Methoden und deren Eigenschaften
- Die Fähigkeit, komplexe Probleme zu erkennen, zu strukturieren und systematisch mittels geeigneter Methoden zu bearbeiten

Es kommt im Kontext von Bachelorarbeiten dennoch oftmals zu Schwierigkeiten, das erworbene Wissen und die Fähigkeiten zielgerichtet auf reale Probleme anzuwenden und deren Ergebnisse zusammenhängend zu dokumentieren.

1.1. Lösungsansatz

Der in dieser Arbeit verfolgte Lösungsansatz ist die Entwicklung einer mobilen Applikation unter Verwendung des hierfür ausgelegten Frameworks Flutter zur Unterstützung von Studierenden beim Bearbeiten der Bachelorarbeit. Dieser Ansatz wird präzisiert durch verschiedene prozessbegleitende Features für Zeit- und Aufgabenmanagement sowie dem Bereitstellen eines zentralen Informationskanals für themenbezogene Inhalte. Als hervorzuhebende Besonderheit gilt hierbei der Einsatz von Gamification zur Steigerung der Motivation bei Bearbeitung der Bachelorarbeit sowie die damit erwartete Förderung des kontinuierlichen Projektfortschritts.

1.2. Zielstellung

Mit dem beschriebenen Ansatz soll es Studierenden ermöglicht werden, ihr gelerntes Wissen und die durch das Studium erlangten Fähigkeiten durch Fokussieren wichtiger Aufgaben und Zusammenhänge auf ihr eigenes Bachelorprojekt zu übertragen und somit einen motivierenden und weiterhin fordernden Rahmen zu schaffen, um so ihr Bachelorprojekt erfolgreich abzuschließen.

1.3. Vorgehensweise

Da die Entwicklung einer solchen Applikation abseits der vorstellbaren Chancen auch mit verschiedenen Abhängigkeiten und möglichen Risiken einhergeht und unter anderem bezüglich der Integration eines Gamification-Systems einen engen zielgruppenspezifischen Umgang erfordert, wird bei Durchführung des Projekts eine hierfür entwickelte Strategie zur Einbindung der beteiligten Personen verfolgt.

Hierbei erfüllt eine durchgeführte Online-Umfrage, welche im Anschluss zu diesem Kapitel kurz beschrieben wird, einen wichtigen Beitrag zur Einschätzung der allgemeinen Situation. Die hierbei ermittelten Schwierigkeiten, Befürchtungen oder Ängste der Studierenden bilden somit eine wichtige Grundlage zur weiteren Untersuchung.

Die darauf folgende detaillierte Untersuchung des Problembereichs und die Einbindung der Beteiligten findet durch organisierte Interviews mit ausgewählten Professoren sowie eine kontinuierlich fortgeführte schriftliche Befragungen von Studierenden des Studiengangs Informatik/Softwareentwicklung der Fachhochschule Lübeck statt.

Zur Beurteilung der entwickelten Software hinsichtlich der Nutzbarkeit, findet projektabschließend eine erneute persönliche Einbindung von Professoren und Studierenden in Form von organisierten Tests statt.

1.4. Voruntersuchung zur Ermittlung der Bedarfssituation

Im Vorfeld zur Entwicklung der Applikation wurde eine Online-Umfrage an verschiedenen Hochschulen durchgeführt (siehe Anhang A.1.9), welche als Voruntersuchung des Problemfeldes dient und somit einen ersten Eindruck über den Bedarf an einer Software zur Unterstützung der Studierenden bei der Bearbeitung der Bachelorarbeit liefert.

Insgesamt haben 44 Studierende unterschiedlicher Hochschulen an der Umfrage teilgenommen, wobei sich diese in weitere drei Gruppen aufteilen lassen. Diese Unterteilung erfolgt nach den unterschiedlichen Phasen, in denen sich die jeweiligen teilnehmenden Studierenden zur Zeit der Umfrage befinden.

Im Folgenden werden die wichtigsten Erkenntnisse der zusammengefassten Umfrage (siehe Abbildung 1.1) beschrieben:

- **Vor Beginn der Bachelorarbeit (Ab 4. Semester)**

Teilnehmerzahl: 24

Die Ergebnisse der Studierenden, die sich noch vor der Bearbeitung der Bachelorarbeit befinden, zeigen bezüglich der Kenntnisse über Ablauf der Arbeit, Erwartungen an den Studierenden sowie deren Einschätzung zu Selbst- und Zeitmanagement ein relativ gleichmäßig verteiltes Bild der Situation auf.

Hierbei fällt auf, dass ein nicht geringer Teil explizit angibt, Schwierigkeiten mit den genannten Aspekten zu haben.

- **Während der Bachelorarbeit**

Teilnehmerzahl: 14

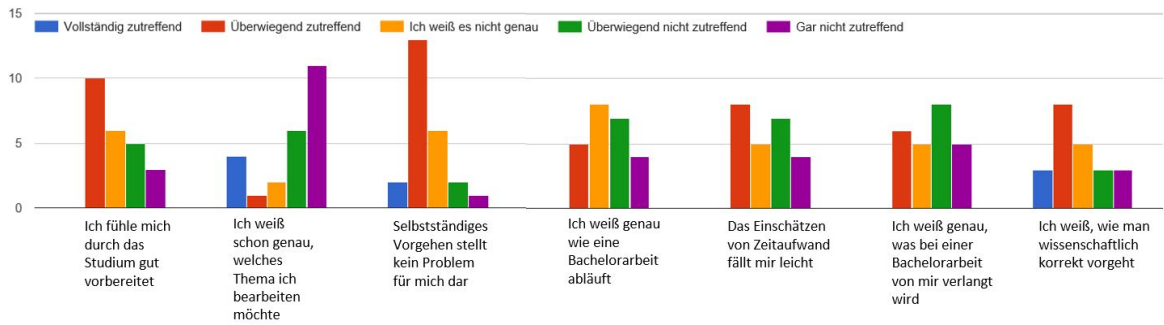
Bei Betrachtung der Ergebnisse der Studierenden, die sich in der Phase während der Bachelorarbeit befinden, ist abzulesen, dass es im Kontext von Bachelorarbeiten unter einem nicht geringen Teil der Studierenden zu Schwierigkeiten kommt.

Hierbei gelten besonders das Zeitmanagement und das wissenschaftliche Arbeiten durchaus als problematisch. Weiterhin sagt über die Hälfte der Studierenden aus, dass sie mit der gesamten Bearbeitung der Arbeit überfordert sind und ein weiterer Teil, dass sie sich nicht ausreichend mit den Formalien einer Bachelorarbeit auskennen. Viele der Studierende fühlen sich unsicher gegenüber dem, was von ihnen erwartet wird, schätzen sich im Allgemeinen jedoch durch das Studium als gut vorbereitet ein.

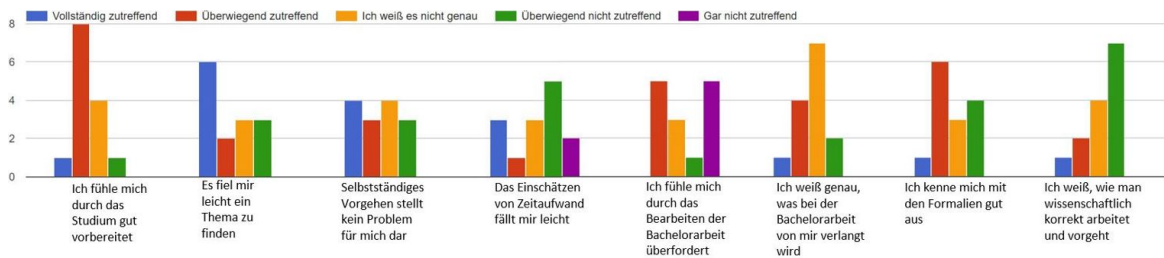
- **Nach Abschluss der Bachelorarbeit**

Teilnehmerzahl: 6

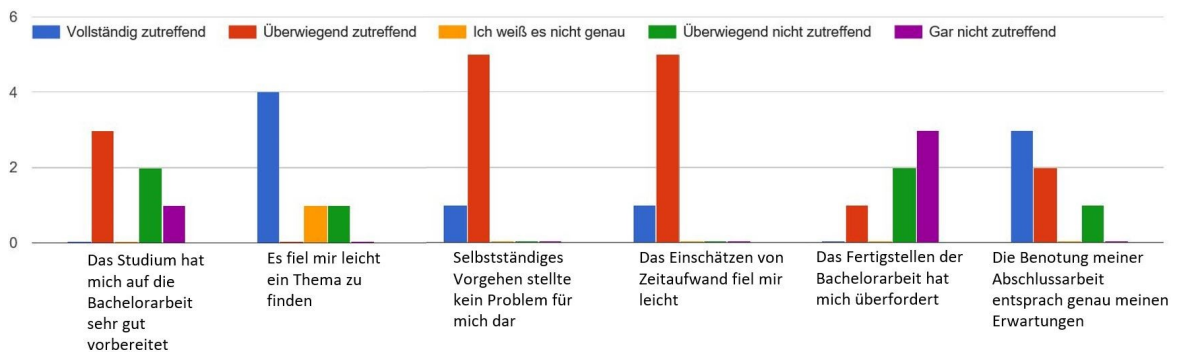
Auffällig einstimmig verhalten sich die Ergebnisse der Studierenden, die sich bereits nach Abschluss der Bachelorarbeit befinden. Das selbstständige Arbeiten sowie das Einschätzen von Zeitaufwand haben nach Aussage der Studierenden offensichtlich wenig bis keine Schwierigkeiten dargestellt. Weiterhin geben die Befragten an, sich bei der Fertigstellung der Bachelorarbeit nicht überfordert gefühlt zu haben. Ein wichtiges Detail ist jedoch, dass über die Hälfte der Befragten angeben, sich durch das Studium nicht ausreichend auf die Bachelorarbeit vorbereitet gefühlt zu haben.



(a) Vor Beginn der Bachelorarbeit



(b) Während der Bachelorarbeit



(c) Nach Abschluss der Bachelorarbeit

Abbildung 1.1.: Online-Umfrage: Zusammenfassung der Umfrageergebnisse

Das Gesamtbild der Umfrage zeigt die verschiedenen Schwierigkeiten, mit denen Studierende im Laufe der Bachelorarbeit zu tun haben und stellt diese als wichtige Untersuchungselemente zur Analyse des Problembereichs heraus.

1.5. Aufbau der Dokumentation

In der folgenden Dokumentation sind verschiedene aufeinander aufbauende Prozesse, Teilschritte und Ergebnisse der Softwareentwicklung dokumentiert.

Beschreibung der Grundlagen

Wie in der Einleitung bereits beschrieben, werden im Rahmen dieses Projekts verschiedene Inhalte aus dem Gebiet der Gamification bezüglich Software-Anwendungen behandelt. Um einen Eindruck über die Umsetzung und Bedeutung von gamifizierten Inhalten in modernen Software-Anwendungen zu erhalten, werden im Anschluss in **Kapitel 2** die für diese Arbeit benötigten Grundlagen beschrieben. Weiterhin wird hier auch das im Rahmen dieses Projekts verwendete Framework Flutter hinsichtlich dessen Funktionsweise und technischen Hintergründen vorgestellt.

Untersuchung des Problemfeldes

Bei der Durchführung dieses Projekts haben sich sowohl Professoren als auch Studierende der Fachhochschule Lübeck bereit erklärt, an Interviews und weiteren Einbindungen in das Projekt teilzunehmen und somit einen Beitrag dazu zu leisten, das zu untersuchende Problemfeld aufzudecken. Die aus der gewählten Befragungsstrategie resultierenden Ergebnisse werden in **Kapitel 3** diskutiert und durch das Ableiten von präzisen Lösungsmöglichkeiten beschrieben.

Entwicklung der Software

Basierend auf den abgeleiteten Anforderungen werden im folgenden Verlauf der Dokumentation die verschiedenen Abstraktionsebenen der Software beschrieben. In **Kapitel 4** wird anhand der Oberfläche der Applikation das Gesamtkonzept unter Einbezug von Hintergründen der gewählten Lösungswege und Designentscheidungen beschrieben, wohingegen in **Kapitel 5** die Beschreibung der wichtigsten Details zur technischen Umsetzung ausgeführt wird.

Evaluation

Die Evaluation der umgesetzten Softwarelösung wird unter der Durchführung von Softwaretests und dem erneuten Einbeziehen von Professoren und Studierenden in **Kapitel 6** behandelt. Zusätzlich wird in diesem Kapitel auch die abschließende Bewertung des Frameworks Flutter durchgeführt und diskutiert.

Zusammenfassung der Ergebnisse

Die im Laufe des Projekts erreichten Ergebnisse werden in **Kapitel 7** zusammengefasst und dienen unter Betrachtung des zukünftigen Projektverlaufs als Grundlage für weitere dort beschriebene potentielle Ideen und Erweiterungsmöglichkeiten der Software.

2. Beschreibung der Grundlagen

Im folgenden Kapitel werden die aus der Aufgabenstellung hervorgehenden und für die Nachvollziehbarkeit dieser Arbeit wichtigen Grundlagen beschrieben. Hierbei wird vor allem das bereits durch die Einleitung motivierte Themengebiet der Gamification beschrieben und im Anschluss das im Rahmen zur Entwicklung der mobilen Applikation gewählte Framework Flutter vorgestellt, um dessen Funktionsweise sowie Besonderheiten gegenüber anderen Alternativen zu erläutern.

2.1. Gamification

in diesem Kapitel werden die für diese Arbeit relevanten Informationen bezüglich Gamification vermittelt. Hierbei wird erst auf die grundlegende Bedeutung sowie deren Abgrenzung gegenüber anderen Ansätzen eingegangen. Im weiteren Verlauf werden dann tiefgreifendere Informationen zu den verschiedenen Benutzerklassen und den jeweils kompatiblen Spiel-Design-Elementen beschrieben.

2.1.1. Was ist Gamification?

Im Kontext dieser Ausarbeitung wird sich auf die in [11] genannte Definition für den Begriff “Gamification“ bezogen:

Gamification is the use of elements of game design in non-game contexts.

Die Definition beschreibt Gamification als das Einsetzen spieltypischer Elemente in einem spielfremden Kontext. Somit lässt sich ein Kontext bereits als “gamifiziert“ beschreiben, sobald Spiel-Design-Elemente in den spezifischen Kontext integriert worden sind, weshalb besonders im Rahmen der Unternehmenseinführung von Gamification-Ansätzen zwischen **Enterprise Gamification** und **Serious Games** unterschieden wird. [38, vgl. Kapitel 1.1 Gamification: Definition und Abgrenzung, Seite 4]

Enterprise Gamification bezeichnet das konkrete Einbinden von Gamification-Elementen in Arbeits- und/oder Lernprozessen mit dem expliziten Ziel der Motivationssteigerung seitens der Anwender.

Gegenüber diesem Ansatz hat sich außerdem der Begriff **Serious Games** etabliert, welcher das explizite Einsetzen von Spielen in einem ernsten Kontext vorsieht und die Unterhaltung nicht im Fokus der Anwendung steht, sondern die Wertschöpfung von Wissen und Realwelterfahrungen.

2.1.2. Bekannte Spiel-Design-Elemente

Im folgenden Kapitel werden die wichtigsten interaktiven Spiel-Design-Elemente, welche in Tabelle 2.1 genannt werden, vorgestellt und beschrieben.

Der Einsatz von Gamification-Elementen basiert auf der Annahme, dass diese die grundsätzlichen Bedürfnisse der Menschen ansprechen (vgl. [38, Kapitel 1.1 Enterprise Gamification - Vorgehen und Anwendung, Seite 5]).

Spiel-Mechanik	Spiel-Dynamik	Motiv
Dokumentation von Verhaltensweisen	Exploration	Wissbegierde
Punktesysteme, Badges, Trophäen	Sammeln	Leistung
Ranglisten	Wettbewerb	Soziale Anerkennung
Ränge, Levels, Reputationspunkte	Statuserwerb	
Gruppenaufgaben	Zusammenarbeit	Sozialer Austausch
Zeitdruck, Aufgaben, Missionen	Herausforderung	Kognitive Stimulation
Avatare, virtuelle Welten, virtueller Handel	Entwickeln/Organisieren	Selbstbestimmung

Tabelle 2.1.: Übersicht der Spiel-Design-Elemente [6]

Die folgenden aufgeführten Informationen wurden größtenteils aus den Werken [38, Kapitel 2.2.2 Analyse einzelner Spiel-Design-Elemente] und [44, Kapitel 4 The Gamification Toolkit Game Elements] entnommen, bei Abweichungen werden die konkreten Literaturwerke aus Gründen des Leseflusses explizit angegeben.

Dokumentation von Verhaltensweisen

Durch die ständige Dokumentation der eigenen Verhaltensweisen werden persönliche Fortschritte für den Anwender sichtbar gemacht. Diese Fortschritte können ein Gefühl von hoher Leistungsfähigkeit auslösen, da auf diese Weise ein Vergleich zu den bisherigen erreichten Leistungen ermöglicht wird und somit bei Verbesserung der eigenen Leistung der persönliche Fortschritt verdeutlicht werden kann.

Punkte

Punktesysteme sind in ihrer Einfachheit eine weitverbreitete Form der Spiel-Design-Elemente, welche hauptsächlich eine Feedback-Funktion erfüllen. Diese Feedback-Funktion kann dem Spieler durch diverse weitere Möglichkeiten, wie zum Beispiel durch das Aufsteigen eines Levels, unterschiedlich stark verdeutlicht werden, weshalb Punkte häufig als Grundbestandteil von Gamification-Anwendungen auftreten. Weiterhin symbolisieren Punkte dem Spieler seinen derzeitigen Spielstand und bieten bei der Kombination mit Bestenlisten eine Vergleichsmöglichkeit zu anderen Spielern. In dieser Kombination können Punktesysteme also je nach Zielstellung auch Gewinner einer Herausforderung identifizieren.

In den zwei in [38] behandelten empirischen Untersuchungen [26] und [27] wurde nachgewiesen, dass der Einsatz von Punktesystemen durchaus einen leistungsfördernden Effekt auf die Spieler haben kann.

Abzeichen

Die Abzeichen, welche sich in der Regel durch erreichbare Achievements, Badges und/oder Trophäen innerhalb von Gamification-Anwendungen etabliert haben, dienen einerseits einer zielsetzenden und darüber hinaus auch einer nicht-kontrollierenden Feedback-Funktion.

Zu erwähnen gilt, dass Badges und Trophäen nicht mit Achievements gleichzusetzen sind, sondern lediglich die visuelle Repräsentation eines Achievements darstellen.

Durch den Einsatz solcher Spiel-Design-Elemente soll der Benutzer sinnvoll gefordert werden und sich im Optimalfall durch das somit klar gesteckte Ziel motiviert fühlen.

Weiterhin lassen sich bei dem Einsatz von Achievements unterschiedliche strategische Ansätze identifizieren, die sich von der bereits genannten zielsetzenden Strategie abheben.

Beispielsweise kann bewusst auf die Angabe des Ziels eines Achievements verzichtet werden, um je nach Kontext die Neugierde des Benutzers zu fördern und den Effekt des Erreichens umso größer ausfallen zu lassen, da diese Achievements durch Ausführen verschiedener Tätigkeiten überraschend erreicht werden. Dies spricht aufgrund des erforschenden Charakters vor allem die Benutzerklasse **Explorer** an (siehe Kapitel 2.1.3).

Auf diese Weise kann auch eine stärkere Bindung einer Community entstehen, da der Austausch zwischen Benutzern, die bereits wissen wie ein bestimmtes Achievement erreicht wird und denjenigen, die es noch nicht erreicht haben, die Kommunikation untereinander fördern kann.

Ranglisten

Die sogenannten Rang- oder auch Bestenlisten dienen der Identifikation und Abgrenzung von unterschiedlichen Benutzern, die im Kontext der Gamification-Anwendung unterschiedliche Erfolge, häufig in Form von Punkten, erreicht haben.

Zentrale Absicht solcher Ranglisten ist das Darstellen der besten Spieler, was je nach Kontext für einen hohen kompetitiven Anwendungsanteil spricht.

Im Rahmen von Ranglisten-Spiel-Elementen können weiterhin verschiedene Kategorien identifiziert werden, welche sich auf unterschiedliche Weise durch verschiedene Vor- und Nachteile des jeweiligen Zielkontexts äußern.

Während individuelle Ranglisten die Leistung des Einzelnen aufzeigen und somit versuchen das Individuum von den restlichen Spielern abzugrenzen, ermöglichen team-basierte Ranglisten das kooperative Vernetzen unterschiedlicher Personen und Identifizieren ganzer Team-Leistungen.

Ränge, Level und Reputationspunkte

Wie in [37, Kapitel 4.5 Reputations, Ranks and Levels, Seite 75] beschrieben wird, können Reputationsmöglichkeiten für Benutzer ein wichtiges Identifikationsmittel sein. Durch die Möglichkeit bestimmte Ränge oder Level zu erreichen, sind Spieler besonders im Rahmen von Multiplayer-Anwendungen dazu angehalten, ihr Ansehen gegenüber anderen Spielern zu steigern oder zu festigen und somit den Status eines Veteranen zu repräsentieren.

Diese Art von Spiel-Elementen ermöglicht das Ausstrahlen von Kompetenzen, Talenten sowie Erfahrungen und somit den Platz, den der jeweilige Spieler in der Spielwelt einnimmt.

Avatare, virtuelle Welten, virtueller Handel

Durch die Möglichkeit der Erstellung eines persönlichen Avatars werden dem Spieler je nach Ausprägung hoch oder gering ausfallende Individualisierungs- und somit auch Identifikationsmöglichkeiten angeboten.

Ein wichtiger Aspekt, welcher je nach Anwendungsfeld unterschiedlich umgesetzt wird, ist das Interagieren mit der virtuellen Welt, in der sich der Avatar befindet. Hierbei kann es zur Interaktion mit anderen Spieler-Avataren oder mit definierten Teilen der virtuellen Welt kommen. Die virtuelle Welt kann hierbei beispielsweise Möglichkeiten bieten, sich frei in einer dreidimensionalen Umgebung zu bewegen.

Die virtuelle Welt muss nicht zwangsweise dreidimensional ausfallen, sondern kann durch geschickte Konzipierung des Interface-Designs auch mit weniger Umgebungsfreiheit den gewünschten Effekt erzielen.

Zeitdruck, Aufgaben, Missionen

In [30, Kapitel 6.1.1 Tasks] wird beschrieben, dass Aufgaben und Missionen im Kontext von Gamification-Anwendungen sehr breit vertretene Varianten von Spiel-Design-Elementen sind. Sie können sehr gut mit anderen Elementen wie Level-Systemen kombiniert werden und lassen sich durch die hohe Individualisierbarkeit der Aufgabeninhalte im Rahmen des jeweiligen Anwendungskontexts als ziel- oder wegweisende Spiel-Elemente benutzen, die gleichzeitig in direkter Form auf das tatsächliche Anwendungsfeld Bezug nehmen.

Gruppenaufgaben

Wie in [30, Kapitel 6.1.1 Tasks] weiterhin beschrieben wird, stehen die Gruppenaufgaben, wie auch die Einzelaufgaben, für eine zu erfüllende Gesamtaufgabe. Eine Gruppenaufgabe hebt sich jedoch insofern ab, dass sie aus mehreren kleinen Aufgaben besteht, welche im Team gelöst werden sollen. Wie auch in den Team-Bestenlisten rückt bei Einsatz dieser Elemente die kooperative Ausführung von Aufgaben in den Vordergrund und soll so durch eine starke Teamidentifikation für eine erhöhte Motivation des Einzelnen sorgen.

2.1.3. Nutzergruppen nach Bartle

Nach Bartle [5] werden vier verschiedene Nutzertypen voneinander unterschieden, welche in vielen Literaturwerken zur Gamification als Grundlage oder zumindest als Orientierung referenziert werden.

Die vier verschiedenen Nutzerklassen werden im folgenden Verlauf vorgestellt und hinsichtlich ihrer Bedürfnisse erläutert:

- **Der Killer**

Die Killer legen den Fokus auf den Wettkampf. Somit beruht ihre Motivationsquelle auf stark kompetitiv ausgeprägte Ansätze. Sie fühlen sich dadurch motiviert, den Kontakt mit anderen Individuen im Rahmen eines Wettkampfes zu suchen und sich durch den eigenen Sieg gegenüber anderen Personen zu behaupten.

- **Der Socialiser**

Die Gruppe der Socialiser sind auf der Suche nach sozialer Interaktion. Ohne diese können sie der Spielwelt wenig bis keine Motivation abgewinnen. Die Spielwelt, in der sie sich bewegen, ist vor allem nur das Mittel zum Zweck der Kommunikation. Der größte Anreiz der Socialiser ist die Freundschaft und der Kontakt zu anderen Personen.

- **Der Achiever**

Die Achiever konzentrieren sich auf das Erbringen von Leistungen und das Abschließen von Herausforderungen. Diese werden so eingeschätzt, dass sie die möglichst optimale geforderte Leistung erbringen wollen und daher ihre Motivation aus dem schrittweisen Sammeln von Erfolgen und dem Erfüllen von Aufgaben ziehen.

- **Der Explorer**

Die Explorer legen den Hauptfokus auf die Welt, in der sie sich bewegen und möchten im Zusammenspiel mit dieser Spielwelt durch immer wieder neu gestellte Aufgaben und erlebte Überraschungen alles entdecken, was es zu entdecken gibt. Sie sind sehr ambitioniert und empfinden Stolz für ihr Wissen und ihre Erfahrung in der Spielwelt.

Die Explorer haben vor allem Interesse an der Spielwelt und weniger an den Personen, die sie im Laufe der Erkundung treffen.

Wie aus den Beschreibungen zu erkennen ist, entsteht bei der allgemeinen Betrachtung aller Benutzerklassen ein Spannungsfeld der Interessen, welches in Abbildung 2.1 verdeutlicht werden soll.

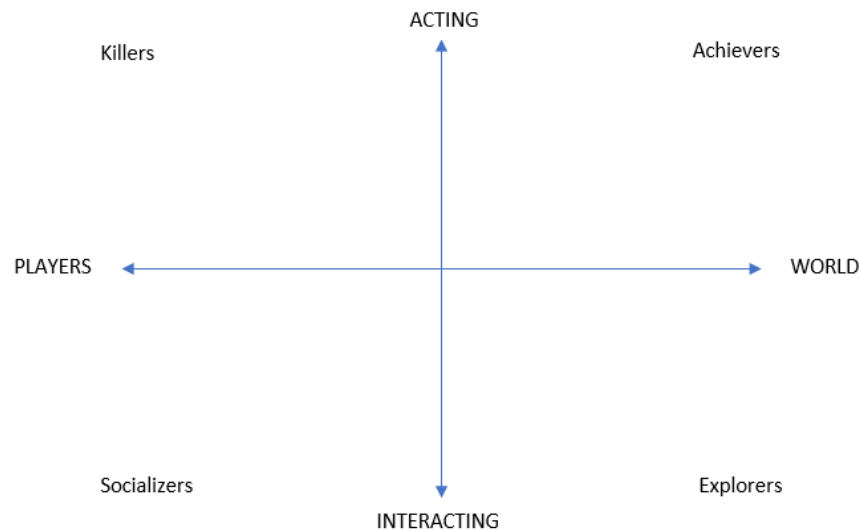


Abbildung 2.1.: Interest Graph nach Bartle [5]

Das Spannungsfeld wird weiterhin definiert durch die vier verschiedenen Interessenausprägungen der Spieler (Players), der virtuellen Welt (World), des Handelns in dieser Welt (Acting) und des Interagierens mit der virtuellen Welt (Interacting).

2.1.4. Bedeutung für die Arbeit

Die in diesem Kapitel recherchierten Ergebnisse zeigen, dass es hinsichtlich des Einsatzes von gamifizierten Inhalten in einer Anwendung von hoher Relevanz ist, eine detaillierte Untersuchung der Bedürfnisse der Zielgruppe durchzuführen, um eine erfolgreiche Gamification-Strategie einsetzen zu können.

2.2. Vorstellung des Frameworks Flutter

Im Rahmen des Projekts ergibt sich durch die Aufgabenstellung die Verwendung des Frameworks Flutter zur Entwicklung der mobilen Ziel-Applikation. Im folgenden Verlauf wird das Framework näher betrachtet und dessen Funktionsweisen und Besonderheiten beschrieben, um einen besseren Eindruck darüber zu bekommen, warum genau Flutter als Teil der Problemlösung eingesetzt wird.

2.2.1. Was ist Flutter?

Flutter ist ein von Google entwickeltes Open-Source-Framework, das auf die Entwicklung mobiler 2D-Applikationen für Android- und iOS-Betriebssysteme ausgelegt ist.

Beworben wird Flutter seit der Ankündigung der Beta-Version, welche in einem offiziellen Blogeintrag der **Google Developers**[3] im Rahmen des **Mobile World Congress 2018**[28] veröffentlicht wurde, unter anderem durch eine besonders flexibel handhabbare und mit Leichtigkeit umzusetzende Gestaltung von hochqualitativen Benutzeroberflächen. Weiterhin wird betont, dass die zu erzielenden Ergebnisse das Produkt einer extrem schnell umsetzbaren Entwicklung “in Rekordzeit“ sind.

Das Flutter Framework besteht aus verschiedenen Basiskomponenten, welche aus aufeinander aufbauenden Systemschichten bestehen (siehe Abbildung 2.2) und sich auf oberster Abstraktionsebene auf die Flutter-Engine und das Flutter-Framework unterteilen lassen.

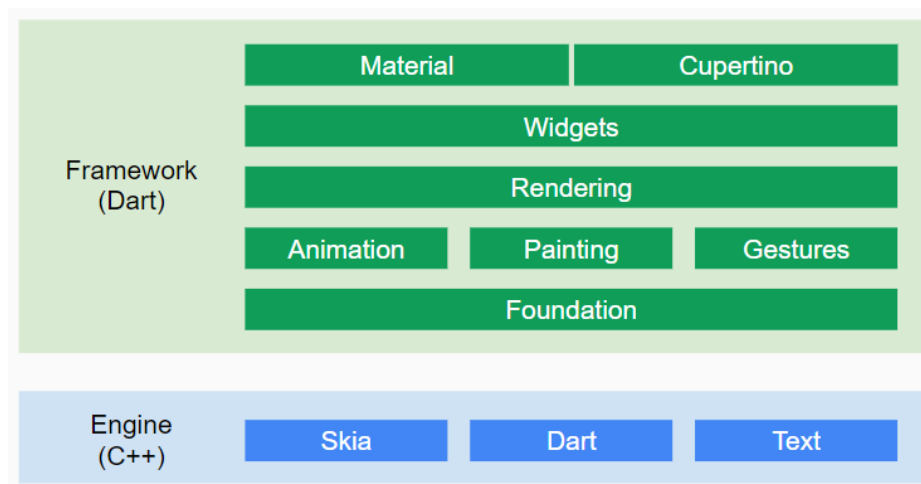


Abbildung 2.2.: Flutter High-Level Systemarchitektur [16]

Flutter Engine

Wie im Flutter-Wiki [43] beschrieben wird, setzt sich die C/C++ basierte Flutter Engine aus den drei Kerntechnologien Skia[39], einer “open source 2D Graphics Library“, die Dart Virtual Machine sowie die verschiedenen verwendeten Bibliotheken zum Rendern von Texten zusammen.

Foundation Library

Die Dart-basierte Foundation Library[17] stellt Basisklassen und -funktionen zur Verfügung, die von den anderen in der Flutter-Architektur überliegenden Schichten genutzt werden (siehe Abbildung 2.2) und stellt somit eine zentralen Grundlage bei der Konstruktion von Applikationen mittels dem Flutter-Framework dar.

Design-specific Widgets

Das Flutter-Framework stellt zwei verschiedene Design-Stile der Widgets zur Verfügung, welche den jeweiligen Design-Sprachen von **Google Material Design**[24], welche 2014 entwickelt wurde und der **iOS Design**[20] kopierenden Design-Sprache **Cupertino**[9] entsprechen.

2.2.2. Plattformübergreifende Entwicklung von mobilen Applikationen

Flutter ermöglicht das plattform-unabhängige Entwickeln von mobilen Applikationen für Android- und iOS-Betriebssysteme und verarbeitet oberflächen-spezifische Darstellungsformen der jeweiligen Zielplattformen framework-intern. Dies bedeutet, dass zum Zeitpunkt der Entwicklung auf extrem abstrakter Ebene ein Oberflächendesign unabhängig von der Zielplattform konzipiert werden kann, da die plattform-spezifischen Darstellungsweisen von dem Framework selbst gelöst werden und von den jeweiligen Entwicklern nicht berücksichtigt werden müssen.

Hinweis: Die in Abbildung 2.3 gezeigten Oberflächen entstammen demselben Programmcode.

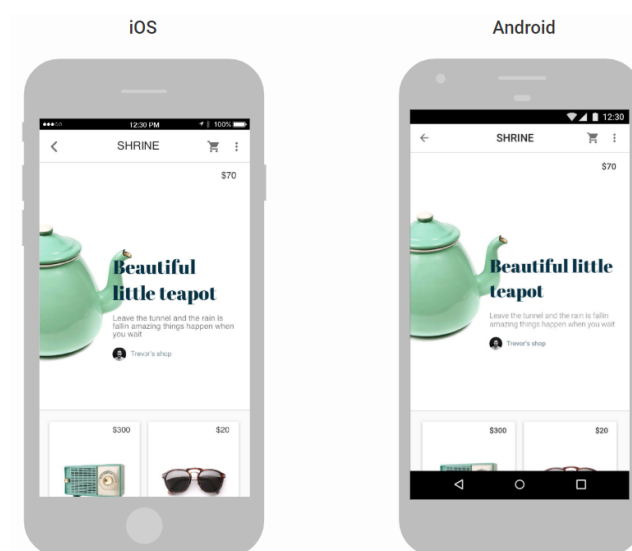


Abbildung 2.3.: Darstellung (iOS und Android): Plattformunabhängige Entwicklung von Applikationen [41, What is Flutter?]

2.2.3. Flutter Widgets

Alle User-Interface Grundbausteine in Flutter werden als sogenannte Widgets bezeichnet und dementsprechend gleich behandelt([16]). Diese Grundbausteine stellen die jeweiligen Design- und Interaktions-Elemente der Applikation dar und umfassen verschieden spezialisierte Ausprägungen, wie in Abbildung 2.4 beispielhaft zu sehen ist.

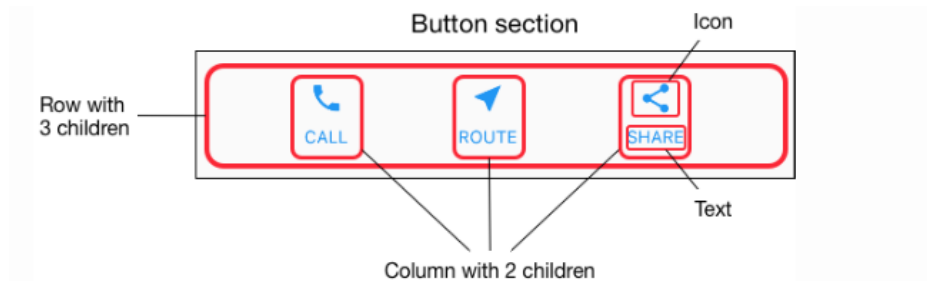


Abbildung 2.4.: Beispiel: Aufbau von Widgets [7]

Ein wichtiger Aspekt ist, dass Flutter im Gegensatz zu anderen Alternativen eine eigens erstellte Systemschnittstellen-Architektur besitzt und nicht auf OEM-Widgets oder DOM-Web Views zurückgreift[45], wie in Abbildung 2.5 verdeutlicht wird. Dies ermöglicht eine extrem freie Handhabung und Individualisierung im Umgang mit den Widgets und plattformspezifischen Diensten.

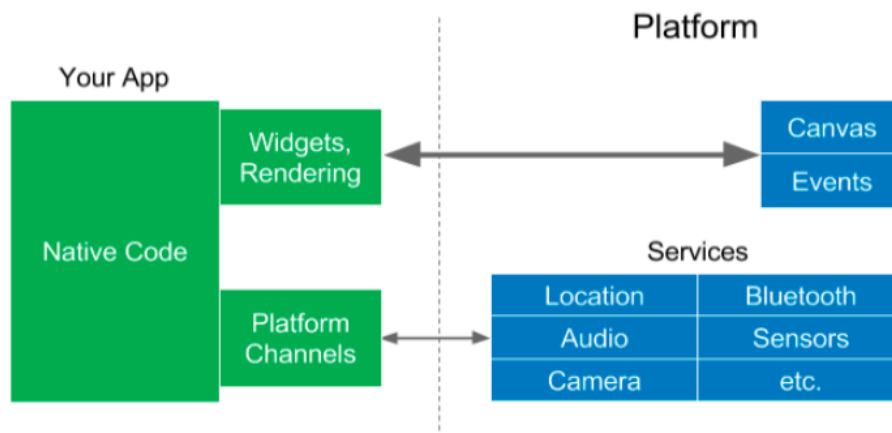


Abbildung 2.5.: Flutter Widgets Architektur [45]

Diese Widgets bilden zur Laufzeit eine hierarchische Struktur und setzen sich selbst aus weiteren Widgets zusammen. Somit ist der Ursprung einer jeden Programmoberfläche ein Wurzel-Widget, welches wiederum Kinder-Widgets enthält und diese somit über mehrere Ebenen eine Gesamtkomposition bilden, den sogenannten **Widget Tree**.

Widgets können also je nach Ausprägung beispielsweise strukturierende Elemente, wie Spalten und Zeilen sowie stil-definierende Elemente in Form von Farbdefinitionen darstellen. Weiterhin können Widgets auch Interaktions-Elemente, wie Buttons und Textfelder und noch viele weitere Elemente mit unterschiedlichen Eigenschaften definieren.

2.2.4. Flutter UI Rendering

Flutter verwendet im Gegensatz zu anderen Alternativen nach eigener Aussage eine selbst entwickelte und optimierte Rendering-Engine.[42].

Sollte sich der aktuelle Zustand der Applikation beispielsweise durch eine Benutzeraktion verändern, so werden die in der Baumstruktur vorhandenen Widgets neu generiert, deren Eigenschaften aktualisiert und dem Benutzer letztendlich angezeigt.

Abbildung 2.6 stellt den soeben beschriebenen Ablauf dar:

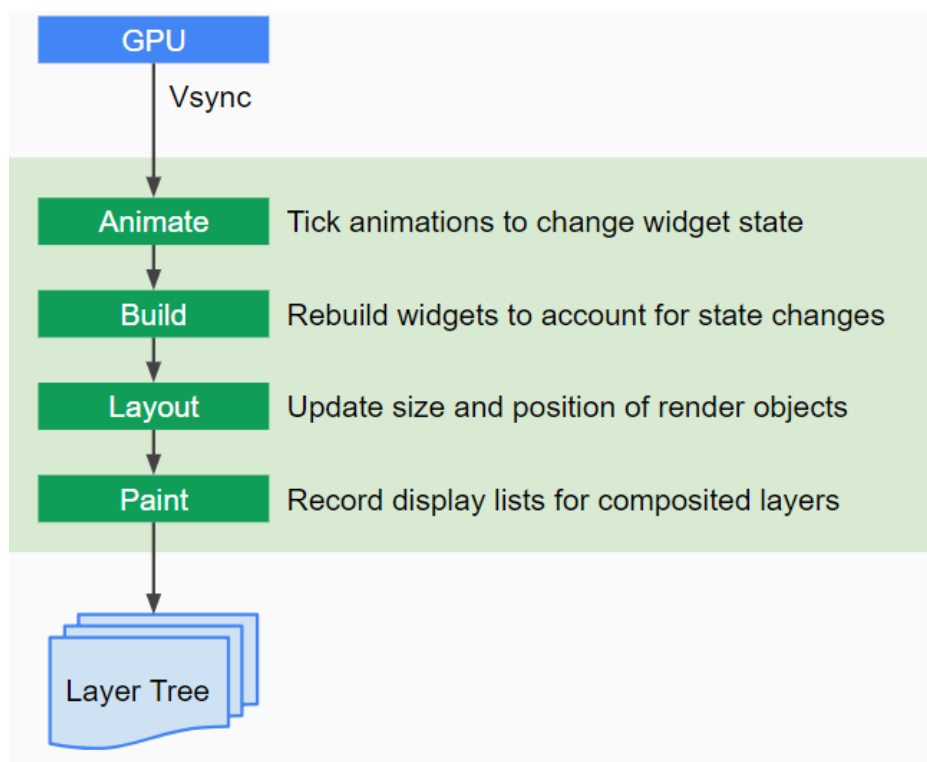


Abbildung 2.6.: Flutter Rendering [16]

3. Untersuchung des Problembereiches

Im folgenden Kapitel werden die Rahmenbedingungen und die Analysestrategie zu den durchgeführten Befragungen von Professoren und Studierenden aus dem Studiengang Informatik/Softwaretechnik der Fachhochschule Lübeck dargestellt. Die daraus resultierenden Ergebnisse und Erkenntnisse beider Seiten werden in zusammengefasster Form beschrieben. Die Diskussion dieser Ergebnisse und die aus den durchgeführten Befragungen abgeleiteten Anforderungen an die zu entwickelnde Software stellt das Produkt der Anforderungsanalyse und somit den Ausgangspunkt für die gesamte Arbeit dar.

3.1. Hypothese

Die Auseinandersetzung mit komplexen Problemstellungen, wie die als abschließende Prüfungsleistung des Studiums zu erbringende Bachelorarbeit, stellt erfahrungsgemäß für viele Studierende eine große Herausforderung dar. Diese Herausforderung ergibt sich aus dem erstmaligen Zusammenspiel aus selbstständigem und eigenverantwortlichem Arbeiten und dem Problemlösen mittels erworbener Fach- und Methodenkenntnisse über einen längeren Zeitraum. Trotz des Verlaufs des Studiums, des angeeigneten Wissens und der somit zahlreich erworbenen Fähigkeiten kommt es im Kontext von Bachelorarbeiten dennoch oftmals zu Schwierigkeiten, diese Kompetenzen auf reale Probleme abzubilden und zu dokumentieren.

3.2. Identifikation der Interessengruppen

Die folgende Tabelle 3.1 bietet eine Übersicht der, im Rahmen des Projektes identifizierten, Interessengruppen und welche Stellung diese zu dem Projekt bezüglich Einfluss, Einstellung und Erwartungen einnehmen.

Bezeichnung	Einfluss	Einstellung	Erwartungen	Bemerkungen
Studierende	Hoch	Positiv	Erhoffen sich optimale Ergebnisse und weniger Fallstricke bei dem Bearbeiten der Bachelorarbeit und diesbezüglich eine insgesamt umfangreich ausfallende Hilfestellung.	Zielgruppe der zu entwickelnden Applikation.
AStA	Hoch	Positiv	Erhoffen sich eine angemessene Förderung und Entlastung der Studierenden bei der Bearbeitung der Bachelorarbeit.	Sprachrohr und Interessenvertreter der Studierenden.
Leiter des Bachelorseminars	Hoch	Positiv	Erhofft sich eine steigende Bereitschaft der Studierenden, im Rahmen des Bachelorseminars Beiträge zu erbringen sowie eine steigende Qualität der Vorbereitung, Bearbeitung und Fertigstellung von Bachelorarbeiten.	Leitet das Bachelorseminar, hat somit direkten Kontakt mit der Zielgruppe und trägt wichtige Erfahrungswerte bezüglich der Probleme der Studierenden mit sich.
Professoren	Hoch	Positiv	Erhoffen sich eine steigende Qualität der Bearbeitung der Bachelorarbeit sowie eine Entlastung bei dem Betreuen von Studierenden, hinsichtlich sich wiederholenden Erklärungen und weiterer Trivialitäten.	Tragen wichtige Erfahrungswerte durch den detaillierten Einblick als Betreuer von Bacheloranden sowie dem Bewerten von Bachelorarbeiten mit sich.
Präsidium	Mittel	Positiv	Erhoffen sich eine Steigerung der Leistung der Studierenden an der Fachhochschule Lübeck.	Machtpromotor der Fachhochschule Lübeck.

Tabelle 3.1.: Übersicht der Interessengruppen

3.3. Wahl der Analysestrategie

Um den Problembereich zu ermitteln und somit eine Analysegrundlage zu schaffen, wurden unterschiedliche Personengruppen der Fachhochschule Lübeck durch verschiedene Befragungs- und Analysemethoden in das Projekt miteinbezogen. Dies soll einen detaillierten Einblick in die Sichtweisen der unterschiedlichen beteiligten Personen und Interessengruppen ermöglichen und somit eine Grundlage für das Verständnis der aktuellen Situation bilden.

Als primäre Einflussgeber, die in die Entwicklung der Applikation stark eingebunden werden, wurde die Gruppe der Professoren und die Gruppe der Studierenden identifiziert. Diese Entscheidung wurde aufgrund der im Rahmen der Bachelorarbeit existierenden unterschiedlichen Sichtweisen sowie der unmittelbaren Berührung mit dem Problembereich der Beteiligten und den unterschiedlichen Erfahrungsständen von Betreuern und Bacheloranden getätigt (siehe Abbildung 3.1). Darüber hinaus handelt es sich bei den Studierenden um die Zielgruppe und potentielle spätere Anwender, weshalb die Aufnahme der Probleme und Meinungen der Studierenden als essenziell angesehen wird.

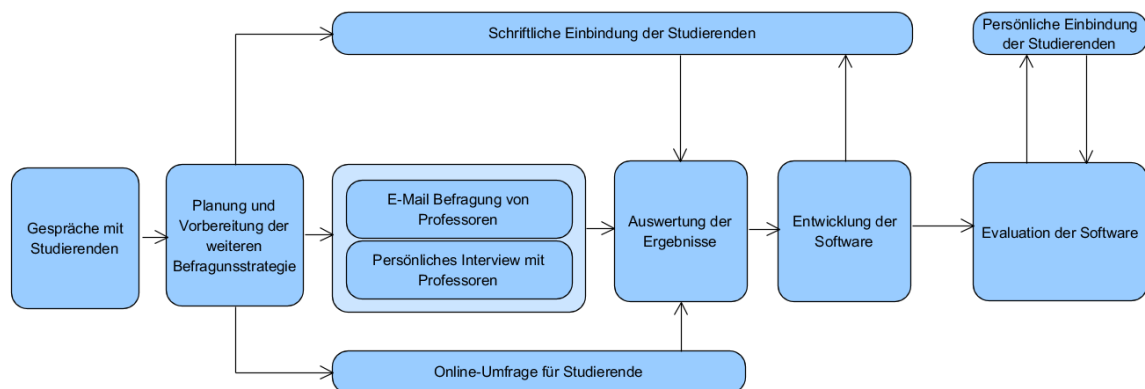


Abbildung 3.1.: Darstellung der Analysestrategie

Durch die initial durchgeführten Gespräche mit den Studierenden ließen sich Erkenntnisse über den Umfang des Problems gewinnen. Auf Grundlage dieser Erkenntnisse wurde die weitere Analysestrategie verfeinert und für die Umsetzung durch die Erstellung von Fragenkatalogen und Interview-Leitfaden vorbereitet (siehe Abbildung 3.1), welche im folgenden Verlauf beschrieben werden.

Die Interviews mit den Studierenden sollen somit als Grundlage zu Konzeption und Design weiterer Befragungsstrategien dienen und stellen in diesem Umfang einen Ausgangspunkt für die Problemanalyse dar. Weiterhin wurden die gewählten Interviewpartner bei Zustimmung über das gesamte Projekt und nach Absprache auch in die Evaluation der Applikation miteinbezogen.

Durch das Durchführen von Gruppeninterviews sollen die Studierenden zu Diskussionen angeregt werden, welche durch den Leiter des Interviews gezielt angeregt werden können. Dies hat den Zweck, die Situation und die Rolle der Studierenden für den Interviewer erkenntlich zu machen. Diese persönlichen Interviews dienen in erster Linie also nicht der Erhebung konkreter Daten, sondern als erster Berührungspunkt mit dem Problembereich und in diesem Rahmen als strategischer Orientierungspunkt für das weitere Vorgehen.

3.3.1. Einbezug der Professoren

Für die Interessengruppe der Professoren aus dem Fachbereich Informatik wurden als Grundlage der Datenerhebung Einzelinterviews mit einer ausgewählten Gruppe von Professoren durchgeführt, während eine weitere Gruppe von Professoren schriftlich per E-Mail befragt wurde (siehe Abbildung 3.1). Dies bietet sowohl den Zugriff auf die unmittelbaren Erfahrungen der einzelnen Professoren als Spezialisten in ihren jeweiligen Fachgebieten, als auch auf die Erfahrungen der Professoren in der Position eines Betreuers und Ansprechpartners für Bacheloranden.

Durch das Durchführen von Einzelinterviews wird ermöglicht, die Erwartungen seitens der Professoren an die Bacheloranden im Detail zu identifizieren und die in dieser Hinsicht priorisierten inhaltlichen und methodischen Aspekte bei der Bearbeitung einer Bachelorarbeit herauszuarbeiten. Die Aufteilung auf Einzelinterviews und E-Mail-Befragungen bietet den Vorteil, beide Strategien simultan zu verfolgen und nach Abschluss der Datenerhebung sowohl die detaillierten Einzelinterviews, als auch die oberflächlicher ausfallenden E-Mail Antworten in bereits dokumentierter Form vorliegen zu haben, um diese dann auszuwerten.

3.3.2. Einbezug der Studierenden

Nach Absprache mit den interviewten Studierenden werden diese weiterhin per schriftlicher Befragung in die Entwicklung des Projektes, unter ergänzender Zunahme anderer Studierender, eingebunden. In diesem Zusammenhang werden die teilnehmenden Studierenden mittels Bildern, Fragen und Gestaltungsideen der Applikation regelmäßig über den aktuellen Stand der Entwicklung informiert. Auf diese Weise sollte es möglich sein, ein breites, jedoch persönliches Feedback zu erhalten, da sich die Einbindung der Studierenden auf diese Weise sehr gut automatisieren lässt. Wesentlich dabei ist auch die Tatsache, dass die Reaktionsfreudigkeit der Studierenden höher ausfällt, als für die im Vergleich existierende Alternative der zeitaufwändigeren persönlichen Interviews.

Aufgrund der Durchführung der schriftlichen Befragung wird der Ablauf bereits dokumentiert und der Fragesteller sowie die Studierenden haben jederzeit die Möglichkeit Fragen und Anregungen zu teilen. Miteinbezogen werden vorzugsweise Studierende der oberen Semester, unabhängig davon, ob sie sich vor Beginn, während der Bearbeitung oder nach Abschluss der Bachelorarbeit befinden, da die unterschiedlichen Sichtweisen und Erfahrungsstände wichtige Impulse für die Entwicklung der Applikation geben.

3.4. Ergebnisse der Professoren

Im Folgenden sind die gewonnenen Eindrücke und Kenntnisse der Einzelinterviews mit den Professoren des Fachbereichs Informatik durch Themenkategorien geordnet und in zusammengefasster Form dokumentiert.

Es haben insgesamt sechs Professoren an persönlichen Einzelinterviews teilgenommen, wobei es für fünf Interviews gestattet wurde, eine Tonaufzeichnung anzufertigen.

Des Weiteren wird auch eine schriftliche Befragung miteinbezogen, die jedoch das gleiche Fragespektrum wie die Interviews einnimmt.

Allgemeine Informationen

Der zeitliche Rahmen der fünf aufgezeichneten Interviews erstreckt sich je über einen Zeitraum von etwa 30 bis 45 Minuten. Für die Auswertung der aufgezeichneten Interviews wurden die Aussagen der Interviewpartner, auf Grundlage der vorliegenden Audioaufnahmen, unter Berücksichtigung des Kontexts aufbereitet und werden nachfolgend dargestellt. Die aufgezeichneten Interviews mit den einzelnen Professoren sind in transkribierter Form im Anhang zu finden (siehe Anhang A.1).

Ein weiteres Einzelinterview, dass nicht aufgezeichnet wurde, erstreckte sich über einen Zeitraum von 75 Minuten. Für dieses Interview wurden begleitende Feldnotizen angefertigt und aufbereitet (siehe Anhang A.1.6). Weiterhin hat einer von sechs angefragten Professoren an der E-Mail-Befragung teilgenommen (siehe Anhang A.1.7).

Die nach dem Interview aufbereiteten Feldnotizen fließen zusammen mit den Ergebnissen der E-Mail-Befragung und den persönlichen Interviews in die folgende Zusammenfassung ein. Die ermittelten Kategorien ergeben sich aus dem gewählten Auswertungsverfahren, das sich an der qualitativen Inhaltsanalyse nach Mayring[25] orientiert und basieren auf den gemäß des Verfahrens herausgearbeiteten Codings.

3.4.1. Art der Arbeit

Im Laufe der Interviews wurden verschiedene Typen von Bachelorarbeiten versucht zu identifizieren. Dabei geht es vor allem darum, die Vielfalt der typischen Arbeiten des Studiengangs Informatik/Softwareentwicklung zu erfassen.

Als allgemeinen auftretende Arten von Bachelorarbeiten wurden die Klassen *Entwickelnde Arbeit* und *Evaluierende Arbeit* identifiziert. Weiterhin gibt es auch *Reine Literaturarbeiten*, welche in dem Studiengang Informatik/Softwareentwicklung jedoch nicht oder nur in einem sehr geringen Vorkommen auftreten.

Es folgt eine stichpunktartige Aufführung der gewonnenen Erkenntnisse:

Konstruktiv/Entwickelnd - Durchlaufen des Softwareentwicklungszyklus

Die Art der konstruktiven/entwickelnden Arbeiten, befassen sich hauptsächlich mit dem Durchlaufen des Softwareentwicklungszyklus. Im Rahmen der Bachelorarbeit werden die verschiedenen Aufgaben der Softwareentwicklung, je nach Schwerpunkt der Arbeit, mehr oder weniger intensiv von dem Bacheloranden bearbeitet und beschrieben.

Typischer fachlicher Inhalt einer solchen Arbeit könnte sein:

1. Anforderungsanalyse
2. Entwurf einer Softwarearchitektur
3. Implementierung eines Softwareprototyps
4. Evaluation des Softwareprototyps

Typische Aufgabenstellungen:

- Entwicklung einer mobilen Applikation zur Interpretation von Bildmaterial.
- Entwicklung einer mobilen Applikation zur Steigerung der Bereitschaft bei Senioren und Seniorinnen Fitnessaktivitäten auszuführen, unter Einbezug von Gamification-Elementen.
- Entwicklung einer Software zur Optimierung der täglichen Arbeitsabläufe im Unternehmen A.

Ein bei externen Arbeiten wichtiger Aspekt sind die möglichen Interessenunterschiede zwischen dem externen Unternehmen und dem internen Betreuer der Fachhochschule, welche einen Einfluss auf die Inhalte der Bachelorarbeit haben können. Externe Unternehmen sind tendenziell eher an dem resultierenden Ergebnis interessiert, während die internen Betreuer darüber hinaus einen hohen Wert auf nachvollziehbare Methodik, Herangehensweise sowie Planung und das saubere wissenschaftliche Arbeiten legen und somit ein hohes Interesse an dem Gesamtprozess haben.

Analytisch/Evaluierend - Vergleich, Auswertung und/oder Nachweis eines Aufgabengegenstandes

Die Art der analytisch/evaluierenden Arbeiten beinhalten hauptsächlich die Analyse und Auswertung eines Aufgabengegenstandes oder mehrerer verschiedener Aufgabengegenstände. Es wird der gesamte Weg der Analysestrategie, der Durchführung der Analyse und der Auswertung behandelt.

Typischer fachlicher Inhalt einer solchen Arbeit könnte sein:

1. Erstellen eines Kriterienkatalogs
2. Aufbau des Experiments
3. Durchführung des Experiments
4. Evaluation und Ergebnisauswertung

Typische Aufgabenstellungen könnten sein:

- Evaluation der Gesichtserkennungsdienste von Unternehmen A, Unternehmen B und Unternehmen C.
- Untersuchung des Verhaltens einer neuen Technologie A, im Vergleich mit einer alten Technologie B.
- Datenbankanalyse unter Anwendung von Machine-Learning-Algorithmen.

Reine Literaturarbeiten

Reine recherchierende Arbeiten finden in dem Studiengang Informatik/Softwareentwicklung aufgrund dem geringen Interesse seitens der Studierenden kaum statt und werden aus Gründen der Vollständigkeit lediglich erwähnt und nicht ausgeführt.

3.4.2. Erwartungen an den Bacheloranden

Im Laufe der Interviews wurden die Professoren hinsichtlich Ihrer Erwartungen an die Bacheloranden befragt und haben in diesem Rahmen häufig gleiche oder ähnliche Punkte ausgeführt. Aus diesem Grund werden im folgenden Verlauf die Meinungen der befragten Professoren aus der Sicht als Betreuer zu den jeweiligen Aspekten als zusammengefasstes Meinungsbild wiedergegeben.

Es folgt die Ausführung der gewonnenen Erkenntnisse:

Selbstständiges Arbeiten

Das selbstständige Arbeiten und Vorgehen ist eine der am häufigsten genannten Erwartungen, was sich in unterschiedlichen Punkten äußert. Dazu zählen vor allem das selbstständige Kommunizieren von Ergebnissen und das Einholen von Feedback sowie die Transparenz bei Problemen oder Schwierigkeiten, um sich Hilfe von dem Betreuer zu holen. Es wurde mehrfach betont, dass es im Allgemeinen nicht die Aufgabe des Betreuers ist, nachzufragen und aufzufordern. Das Einbinden des Studierenden in die Erarbeitung der Aufgabenstellung ist ein häufig genannter Punkt, der bereits frühzeitig Engagement des Studierenden erfordert. Des Weiteren stellt das selbstständige Einarbeiten in die Probleme und die Auswahl geeigneter Methoden und Werkzeuge einen wesentlichen Inhalt der Arbeit dar.

Die Vorgehensweise

Sehr häufig wird betont, dass die Vorgehensweise hinsichtlich der wissenschaftlichen Arbeitsweise von essenzieller Bedeutung ist. In diesem Rahmen soll der Studierende auch zeigen, dass er in der Lage ist, große Probleme systematisch in Teilprobleme zu zerlegen und diese unter Berücksichtigung der im Studium vermittelten Methoden, Modelle und Techniken zu bearbeiten. Sehr wichtig ist dabei das vorausschauende Planen von Teilprozessen, wie beispielsweise das Evaluieren der Ergebnisse. Dies sollte von Anfang an berücksichtigt werden und die in diesem Rahmen getroffenen Entscheidungen sollten nachvollziehbar erklärt werden können. Dazu zählt weiterhin das Erstellen eines Zeitplans, welcher sich im Verlauf jedoch durchaus verändern kann. Es wird sehr viel Wert darauf gelegt, zu sehen, dass die Studierenden einen weiten Blick auf das gesamte Projekt entwickeln und pflegen.

Der Literaturteil

Es wird betont, dass vor allem Wert auf einschlägige Quellen gelegt wird. In diesem Zusammenhang ist es wichtig, dass die Studierenden Literaturquellen verwenden sollten, die bereits eine anerkannte längere Gültigkeit besitzen. Weiterhin sollten die Studierenden über den Umfang von Grundlagenliteratur hinaus blicken und je nach Themengebiet und Arbeitsstand spezifischere Fachliteratur in die Arbeit einbeziehen. Dies kann auch bedeuten, dass auf wissenschaftliche Papiere und Primärquellen verwiesen werden soll. Je nach Thema und Aufgabenstellung kann der Literaturteil mehr oder weniger umfangreich ausfallen, was sich durch den praktischen Teil der Arbeit ausgleichen lässt.

Der praktische Teil

Im Wesentlichen soll der praktische Teil den Umfang der Aufgabenstellung abdecken und gegebene und/oder erhobene Anforderungen erfüllen. Je nach Thema und Aufgabenstellung der Arbeit nimmt dieser Teil einen höher oder niedriger ausfallenden Umfang ein. Der Studierende soll bei der praktischen Bearbeitung der Aufgabe zeigen, dass er in der Lage ist, im Studium gelerntes Wissen sowie kennengelernte Methoden und deren Ausführung umzusetzen.

Das Ergebnis der Bachelorarbeit

Grundsätzlich soll die Bachelorarbeit aus zwei Teilen von Leistungen bestehen. Die von dem Studenten durchgeführte Literaturarbeit nimmt einen Teil der Arbeit ein, während die eigenständige praktische Leistung den anderen Teil erfüllt. Der Umfang der jeweiligen Anteile kann dabei je nach Themengebiet und Aufgabenstellung stark variieren.

Die angestrebte Zielstellung einer Bachelorarbeit soll wissenschaftlich erarbeitet werden und somit im Ganzen auch nachvollziehbar und belegbar sein. Es kann jedoch auch vorkommen, dass das angestrebte Ziel aus verschiedenen Gründen nicht erreicht wurde. Dies muss nicht bedeuten, dass es zu einer schlechten Bewertung der Bachelorarbeit kommt, sofern der Grund oder die Erkenntnis über ein bestimmtes aufgetretenes Problem belegbar und nachvollziehbar dokumentiert ist.

3.4.3. Häufig auftretende Probleme

Es stellte sich im Verlauf des Interviews heraus, dass Studierende immer wieder mit gleichen oder ähnlichen Problemen zu kämpfen haben. Im folgenden Verlauf werden diese genannten Probleme in aufbereiteter Form stichpunktartig beschrieben.

Es folgt die Ausführung der gewonnenen Erkenntnisse:

Zeitmanagement

Der am häufigsten genannte Aspekt ist das mangelhafte Zeitmanagement der Studierenden. Im Laufe der Bearbeitung der Bachelorarbeit kommt es häufig zur Unterschätzung des nötigen Zeitaufwandes, besonders hinsichtlich des Schreibens der Dokumentation.

Viele Studierende verschieben das Schreiben der Dokumentation auf einen späteren Zeitpunkt und geraten im späteren Verlauf der Bearbeitungszeit somit unter Zeitdruck. Die Betreuer haben unter diesen Umständen wenige Möglichkeiten rechtzeitiges und hilfreiches Feedback zu liefern. Es wird häufig empfohlen, frühzeitig mit dem Schreiben zu beginnen und begleitend zum Arbeitsfortschritt die Dokumentation an mehreren Stellen wachsen zu lassen. Trotz vieler Hinweise seitens der Betreuer kommt es in diesen Belangen häufig zu großen Problemen.

Das Problem des mangelnden Zeitmanagements äußert sich auch darin, dass die Studierenden sich bei der Bearbeitung in nicht relevanten Details verlieren, da sie die Schwerpunkte der Arbeit nicht erkennen.

Wissenschaftliches Arbeiten

Die Studierenden erfassen teilweise nicht die Bedeutung des wissenschaftlichen Arbeitens. Es kommt immer wieder zu Schwierigkeiten und Unklarheiten über den eigentlichen Umfang der Arbeit und wodurch sich das wissenschaftliche Arbeiten auszeichnet. Oft wird der falsche Umgang mit Literatur und Quellen als negatives Beispiel genannt.

Ein Kritikpunkt ist, dass es in dem Studiengang Informatik/Softwareentwicklung keinen Kurs gibt, welcher die Studierenden auf das wissenschaftliche Arbeiten vorbereitet. In einigen Wahlpflichtmodulen werden diesbezüglich zwar Ansätze im Rahmen von Projekten integriert, jedoch gilt dies somit nur für die an dem Wahlpflichtmodul teilnehmenden Studierenden und steht auch nicht im Fokus der Projektarbeit.

Kommunikation

Kommunikation und Transparenz ist ein weiteres angesprochenes Problem. Es kommt vor, dass Studierende und Betreuer unterschiedliche Ansichten über die Zusammenarbeit haben, die sich dadurch äußern, dass die Studierenden auf Aufforderungen bezüglich Leistungsständen oder Ergebnissen seitens der Betreuer warten oder sich sogar aus diversen Gründen nicht trauen, ihren aktuellen Arbeitsstand oder ihre Probleme mit dem Betreuer zu teilen.

Die Vorbereitung der Studierenden

Bei der Vorbereitung der Studierenden nennen die beteiligten Interviewpartner unterschiedliche Aspekte. Es wird zum einen das mangelhafte selbständige Informieren der Studierenden kritisiert, andererseits wird jedoch auch eine besser zu gestaltende Vorbereitung der Studierenden seitens der Fachhochschule für den Studiengang Informatik/Softwareentwicklung betont.

Das angebotene Bachelorarbeit-Seminar wird positiv erwähnt, da es einen positiven Einfluss auf die Arbeit der Studierenden hat. Es müssen weniger Aspekte einer Bachelorarbeit erklärt werden, jedoch müssen viele Aspekte mehrfach wiederholt werden.

Es wird betont, dass auch viele Informationsmaterialien im Lernraum der Fachhochschule Lübeck zu finden seien, auf die auch oft hingewiesen wird, jedoch von den Studierenden nicht in dem Umfang beachtet werden, für den die Materialien gedacht sind. Dabei wird unter anderem auch kritisiert, dass die Informationsmaterialien teilweise schwer auffindbar sind, da sie nicht an einer zentralen Stelle, sondern verteilt im Lernraum liegen.

Weiterhin wird jedoch auch betont, dass die Studierenden zu wenig Engagement aufbringen, sich trotz vieler Möglichkeiten selbstständig zu informieren.

3.4.4. Die Applikation – Wünsche und Anregungen

In den Interviews wurden die Professoren nach ihren persönlichen Wünschen, Ideen und Anmerkungen bezüglich der Applikation gefragt. Es folgt die Ausführung der aufgenommenen Wünsche und Ideen.

Neuer Kanal zu den Studierenden

Es wird der Wunsch geäußert, dass die Applikation verwendet werden kann, um in zentraler Form konkrete interne oder externe Bachelorarbeit-Themen sowie Beispielthemen angeben zu können, da dies an der Fachhochschule Lübeck bisher nicht ermöglicht ist.

Plattform als Informationsquelle

Es besteht der Wunsch, die im Lernraum vorliegenden Informationsmaterialien, durchaus auch in aufbereiteter Form, durch die Applikation den Studienreden zugänglicher zu machen.

Applikation zur Unterstützung des Zeitmanagements

Es wird der Wunsch geäußert die Studierenden beim Zeitmanagement, unter anderem durch Erinnerungen, zu unterstützen. In diesem Rahmen wird der Vorschlag eingebracht möglichst detaillierte Beschreibungen von Arbeitspaketen in dem Tool zu verlangen, damit die Studierenden dazu gezwungen sind, sich rechtzeitig mit der Aufwandseinschätzung zu beschäftigen.

3.4.5. Die Applikation - Chancen und Risiken

In jedem Interview bekamen die Professoren abschließend die Möglichkeit, ihre Erwartungen an eine solche Applikation auszuführen und besonders auf die, aus ihrer Sicht, möglichen Chancen und Risiken einzugehen.

Diese Anmerkungen werden folgend zusammengefasst dargestellt:

Chancen

- Die Applikation als neuer Kanal für die Studierenden, der dafür dienen kann, dass Studierende sich besser informieren können. Dies wird besonders in Bezug auf die Formalien einer Bachelorarbeit betont, da viele Studierende gar nicht wissen, was die Rahmenbedingungen einer Bachelorarbeit sind oder welche Regeln und Anforderungen es überhaupt gibt.
- Die Applikation kann im Gegensatz zum Bachelorseminar begleitend während der eigenen Bachelorarbeit genutzt werden kann. Dies kann dafür sorgen, dass die Aufnahmebereitschaft der Studierenden für Tipps, Empfehlungen und weitere Aspekte gesteigert wird, da sie sich zu diesem Zeitpunkt mit dem Problem konfrontiert sehen und somit der Lerneffekt am höchsten ist. In diesem Ansatz wird auch die Chance erkannt, dass der Fokus der Studierenden zum richtigen Zeitpunkt auf bestimmte wichtige Fragen gelenkt werden kann und somit grobe Fehler minimiert werden können.
- Besseres Zeitmanagement der Studierenden und die somit geförderten organisatorischen Fähigkeiten der Studenten.
- Die App könnte Probleme im Projektmanagement und bei der Gestaltung der Dokumentation minimieren.
- Das Senken des Beratungsaufwandes für Professoren und somit das Minimieren von sich wiederholenden Arbeitsabläufen für die unterschiedlichen Bacheloranden wird als Chance genannt, da in einfacher Form auf eine Applikation verwiesen werden kann, die alle nötigen Informationen enthält.

Risiken

- Die Applikation kennt nicht den realen Stand der Bachelorarbeit, sondern die Studierenden sind für die Verwaltung selbst zuständig. Wenn der Benutzer eine der Aufgaben abhakt, stellt dies unter Umständen nicht den echten Zustand der Bachelorarbeit dar und könnte dem Studierenden einen falschen Eindruck des Fortschritts geben.
- Die Applikation regt dazu an, sich durch das Zeitmanagementtool zu über-planen, was dafür sorgt, dass der Benutzer von der eigentlichen Arbeit abgehalten wird. In diesem Rahmen kann die Applikation dem Benutzer nicht die Eigenverantwortung abnehmen. Der Studierende kann der Applikation nicht die Schuld für einen Misserfolg geben.
- Gamification-Elemente könnten unter Umständen einen sehr begrenzten Effekt haben, da sie kein Interesse an einem Thema wecken können, sondern das Grundinteresse aus der Eigenmotivation erzeugt werden muss.
- Befürchtung, dass Studenten gegebenenfalls die Applikation als Leitfaden als unumstößlich ansehen könnten und somit durch unterschiedliche Ansichten in einen Konflikt mit dem Betreuer geraten können.
- Die App könnte missverstanden werden als Ersatz für die persönliche Betreuung – insbesondere fachliche Aspekte wird eine App naturgemäß nicht abdecken können. Es könnte weiterhin zu einem “Device Mismatch“ kommen, da Bachelorarbeiten üblicherweise nicht an mobilen Endgeräten entstehen man müsste also immer zwei Geräte bedienen: Notebook/Desktop-PC und Smartphone.
- Die Applikation könnte den Studierenden zu sehr unterstützen, sodass die eigentliche Arbeit der Prüfungsleistung gar nicht von den Studierenden selbst ausgeführt wird, sondern durch die Applikation. Dies könnte zu einer Minderung des Lerneffektes bei der Bearbeitung der Bachelorarbeit führen.

3.5. Ergebnisse der Studierenden

Das folgende Kapitel liefert einen Einblick in die Ergebnisse der Befragung der Studierenden und beschreibt die in diesem Rahmen gewonnenen Erkenntnisse.

Im Fokus der Befragung stehen die Erwartungen an die Bachelorarbeit und den Betreuer und die typischen Probleme und Sorgen der Studierenden. Die Erkenntnisse der Befragung werden abschnittsweise in zusammengefasster Form dargestellt. Die gesamten Antworten der schriftlich befragten Studierenden sind in kategorisierter Form im Anhang zu finden (siehe A.1.8). Weiterhin stehen auch die nicht aufbereiteten Originalaufzeichnungen der schriftlichen Befragung zur Verfügung, aufgrund des Umfangs aber nur in elektronischer Form auf der CD im Anhang.

Informationen zu der schriftlichen Befragung:

- Insgesamt wurden 17 Studierende schriftlich befragt.

3.5.1. Erwartungen an den Betreuer

In diesem Kapitel werden die ermittelten Erwartungen der Studierenden an den Betreuer einer Bachelorarbeit zusammengefasst beschrieben.

Teilen von Erfahrungen

Es werden sich Erkenntnisgewinne durch das Teilen von Erfahrungen der Betreuer gewünscht, was sich besonders auf die Arbeit mit externen Unternehmen bezieht. Hierzu zählen Hinweise zu Besonderheiten und die wichtigen Unterschiede zu einer internen Bachelorarbeit.

Weiterhin wird das gemeinsame Erarbeiten eines Themas, was bezüglich des Umfangs durch die Erfahrungen des Betreuers eine angemessene Form annehmen sollte, betont.

Kommunikation

Was die Kommunikationsbereitschaft betrifft, wird vor allem eine hohe Erreichbarkeit, das (schnelle) Beantworten von Fragen und die Möglichkeit, persönliche Treffen wahrnehmen zu können, von dem Betreuer erwartet.

Feedback

Grundsätzlich wird von dem Betreuer erwartet, dass dieser sich mit der Bachelorarbeit der Studierenden insofern beschäftigt, als dass die einzelnen Feedback-Punkte zum aktuellen Stand, Inhalt, Ideen und Umfang konstruktiv diskutiert werden können. Darüber hinaus erwarten die Studierenden, dass der Betreuer Hinweise bei Unverständlichkeiten der Dokumentation gibt und gegebenenfalls schlechte Ideen und Entwurfsentscheidungen hinterfragt.

In diesem Rahmen wird auch vereinzelt erwartet, dass der Betreuer Interesse an der Arbeit des Studierenden hat sowie Bereitschaft für die Beantwortung von Fragen und auf das Diskutieren von Lösungsmöglichkeiten eingeht.

Impulse des Betreuers

In wenigen Fällen erwarten die befragten Studierenden, dass der Betreuer aktiv an dem Prozess der Erstellung der Bachelorarbeit teilnimmt. Das bezieht sich auf die Unterstützung bei Zeitmanagement, dem Geben von Denkanstößen und Anregungen und das vom Betreuer ausgehende Einfordern von Leistungen.

3.5.2. Die größten Probleme der Studierenden

Die Studierenden wurden zu ihren, nach eigener Einschätzung, größten Unsicherheiten und Problemen gefragt und haben einige, sich im Laufe der Befragung wiederholende, Aspekte angesprochen, die im folgenden Verlauf zusammengefasst werden.

Schreiben der Dokumentation

Die von den Studierenden angesprochenen Punkte befassen sich in erster Linie mit Unsicherheiten bei der Einschätzung von Umfang, Inhalt und Aufbau der Dokumentation.

Des Weiteren werden Rechtschreibung und der korrekte Umgang mit Quellen genannt.

Zeitmanagement

Der Punkt Zeitmanagement stellt für viele Studierende eine grundsätzliche Herausforderung dar.

Fachliche Probleme

Teilweise werden auch fachliche Probleme unterschiedlicher Natur als große Herausforderung genannt. Dazu zählen zum einen Probleme mit der im Rahmen der Bachelorarbeit zu verwendenden Hardware, andererseits aber auch Softwareprobleme, das Entwickeln von sinnvollen Tests und das präzise Ermitteln von Anforderungen.

Sonstige Probleme

Weitere Probleme, die vereinzelt genannt werden, sind Probleme mit der Themenfindung, die Kommunikation zwischen internem Betreuer und der externen Firma oder die allgemeine Überforderung bei der Bearbeitung der Bachelorarbeit.

3.5.3. Wunsch nach besserer Vorbereitung auf die Bachelorarbeit

Um das Bild der Situation der Studierenden besser zu verstehen und darüber hinaus auch Impulse für die Applikation zu erfassen, wurden die Studierenden hinsichtlich Verbesserungsvorschlägen für den Verlauf des Studiums, unter dem Aspekt, was ihnen zu einer besseren Vorbereitung zur Bearbeitung der Bachelorarbeit verhelfen würde oder verholfen hätte, befragt.

Zeitmanagement

Im Allgemeinen wünscht sich ein großer Teil der Studierenden eine bessere Vorbereitung bezüglich des Zeitmanagements. Die Studierenden bemängeln eine unzureichende Vorbereitung auf das Abschätzen von Zeitaufwand sowie Planung von größeren Projekten und nennen in diesem Zusammenhang den Wunsch nach Möglichkeiten und Maßnahmen im Studium, wie das häufigere Einbinden von Projektarbeiten.

Wissenschaftliches Arbeiten

Häufig wird der Wunsch nach besserer Vorbereitung auf das wissenschaftliche Arbeiten genannt, es gibt Unsicherheiten bezüglich des Aufbaus einer wissenschaftlichen Arbeit, des Umgangs mit Quellen und des korrekten Einbindens von Quellcode-Inhalten. Als vorkommender Lösungsvorschlag wird ein Kurs, welcher zur besseren Vorbereitung vor dem 6. Semester stattfinden sollte, genannt.

Aufklärung zur Bachelorarbeit

Viele der befragten Studierenden wünschen sich vorbereitende und informierende Maßnahmen zur Bachelorarbeit. Dies bezieht sich vor allem auf das Klären organisatorischer Fragen, der wichtigen Formalien und des Umfangs der Bachelorarbeit. Im Detail werden auch vorbereitende Maßnahmen gewünscht, die sich auf die Vor- und Nachteile einer extern oder intern betreuten Bachelorarbeit beziehen.

Sonstige Anmerkungen

Weiterhin treten bei der Befragung auch Fälle auf, bei denen keine expliziten Veränderungen gewünscht werden, da die Studierenden zufrieden sind. Der Wunsch nach einem stärkeren Fokus auf Design-Pattern im Laufe des Studiums wurde einmal geäußert.

3.5.4. Einbezug der Studierenden in die Gestaltung der Applikation

Die Einbindung der Studierenden befasst sich mit spezifischeren Fragen hinsichtlich der motivierenden Gamification-Inhalte der zu entwickelnden Applikation. Der Verlauf der folgenden Befragungen fand somit nicht in einem Durchlauf statt, sondern ergab sich aus dem kontinuierlichen Einbinden der Studierenden. Es wurden bei der Befragung zwei Gamification-Ansätze vorgestellt.

Der erste Ansatz fällt unter anderem in die Benutzergruppe der **Socialiser** (Siehe Kapitel 2.1). Inhalt des Ansatzes ist es, die Motivation durch den sozialen Kontakt und Austausch mit anderen Teilnehmern zu erzeugen, indem diese den Fortschritt bei der Bearbeitung der Bachelorarbeit anderer Studierender sehen können.

Als Gegengewicht zum ersten Ansatz orientiert sich der zweite Ansatz an der Benutzergruppe **Achiever** (Siehe Kapitel 2.1). Dieser Ansatz beruht vor allem auf dem Abschließen und Sammeln von Erfolgen. Im Rahmen der Applikation wird es dem Benutzer bei dieser Idee ermöglicht, Erfolge durch Erfüllen von definierten Aufgaben zu erlangen und somit den Fortschritt des Projektes durch das Abschließen von geeigneten Aufgaben zu verfolgen.

Primäres Ziel des Einbezugs der Studierenden ist, zu bestimmen, welche Einbindung von Gamification-Strategien ansprechend auf die Zielgruppe der Studierenden wirken und in welcher Weise sich das Einbinden von Gamification-Elementen auf die Steigerung der Motivation von Studenten bei der Bearbeitung der Bachelorarbeit auswirkt.

In der folgenden Tabelle werden die Ergebnisse der Befragung verdeutlicht:

Art der Gamification-Strategie	Art der Reaktion	Häufigkeit
Motivation durch soziale Interaktion	Positiv	3
	Negativ	12
	Neutral	2
Sammeln von Achievements und Erfüllen von Aufgaben	Positiv	15
	Negativ	1
	Neutral	1
		17

Tabelle 3.2.: Einbezug der Studierende in Wahl der Gamification-Strategie

3.6. Untersuchung der Ergebnisse

Im folgenden Kapitel werden alle relevanten Kenntnisse, die aus der Datenerhebung und -auswertung hervorgehen, untersucht und analysiert. Dazu gehört die Identifikation des Problembereiches, die Auswahl der zu verfolgenden Motivationsstrategie bezüglich der Wahl der Spiel-Design-Elemente sowie die relevanten Produktfunktionen und deren abgeleitete Anforderungen, Chancen und Risiken.

3.6.1. Identifikation des Problembereichs

Durch die Befragung der Beteiligten und der Analyse der gewonnenen Informationen haben sich die Erwartungen der Beteiligten an die Benutzung der Applikation herausarbeiten lassen, welche in der folgenden Tabelle (siehe Tabelle 3.3) dargestellt sind.

Die dabei genannten Ziele und Erwartungen entsprechen dem allgemeinen Bild, das durch die Interviews und schriftlichen Befragungen ermittelt werden konnte, weshalb kleine Abweichungen in dieser Tabelle nicht mit aufgenommen wurden, da diese einen zu großen verzerrenden Einfluss auf das Gesamtbild hätten.

Bezeichnung der Beteiligten	Wünsche der Beteiligten
Professoren	Steigerung der Fragebereitschaft von Studierenden Weniger Aufwand in das sich wiederholende Informieren des Studierenden stecken Zielgerichtete Vorbereitung auf Besprechung mit dem Betreuer Höhere Qualität von Abschlussarbeiten und Dokumentationen Bessere Zeitplanung der Studierenden Systematische(re) Vorgehensweise der Studierenden Optimale Vorbereitung auf die Bearbeitung der Bachelorarbeit
Studierende	Optimales Ergebnis der Bachelorarbeit Kenntnisse über Formalien, Umfang und Ablauf der Bachelorarbeit Kontrolle über die Bachelorarbeit und den Ablauf Unterstützung Weniger Fallstricke bei der Bearbeitung der Bachelorarbeit

Tabelle 3.3.: Erwartung der Beteiligten an die Applikation

Aus den Ergebnissen der Befragungen geht hervor, dass die Probleme nicht in den unterschiedlichen Ansichten der Studierenden und der Professoren liegen, da beide beteiligte Gruppen sehr kompatible Erwartungen und Ziele besitzen.

Häufig wird von den Studierenden geäußert, dass sie sich unzureichend vorbereitet für die Bachelorarbeit fühlen und die größten Probleme der Studierenden im Bereich Zeitmanagement und dem wissenschaftlichen Arbeiten liegen. Zusätzlich äußern sich die Studierenden zu

fehlenden oder unzureichend aufgebauten Kenntnissen bezüglich der Rahmenbedingungen und weiteren Informationen zu Bachelorarbeiten als Ganzes.

Die Professoren sehen die Probleme bei den Studierenden eindeutig auch bei dem Zeitmanagement, dem wissenschaftlichen Arbeiten, der Kommunikation und einer mangelnden Vorbereitung, bezüglich der Formalitäten und des Ablaufs einer Bachelorarbeit.

Hervorzuheben ist der Aspekt, dass die existierenden Probleme sowohl den Professoren, als auch den Studierenden bewusst sind und unabhängig der unterschiedlich ausfallenden Sichtweisen übereinstimmen.

Aufgrund der resultierenden Ergebnisse der Problemuntersuchung lässt sich erwarten, dass die angestrebten Ziele der Studierenden und die Erwartungen der Professoren aufgrund dieser Wissens- und Erfahrungslücken unter Umständen nicht reibungslos im Rahmen der Bachelorarbeit erfüllt werden können. Auf diese Weise werden die gemeinsamen Ziele der Studierenden und der Professoren gefährdet und können unter Umständen nicht erfüllt werden, obwohl dies im Sinne beider Beteiligten ist.

3.6.2. Auswahl der Motivationsstrategie

Wie aus den Ergebnissen in Kapitel 3.5 in Tabelle 3.2 abzulesen ist, äußern sich die befragten Studierenden gegenüber dem vorgestellten Designvorschlag, welcher die soziale Einbindung der Studierenden vorsieht, überwiegend sehr kritisch. Im Vergleich dazu wird die vorgestellte Variante des Einsetzens von Achievements und Aufgaben in Kombination mit Herausforderungen stark positiv bewertet.

Aufgrund der eindeutigen Reaktionen der eingebundenen Studierenden wird die Entscheidung getroffen, den ersten Ansatz nicht weiter zu verfolgen, dafür aber den zweiten Ansatz auszubauen. Somit fallen die Nutzergruppen überwiegend in den Spannungsbereich der **Explorers** und **Achievers**, wie in Abbildung 3.2 verdeutlicht wird.

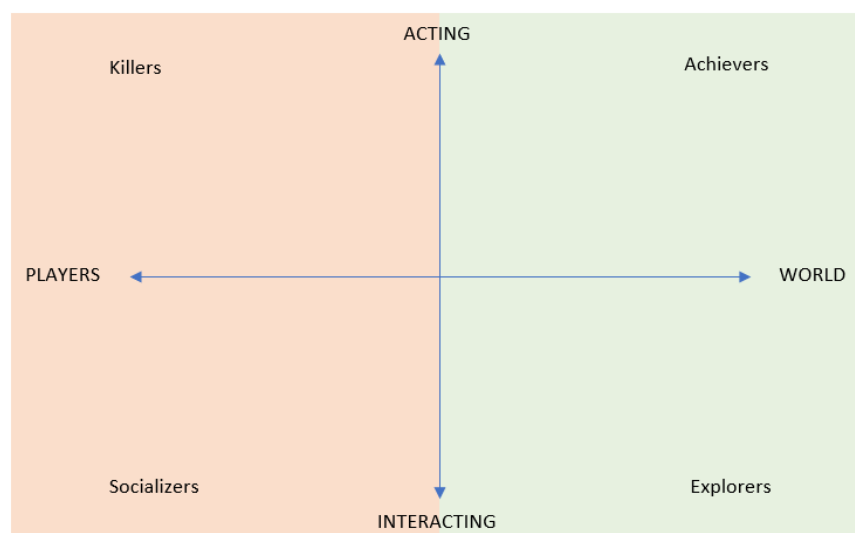


Abbildung 3.2.: Eingrenzung der Bartle-Benutzerklassen

Durch den vorgesehenen Kontext der Applikation, die Studierenden während der Bearbeitung der Bachelorarbeit wegweisend zu unterstützen und zu motivieren, bietet sich somit an, Herausforderungen und die damit verbundenen Achievements in Form von kontextbezogenen fokuslenkenden Game-Design-Elementen (siehe 2.1) in die Applikation als festen Bestandteil der Motivationsstrategie zu integrieren.

Wie bereits in Kapitel 2.1.2 beschrieben wird, besitzen Achievements verschiedene Eigenschaften, die in Bezug auf Zielstellungen einer steuernden und in diesem Sinne auch fordernden Funktion dienen können. Ein wichtiges Kriterium bei der Wahl der Achievements als Spiel-Design-Elemente ist, dass sie dem Benutzer ein nicht-kontrollierendes, positives Feedback geben und somit nicht unangenehm kontrollieren oder den ohnehin schon sehr hoch ausfallenden Druck auf die Studierenden bei der Bearbeitung der Bachelorarbeit erhöhen. Es sollte in diesem Sinne nicht aus den Augen verloren werden, dass die Applikation den Studierenden positiv unterstützen soll.

Im weiteren Verlauf werden, den Nutzergruppen entsprechend, auf Spiel-Design-Elemente wie Bestenlisten oder andere soziale Motivationsmodelle verzichtet.

Weiterhin wird das Integrieren eines Punkte- oder Level-Systems sowie einer Avatar-Funktion nicht verfolgt. Dies dient der Absicht, dass die Applikation zwar motivierende Spiel-Design-Elemente enthalten soll, jedoch in einem Umfang, der keine an Gamification desinteressierten Benutzer ausschließen und somit in diesem Fall weiterhin eine brauchbare Hilfestellung bieten soll.

3.6.3. Übersicht der abgeleiteten Produktfunktionen und Anforderungen

Im folgenden Verlauf sind die aus der Aufgabenstellung und der Anforderungsanalyse erhobenen Produktanforderungen in natürlicher Sprache vorzufinden. Diese wurden an einer Anforderungsschablone aus dem Lehrbuch der Softwaretechnik von Helmut Balzert orientiert [4] und nach Bedarf spezifiziert:

PR1 Unterstützung bei Zeit- und Fortschrittsmanagement

[Req1.1] Die Software muss den Studierenden ermöglichen, Meilensteine zu verwalten.

[Req1.1.1] Das Anlegen von Meilensteinen

[Req1.1.2] Das Entfernen von Meilensteinen

[Req1.1.3] Das Ändern von Meilensteinen

[Req1.1.4] Das Verschieben von Meilensteinen

[Req1.2] Die Software muss dem Studierenden ermöglichen, dem Meilensteinen zugehörige Aufgaben zu verwalten.

[Req1.2.1] Das Anlegen von Aufgaben

[Req1.2.2] Das Entfernen von Aufgaben

[Req1.2.3] Das Ändern von Aufgaben

[Req1.2.4] Das Markieren von Aufgaben als *abgeschlossen*

[Req1.3] Die Software soll die Studierenden auf bevorstehende Meilensteine und Aufgabenpakete aufmerksam machen.

[Req1.3.1] Das Auslösen von Erinnerungen/Benachrichtigungen

PR2 Unterstützung bei der Bearbeitung der Bachelorarbeit

[Req2.1] Die Software muss den Studierenden allgemeine Informationen zur Bearbeitung der Bachelorarbeit bereitstellen.

[Req2.1.1] Die Software muss den Studierenden Informationen über die Rahmenbedingungen und Formalien der Bachelorarbeit an der Fachhochschule Lübeck bereitstellen können.

[Req2.1.2] Die Software muss den Studierenden Hinweise und Empfehlungen zu Aufbau und Struktur von Bachelorarbeiten bereitstellen können.

[Req2.1.3] Die Software muss den Studierenden Hinweise und Empfehlungen zur Durchführung von Bachelorarbeiten bereitstellen können.

[Req2.2] Die Software muss den Studierenden eine Übersicht über studiengang-spezifische und -übergreifende relevante Techniken und Methoden bereitstellen können.

[Req2.4.1] Hilfestellung/Übersicht zu Themenfindung für Bachelorarbeit

[Req2.4.2] Hilfestellung/Übersicht von studiengang-spezifischen Inhalten

[Req2.4.3] Hilfestellung/Übersicht zu Aufbereitung von Ergebnissen

[Req2.4.4] Hilfestellung/Übersicht zu Nachweisführung

[Req2.3] Die Software muss den Umgang von zielgruppenspezifischen Anpassungen der Inhalte ermöglichen können.

[Req2.4] Die Anpassungen der Inhalte soll software-extern realisierbar und integrierbar sein.

PR3 Einsatz von Gamification-Elementen zur Steigerung der Motivation

[Req3.1] Die Software muss im Rahmen der Gamification-Strategie Spiel-Design-Elemente bereitstellen, die den ermittelten Benutzerklassen entsprechen.

[Req3.2] Die Spiel-Design-Elemente sollen unter Berücksichtigung der wegweisenden Eigenschaft geeignet an dem Ablauf einer Bachelorarbeit orientiert sein.

PR4 Weitere Anforderungen

[Req4.1] Die Software muss sämtliche Inhalte und Leistungen auch für fremdsprachige Personengruppen enthalten.

[Req4.2] Die Software kann die Möglichkeit bieten, dass Betreuer ihre persönlichen Themenvorschläge für die Studierenden sichtbar per Weboberfläche hinzufügen/entfernen können

3.6.4. Übersicht der Risiken

Im Laufe der mit den Professoren durchgeführten Interviews wurden häufig Bedenken bezüglich Risiken geäußert, die der Einsatz der Software für die Studierenden mit sich bringen könnte.

Die Stärke der Auswirkungen wird hierbei in die Klassen **Hoch** und **Gering** unterteilt, wobei sich diese nach den folgenden Eigenschaften definieren:

- Auswirkungen werden als **Hoch** bezeichnet, wenn ein schwerwiegend manipulativer Einfluss auf den Prozess der Bachelorarbeit zu erwarten ist.
- Auswirkungen werden als **Gering** bezeichnet, wenn kein absehbarer Einfluss auf den gewohnten Verlauf einer Bachelorarbeit zu erwarten sind.

Es folgt die Aufzählung der ermittelten Risiken sowie einer Gewichtung in Bezug auf die Stärke der negativen Auswirkungen bei Eintreten dieser Risiken:

[R1] Auswirkungen: Hoch

Durch die Nutzung der Applikation wird dem Studierenden so viel Arbeit abgenommen, dass der Lerneffekt der Bachelorarbeit zu gering ist.

[R2] Auswirkungen: Hoch

Die Applikation kennt den realen Zustand und Fortschritt der Bachelorarbeit nicht und könnte somit zu einer Verfälschung des realen Bildes führen. Studierende könnten somit den eigentlichen Zustand der Arbeit aus den Augen verlieren.

[R3] Auswirkungen: Hoch

Studierende könnten die Hilfestellung der Applikation als unumstößlich ansehen und so unter Umständen in Konflikt mit dem Betreuer geraten.

[R4] Auswirkungen: Hoch

Die Applikation kann als Ersatz eines Betreuers wahrgenommen werden, was zu weitreichenden Konsequenzen für die Qualität der Abschlussarbeit führen könnte.

[R5] Auswirkungen: Hoch

Die Applikation könnte im Rahmen des Zeitmanagement-Tools zu einer “Überplanung” führen, was die Studierenden von der eigentlichen Arbeit abhalten könnte.

[R6] Auswirkungen: Gering

Gamification-Elemente könnten keinen Effekt auf die Motivation der Studierenden haben.

3.6.5. Übersicht der Chancen

Aus den gewonnen Informationen, die sich aus dem Verlauf der Interviews mit den Professoren und der schriftlichen Befragung mit den Studierenden ergeben haben, ließen sich Chancen für Betreuer und Studierende durch Einsatz der Applikation identifizieren, welche im Folgenden genannt werden.

Die Stärke der Auswirkungen wird hierbei in die Klassen **Hoch** und **Gering** unterteilt, wobei sich diese nach den folgenden Eigenschaften definieren:

- Auswirkungen werden als **Hoch** bezeichnet, wenn ein positiver Einfluss auf den Prozess der Bachelorarbeit entgegen der Ist-Situation zu erwarten ist.
- Auswirkungen werden als **Gering** bezeichnet, wenn kein weitreichend positiver Einfluss auf den gewohnten Verlauf einer Bachelorarbeit zu erwarten ist.

Es folgt die Aufzählung der ermittelten Chancen sowie einer Gewichtung in Bezug auf die Stärke der positiven Auswirkungen bei Erreichen dieser Chancen:

[C1] Auswirkungen: Hoch

Die Aufnahmebereitschaft für Tipps und Empfehlungen könnte im Gegensatz zum Bachelorarbeit-Seminar gesteigert werden, da die Verwendung der App von begleitender Natur ist und sich die Studierenden besser mit dem Problem identifizieren könnten.

[C2] Auswirkungen: Hoch

Vermeiden von großen Fehlern durch Lenken des Fokus der Studierenden auf wichtige Aspekte der Bachelorarbeit, bei denen üblicherweise häufig Probleme auftreten.

[C3] Auswirkungen: Hoch

Besseres Zeitmanagement der Studierenden und die Steigerung der organisatorischen Fähigkeiten durch Anwendung der Applikation.

[C4] Auswirkungen: Hoch

Systematischere Vorgehensweise der Studierenden bei Problemanalyse, Anforderungserhebung und Nachweisführung

[C5] Auswirkungen: Hoch

Neuer Informationskanal für Studierende, welcher Wissenslücken bezüglich der Rahmenbedingungen und Regeln einer Bachelorarbeit schließen kann.

[C6] Auswirkungen: Hoch

Zielgerichteterer Methoden- und Werkzeugeinsatz sowie Anwendung fachspezifischer und fachübergreifender Techniken.

[C7] Auswirkungen: Hoch

Bessere Vorbereitung der Studierenden auf Gesprächstermine mit dem Betreuer.

[C8] Auswirkungen: Gering

Senken des Beratungsaufwandes seitens der Betreuer bei Trivialitäten und bei sich wiederholenden Fragestellungen.

4. Konzeptvorstellung der Applikation

4.1. Beschreibung der Software

Die zu entwickelnde Applikation kann in verschiedene Kern-Softwareabschnitte unterteilt werden. Diese Abschnitte lassen sich auf folgende namensgebende Aufgabenbereiche aufteilen, die sich an den erhobenen Anforderungen und der Aufgabenstellung orientieren (siehe Kapitel 3.6.3).

- Meilensteinverwaltung
- Bachelorarbeit Guide
- Herausforderungen
- Achievements
- Dashboard
- Sonstige Softwareinhalte

4.1.1. Meilensteinverwaltung

Die Meilensteinverwaltung ist eine Funktion, die das Anlegen, Planen und Verwalten von Meilensteinen sowie der zugehörigen Unteraufgaben ermöglichen soll.

Ferner soll die Meilensteinverwaltung nicht den detaillierten Prozess des gesamten Projektmanagements abbilden, sondern die Aufgabe der übersichtlichen Handhabung von Unteraufgaben und Verdeutlichung von Fristen in Form einer dynamischen To-Do-Listenverwaltung unterstützen. Hierbei soll dem Anwender weiterhin die größtmögliche Freiheit der Gestaltung des Plans zukommen, um individuelle Arbeitsweisen zu berücksichtigen.

Der im Projektmanagement definierte Begriff “Meilenstein” bezeichnet hierbei einen Zeitpunkt, zu welchem der vom Nutzer definierte Arbeitsstand erreicht sein soll.

Es folgt eine stichpunktartige Auflistung und Beschreibung der Kernfeatures des Tools:

- **Erstellung und Verwaltung von Meilensteinen**

Es können Meilensteine erstellt [Req1.1.1], bearbeitet [Req1.1.3], verschoben [Req1.1.4] und entfernt [Req1.1.2] werden.

- **Erstellung und Verwaltung von Unteraufgaben**

Jeder Meilenstein enthält optionale Unteraufgaben [Req1.2], die von dem Benutzer einem Meilenstein hinzugefügt [Req1.2.1], bearbeitet [Req1.2.3] und gelöscht [Req1.2.2] werden können. Weiterhin lassen sich die Unteraufgaben als *abgeschlossen* markieren [Req1.2.4], um dem Benutzer die Möglichkeit zu bieten, seine Aufgaben zu organisieren.

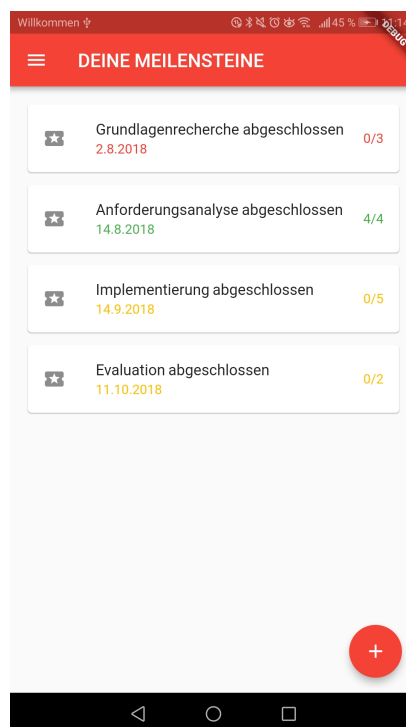


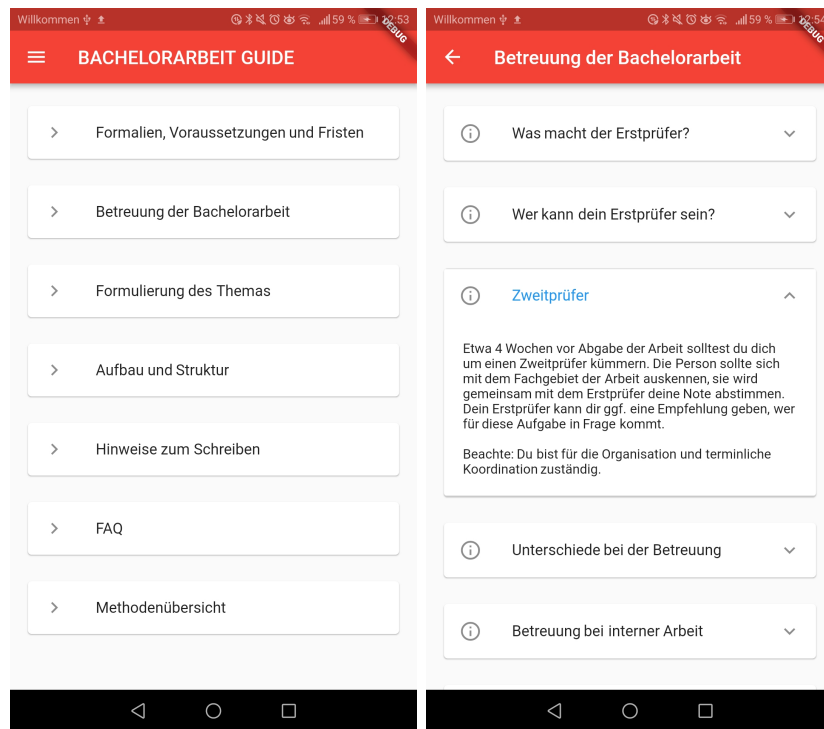
Abbildung 4.1.: Benutzeroberfläche Meilensteinverwaltung

4.1.2. Bachelorarbeit Guide

Der Guide stellt den Teil der Applikation dar, der die Bereitstellung von Hinweisen, Tipps und weiteren hilfreichen Informationen zur Bearbeitung der Bachelorarbeit abdecken soll. Dieser Inhalt ergibt sich aus der Verwendung des von der Fachhochschule Lübeck ausgehändigten Dokuments, welches als Ratgeber bei der Erstellung von Bachelorarbeiten funktionieren und somit die speziellen Anforderungen der Fachhochschule Lübeck berücksichtigen soll. [2]

In dieser Hinsicht soll der Guide als genereller Anlaufpunkt fungieren, der eine Hilfestellung für Bacheloranden darstellt. [Req2.2]

Die Bandbreite der Hinweise, Tipps und weiteren Informationen sollen somit den gesamten Verlauf der Bachelorarbeit abdecken und lässt sich in diesem Zusammenhang in folgende Bereiche und Inhalte zerlegen, welche sich an dem genannten Leitfaden der Fachhochschule Lübeck zur Erstellung einer Bachelorarbeit orientieren.



(a) Guide Hauptbildschirm

(b) Guide Themen-Ebene

Abbildung 4.2.: Benutzeroberfläche Guide

Es folgt eine Interpretation und Beschreibung der Unterteilung der Themengebiete:

Anmerkung: Die folgenden Themengebiete stellen nur eine Möglichkeit der Aufteilung dar. Bei Einsatz der Software sollen die hier dargestellten Inhalte individuell erweitert oder verändert werden können([Req2.3], [Req2.4]).

- **Formalien, Voraussetzungen und Fristen**

Dieser Abschnitt beinhaltet die allgemeinen Informationen, die vor allem zur Klärung der Formalien bei der Bearbeitung der Abschlussarbeit wichtig sind. [Req2.1.1]

- **Formulierung des Themas**

In dieser Sektion befinden sich zahlreiche Tipps und hilfreiches Wissen zur Formulierung des eigenen Themas und der zu verfassenden Aufgabenstellung.

- **Betreuung der Bachelorarbeit**

Informationen zur Betreuung von Bachelorarbeiten und was hinsichtlich des Engagements und der Vorgehensweise von den Studierenden erwartet wird, sind in diesem Abschnitt zu finden.

- **Aufbau und Struktur**

Dieser Abschnitt befasst sich näher mit dem Aufbau der Bachelorarbeit und stellt vor allem Informationen und Empfehlungen bereit, welche sich auf die Strukturierung und die Bedeutung der einzelnen Kapitel der Bachelorarbeit beziehen. [Req2.1.2]

- **Hinweise zum Schreiben**

In diesem Abschnitt werden tiefgehende Informationen und Empfehlungen behandelt, welche sich auf die praktische Umsetzung des Schreibens im Detail beziehen. Beispielhaft hierfür sind der Schreibstil, die Verwendung von Zeiten oder die Einbindung von Bildern, Tabellen und Programmcode. [Req2.1.3]

- **Methodenübersicht**

Dieser Abschnitt bietet eine Übersicht wichtiger Methoden der Informatik, die sich bei der Bearbeitung und Dokumentation der Bachelorarbeit als hilfreich erweisen könnten. [Req2.2]

Hinweis: Der Inhalt liegt im Rahmen der Prototypentwicklung jedoch nur angedeutet vor und wurde hierbei aus unterschiedlichen Quellen abgeleitet. [23] [4]

- **FAQ**

Dieses Kapitel behandelt oft gestellte Fragen und verdeutlicht die Informationen in der Form eines Frage-Antwort Schemas.

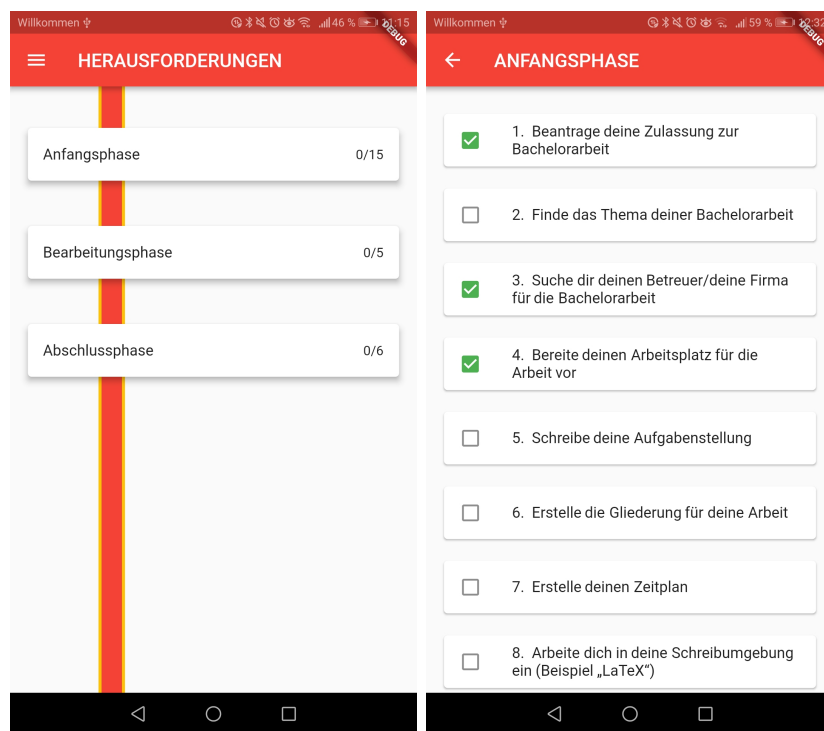
4.1.3. Herausforderungen

Die sogenannten Herausforderungen [Req3.1] sollen im Rahmen der Applikation, im Gegensatz zu den frei definierbaren Meilensteinen, eine fest definierte kategorisierte Aufteilung von typischen im Verlauf einer Bachelorarbeit zu leistenden Aufgaben darstellen [Req3.2].

Diese Aufgaben sollen vor allem als interaktiver Leitfaden dienen, an welchem sich die Studierenden orientieren können. Hierbei wird das Ziel angestrebt, den Studierenden eine Perspektive auf alle Phasen einer Bachelorarbeit sowie den jeweiligen einzelnen Arbeitsschritten zu geben, welche von den Studierenden abgehakt werden können.

Durch die in der Anforderungsanalyse ermittelten Daten hat sich der Eindruck ergeben, dass viele Studierende gerade zu Beginn der Bearbeitungszeit dazu neigen, die Arbeit, aufgrund der noch viel verbleibenden Zeit, aufzuschieben. Unter diesem Aspekt wird hierbei versucht, durch das Abhaken von kleinen konkreten Aufgaben bereits frühzeitig einen fließenden Arbeitsablauf (Workflow) seitens der Studierenden anzuregen.

Hinweis: Der Inhalt und Ablauf der dargestellten Herausforderungen ist im Rahmen der Prototypentwicklung lediglich beispielhaft erfolgt und stellt somit keinen korrekten und vollständigen Entwurf dar



(a) Herausforderungen - Roter Faden

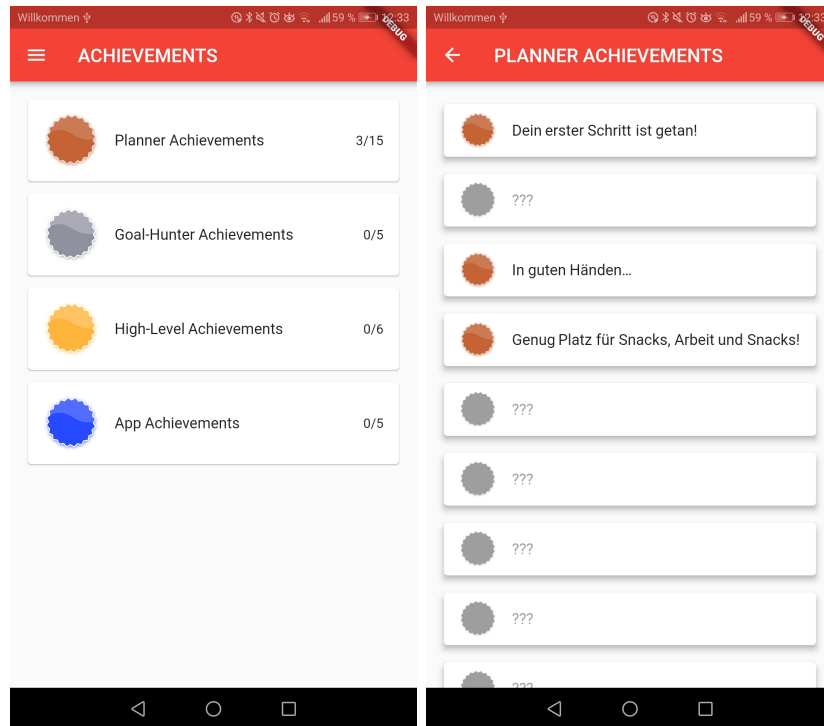
(b) Herausforderungen - Details

Abbildung 4.3.: Benutzeroberfläche Herausforderungen

4.1.4. Achievements

Der Achievement-Hauptbildschirm [Req3.1] enthält eine Übersicht der verschiedenen Kategorien von Achievements (siehe Abbildung 4.7). Die nächste Ebene dieses Abschnitts zeigt die detaillierte Ansicht der jeweiligen Achievements an, bei welcher der Benutzer einsehen kann, welche Achievements er bereits erhalten hat.

Hinweis: Weitere Informationen zu den einzelnen Achievements und den Abstufungen der jeweiligen Kategorien befinden sich in Kapitel 4.2.



(a) Achievements Hauptbildschirm (b) Beispiel: Planner Achievements

Abbildung 4.4.: Benutzeroberfläche Achievements

4.1.5. Sonstige Softwareinhalte

Über die eigentlichen individuellen Funktionalitäten hinaus sind weitere Grundfunktionalitäten vorhanden, welche unter den Punkt **Sonstige Softwareinhalte** fallen.

Es folgt eine stichpunktartige Auflistung und Beschreibung der Inhalte:

- **Menü**

Das Menü zeigt die existierenden Softwareabschnitte in einer klassisch gelisteten Menüstruktur (siehe Abbildung 4.5).

- **Einstellungen**

Die Einstellungen beinhalten Optionen, die sich auf die Eigenschaften der Applikation auswirken, beispielsweise das Verändern des Designs oder der Sprache^[Req4.1]

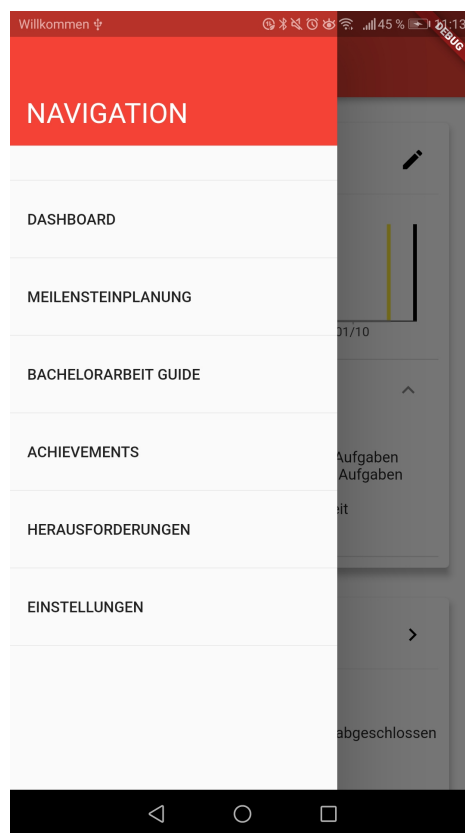


Abbildung 4.5.: Benutzeroberfläche Navigations-Menü

4.1.6. Dashboard

Das Dashboard stellt den Ausgangspunkt und Hauptbildschirm der Applikation dar und trägt somit die Aufgabe, aktuelle Informationen für den Nutzer aufbereitet anzuzeigen und die verschiedenen Funktionen der Applikation im Bund darzustellen.

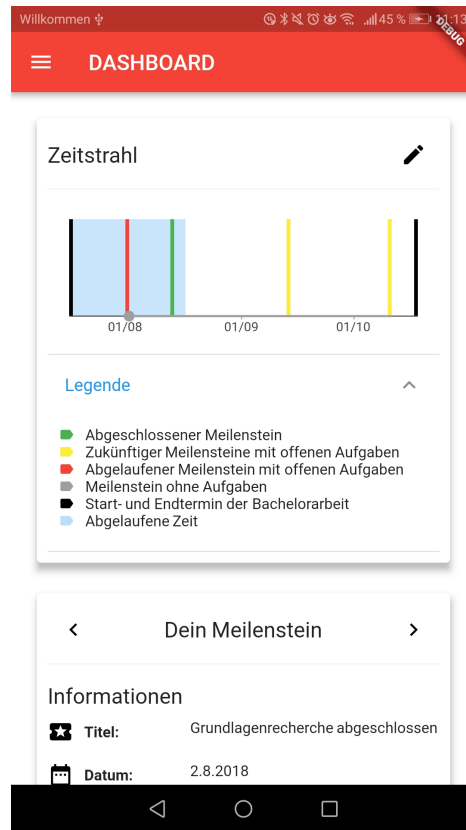


Abbildung 4.6.: Benutzeroberfläche Dashboard

Der Ausschnitt auf Abbildung 4.6 zeigt einen Zeitstrahl als primäres Fortschrittsverfolgungs-Tool. Dieser Zeitstrahl zeigt die von dem Nutzer erstellten Meilensteine sowie deren Status in Form einer spezifischen Farbvisualisierung. Die hierbei übermittelten Informationen ermöglichen dem Benutzer einen Überblick über abgelaufene (rot), aktive (gelb), erfüllte (grün) und statuslose (grau) Meilensteine.

Des Weiteren wird dem Benutzer durch einen blau hinterlegten Bereich die bereits abgelaufene Zeit verdeutlicht, welche zwischen Start- und Endpunkt der Bearbeitungszeit auf einfache Weise eine bessere Einschätzung von Fortschritts- und Zeitmanagement ermöglichen soll.

Auf dem Dashboard befindet sich auch ein Abschnitt, der den aktuell ausgewählten Meilenstein anzeigt. Dieser kann durch Interaktion mit dem Zeitstrahl oder mit den dafür vorgesehenen Pfeilen ausgewählt werden, so dass jederzeit Unteraufgaben abgehakt oder hinzugefügt werden können.

Es folgt eine stichpunktartige Auflistung und Beschreibung weiterer möglicher Ideen zu Inhalten des Dashboards, welche bei Planung der Software entwickelt wurden und durch einen frühen Oberflächenprototyp in die Anforderungsanalyse eingebunden wurde (siehe Anhang A.3):

- **Dynamische Anzeige des Fortschritts**

Der bisher geleistete Gesamtfortschritt wird verdeutlicht, indem angezeigt wird, wie viele Meilensteine in Relation zu den Gesamtmeilensteinen bisher abschlossen wurden. Weiterhin wird der bisher geleistete Wochenfortschritt verdeutlicht, indem zu sehen ist, wie viele Aufgabenpakete der Phase bis zum nächsten Meilenstein, in Relation zu den vorher definierten gesamten Arbeitspaketen, schon erfüllt wurden.

- **Anzeige der noch offenen Herausforderungen**

Es wird angezeigt, wie viele Herausforderungen der jeweiligen Kategorie abgeschlossen wurden, in Relation zu deren Gesamtanzahl.

- **Dynamische Anzeige der letzten erreichten Achievements**

Die Anzeige zeigt die letzten erreichten Achievements an. Zugehörig sind in diesem Fall die Darstellung des Typs des Achievements in Form der jeweiligen Färbung und des Titels der Herausforderung. Das zuletzt erreichte Achievement wird hierbei durch Größe der Darstellung und Beschreibung hervorgehoben.

- **Dynamische Anzeige von allgemeinen Tipps**

Die Anzeige zeigt zufällig ausgewählte allgemeine Tipps zur Bearbeitung der Bachelorarbeit an, die im Laufe der Zeit wechseln.

4.1.7. Übersicht der Produktfunktionen

Die unterschiedlichen Abschnitte der Software stehen unterschiedlich stark in Verbindung mit den verschiedenen definierten Produkthanforderungen. Um eine Übersicht über die groben Zusammenhänge zu ermöglichen, folgt eine zusammenfassende Tabelle (Siehe Tabelle 4.1).

Bezeichnung	Beschreibung	A1	A2	A3	A4	A5	A6
[Req1.1]	(Verwaltung von Meilensteinen)	●				●	
[Req1.2]	(Verwaltung von Aufgaben)	●				●	
[Req1.3]	(Erinnerung an Meilensteine)	●				●	
[Req2.1.1]	(Hinweise zu Rahmenbedingungen und Formalien)		●				
[Req2.1.2]	(Hinweise zu Aufbau und Struktur)		●				
[Req2.1.3]	(Hinweise zu Durchführung)		●				
[Req2.2]	(Übersicht von Methoden und Techniken)		●				
[Req2.3]	(Anpassbarkeit an Zielgruppe)		●				
[Req2.4]	(Software-externe Anpassung)		●				
[Req3.1]	(Benutzerklassen-orientierte Spiel-Design-Elemente)			●	●	○	
[Req3.2]	(Spiel-Design-Elemente sinnvoll an Bachelorarbeit orientieren)			●	●		
[Req4.1]	(Mehrsprachigkeit)						○
[Req4.2]	(Bachelorarbeitsthemen über Applikation ausschreiben)						○

- A1 - Meilensteinverwaltung
- A2 - Bachelorarbeit Guide
- A3 - Herausforderungen
- A4 - Achievements
- A5 - Dashboard
- A6 - Sonstige Softwareinhalte
- - Teil der Applikation
- - Als Erweiterung vorgesehen

Tabelle 4.1.: Abdeckung der Produktfunktionen

4.1.8. Umgang mit Risiken

Die Befürchtung [R1], dass die Applikation den Studierenden so viel Arbeit abnimmt, dass sie zu wenig eigenständige Erfahrung sammeln und somit der Lerneffekt zu gering ist, wird berücksichtigt, indem die Applikation einen eher informierenden Charakter aufweist, und keine Funktionalität bezüglich dem Liefern von Ergebnissen vorsieht.

Im folgenden Verlauf wird der Umgang mit den in Kapitel 3.6.4 dargestellten Risiken diskutiert und beschrieben.

Da sich die Herausforderungen an dem Ablauf einer Bachelorarbeit orientieren und den Studierenden somit vor typische Aufgaben stellen, kam es im Verlauf der Interviews zu den Befürchtungen [R2], dass die Applikation nicht den realen Zustand der Bachelorarbeit kennt und somit der Studierende durch das Erfüllen von Herausforderungen ein Gefühl von Erfolg hat, obwohl diese Herausforderungen nicht den geforderten Qualitäten entsprechen und der Studierende dies durch das positive Feedback nicht merkt.

Um diesem Risiko entgegenzuwirken, sollen die erstellten Herausforderungen möglichst auffordernder Natur sein (Beispiel: “Beginne deine Einleitung“). Bei Einsatz von Herausforderungen, welche auf inhaltliche Aspekte abzielen, sollen Checklisten zum Abschließen der Herausforderungen genutzt werden, um den Fokus auf eine allgemeingültige Vorgehensweise zu reduzieren, für deren Erfüllung die Studierenden jedoch selbst verantwortlich sind. Auch hier wird eine klare Kommunikation durch die Applikation als angebracht angesehen.

Die Gefahren der Risiken [R3] und [R4], die sich damit befassen, dass die Applikation fälschlicherweise als Ersatz eines Betreuers wahrgenommen oder der Inhalt der Applikation als unumstößlich angesehen werden könnte, soll durch eine klare Kommunikation mit den Anwendern reduziert werden. Es sollten als Maßnahmen einerseits immer die Quellen der Informationen angegeben und weiterhin eine allgemein sichtbare Erklärung zur Anwendung oder Fehlanwendung der Applikation deutlich kommuniziert werden.

Das identifizierte Risiko [R5], welches die Befürchtung einer Überplanung der Studierenden beinhaltet, was die Auswirkung haben kann, dass die Studierenden von der eigentlichen Arbeit abgehalten werden, wird als nicht ausschlaggebend betrachtet. Studierende, die zur Überplanung neigen, würden dies auch ohne Applikation tun.

Wichtiger wird in dieser Hinsicht die Freiheit der einzelnen Planungs-Bedürfnisse durch die individuelle Arbeitsweise der Studierenden angesehen, weshalb keine weiteren Maßnahmen zur Reduzierung der Planungsfreiheit getroffen werden. Weiterhin steht die Intensität der Planung nicht in der Verantwortung der Applikation, sondern in der Verantwortung des Benutzers.

Das Risiko [R6], das sich mit dem Nichtwirken der Gamification-Motivation befasst, wird als nicht weitreichend betrachtet, da die Gamification-Inhalte zwar Bestandteil der Applikation, jedoch nicht zwangsweise notwendig zur Benutzung der Applikation sind. Die Spiel-Design-Elemente sollen nicht das Interesse an einem bestimmten Thema steigern, sondern dienen im Kontext der Bachelorarbeit einer wegweisenden Natur, welche motivierende Einflüsse auf die Bearbeitenden hervorrufen können, jedoch nicht müssen, um ihren Nutzen zu erfüllen.

4.2. Beschreibung der Gamification-Elemente

Im folgenden Kapitel werden die Gamification-Elemente, die in der Applikation zum Einsatz kommen, näher beschrieben. Dabei wird vor allem auf die identifizierten Kategorien der Achievements eingegangen, welche den Kern der gewählten Motivationsstrategie bilden. Darüber hinaus wird auch beschrieben, in welcher Form die gewählte Motivationsstrategie in der Realität, in Bezug auf den Fortschritt bei der Bearbeitung der Bachelorarbeit, funktioniert.

Die Achievements verfolgen im Rahmen der Applikation die Aufgabe, motivierende Inhalte vor allem für die identifizierten Benutzergruppen der **Achiever** und **Explorer** (siehe 2.1.3) zu bieten. Dies soll erreicht werden, indem die Achievements dem Benutzer als Belohnung für eine abgeschlossene Herausforderung als Sammelstück hinzugefügt werden.

Im Rahmen der gewählten Motivationsstrategie werden verschiedenen Arten von Achievements in die Applikation integriert. Dies passiert unter anderem mit dem Ziel, eine an den Ablauf einer Bachelorarbeit gekoppelte Abstufung von Fortschritt und Erfolgen zu ermöglichen. Weiterhin werden auch die sogenannten *App-Achievements* Einzug in die Applikation erhalten. Diese sollen weitestgehend die Bedürfnisse der Benutzergruppe **Explorer** ansprechen und im Gegensatz zu den anderen Achievements nicht an einer konkreten Herausforderung gekoppelt sein, sondern im Laufe der Nutzung der Applikation durch *versteckte* Aufgaben, wie das Erstellen eines Meilensteins in der App, freigeschaltet und offen gelegt werden können.

Um den Achievements, über das Sammeln und die Vervollständigung hinaus, eine Wertigkeit zu geben, werden diese durch verschieden visualisierte Medaillen dargestellt.



Abbildung 4.7.: Visualisierung der Achievements

Hinweis: Diese Achievements basieren auf einem Bild aus der gemeinfreien Bilderdatenbank Pixabay (vgl. [31]) und wurden mit einer Bildbearbeitungssoftware für die eigenen Zwecke bearbeitet.

Es folgt eine Übersicht und Beschreibung der verschiedenen Achievement-Typen:

- **Anfangsphase-Achievements** (Bronze)

Beinhaltet Achievements, die an die Herausforderungen gekoppelt sind, die nach Beginn der Bachelorarbeit erfolgen, sich aber noch nicht mit den einzelnen inhaltlichen Schritten der Bachelorarbeit auseinandersetzen.

- **Bearbeitungsphase-Achievements** (Silber)

Beinhaltet Achievements, welche an die Herausforderungen gekoppelt sind, die nach Beginn der Bachelorarbeit erfolgen und sich mit der eigentlichen Bearbeitung der Bachelorarbeit auseinandersetzen.

- **Abschlussphase-Achievements** (Gold)

Beinhaltet Achievements, welche an die Herausforderungen gekoppelt sind, die sich hauptsächlich mit der Abschlussphase einer Bachelorarbeit beschäftigen.

- **App-Achievements** (Blau)

Beinhaltet Achievements, welche an keine Herausforderungen gekoppelt sind, sondern nur durch Benutzung der Applikation erhalten werden können. Die hierbei von den klassischen Abstufungen der Wertigkeit (Bronze, Silber, Gold) abweichend gewählte Farbe (Blau) soll die semantische Differenz zwischen den Herausforderungen und den durch die Benutzung der App ausgelösten Achievements verdeutlichen.

Der Inhalt wird durch unterschiedliche humorvolle oder auflockernde Texte dargestellt. Hierbei geht es darum, den mit den Herausforderungen gekoppelten Achievements durch das Verfassen passender Texte einen interaktiven Charakter zu verleihen. Bei Erreichen des Achievements wird dieser Inhalt angezeigt.

Hinweis: Eine für den Prototypen gewählte Übersicht der Achievements ist im Anhang zu finden (siehe Anhang A.2).

5. Architektur und prototypische Implementierungsdetails

Das Ziel dieses Kapitels ist es, das Verständnis für die architekturspezifischen Zusammenhänge sowie die jeweiligen software-internen Abläufe zu vermitteln. Auf dieser Basis werden die einzelnen Architektur-Komponenten, deren Beziehungen zueinander sowie die Funktionsweise des Gesamtsystems erläutert.

Aufgrund der Vorzüge einer konsequenten zentralen Verwaltung des Zustands der Applikation, welche auch mit einer komfortablen Wartbarkeit und den strikt getrennten und somit auch eindeutig nachverfolgbaren Datenflüssen und Operationen einhergeht, wurde eine Architektur konstruiert, die sich an dem sogenannten Redux-Architekturmuster orientiert.

Was eine Redux-Architektur genau ist und wodurch sich das Architekturmuster auszeichnet, wird einleitend beschrieben und führt nachfolgend über die eigentliche Architekturbeschreibung der entwickelten Software bis hin zu den wichtigsten Implementierungsdetails, welche an einem durchgängigen beispielhaften Ablauf einer Nutzeraktion beschrieben werden.

5.1. Was ist Redux?

Redux ist eine von Dan Abramovic und Andrew Clark entwickelte, im Jahr 2015 veröffentlichte Open-Source Bibliothek für JavaScript-Anwendungen [35]. Durch die zentrale Verwaltung von Zustandsinformationen soll eine erleichterte Pflege der Konsistenz und aufgrund dessen auch eine “einfache” Wartbarkeit und Testbarkeit der Software ermöglicht werden. [32]

*Hinweis: Wie auch in [32] angemerkt ist, wird an dieser Stelle darauf hingewiesen, dass es sich hierbei nicht um das WordPress-Framework **Redux Framework**[34] handelt.*

Innerhalb einer Redux-Architektur existieren verschiedene Komponenten, welche jeweils einen eigenen Zweck erfüllen und durch das modulare Zusammenspiel mit anderen Komponenten in dem System eine Gesamtkomposition bilden. Die hierbei wichtigsten Redux-spezifischen Komponenten sind **Actions**, **Reducers** und der **Store**, welche hinsichtlich Bedeutung und Funktionsweise in den nachfolgenden Kapiteln beschrieben werden.

5.1.1. Actions

Wie in den Basics der Online-Dokumentation [1] beschrieben wird, definieren Actions im Rahmen von Redux hauptsächlich Informationspakete, welche unter dieser Bedeutung auch als “Payload“ bezeichnet werden. Sie enthalten in erster Linie nur Informationen über die Action-spezifische Operation, wie beispielsweise das Ändern bestimmter Bestandteile des aktuellen Systemzustands.

Hinweis: Als Beispiel könnte denkbar sein, dass ein neuer Eintrag einer To-Do-Liste angelegt werden soll. Hierbei würde die Action beispielsweise einen Namen wie “AddTodoAction“ tragen und die Informationen des Titels des neuen Eintrags enthalten.

Das Ausführen dieser Zustandsänderung wird also nicht von den Actions selbst ausgeführt, für diesen Zweck dienen die sogenannten Reducer-Funktionen.

5.1.2. Reducers

Reducer-Funktionen erfüllen den Zweck, die durch die jeweilige Action übermittelten Informationen zu verarbeiten und die eigentliche, durch die Action spezifizierte, Zustandsänderung durchzusetzen. [33]

Reducer liegen im Allgemeinen als Pure Functions (auch Reine Funktionen) vor. Dies bedeutet im Rahmen der Redux-Architektur, dass die Funktionen keine direkten Änderungen an den durch die Actions übermittelten Informationen oder an dem Status selbst vornehmen dürfen und unter keinen Umständen Seiteneffekte produzieren dürfen. Die Auswirkungen von Pure Functions müssen deterministisch bestimmbar sein, also muss unter den gleichen Eingabebedingungen immer das gleiche zu erwartende Ergebnis folgen. Im Detail werden hierbei keine direkten Änderungen an Objekten vorgenommen, sondern der gesamte aktuelle Zustand durch Neugenerierung überschrieben.

Hinweis: Eine nicht den Eigenschaften von Pure Functions entsprechende Funktion wäre das Bestimmen des Datums mit der JavaScript-Funktion `Date.now()`, da hier bei Ausführung immer ein unterschiedliches Ergebnis herauskommt.

5.1.3. Store

Der Store ist in der Redux-Architektur das zentrale Element, welches nicht nur für die Datenhaltung des aktuellen Zustands (State) der Applikation zuständig ist, sondern auch als Schnittstelle für die Bereitstellung und Veränderung des aktuellen Systemzustands dient. [40] Im Gegensatz zu den anderen Komponenten ist der Store innerhalb der Architektur einmalig, was viele Vorteile der Redux-Architektur überhaupt erst ermöglicht.

Hinweis: Der Store enthält alle für die Applikation relevanten Informationen an zentraler Stelle im System in einem State-Objekt.

Die gespeicherten Informationen liegen im besten Fall in normalisierter Form vor. Insgesamt ist ein flacher Key-Value Store dem Handhaben komplexer Objekte vorzuziehen

5.1.4. Flutter Redux Library

Die externe Flutter Library `flutter_redux`^[15] setzt die auf JavaScript basierende Redux-Funktionsweise für die Entwicklung mobiler Applikationen mit Flutter und Dart um, wodurch die beschriebene komfortable Nutzung der Redux-Architektur ermöglicht wird.

Grundsätzlich werden also dieselben Funktionsweisen durch die Nutzung der Flutter Redux Library abgebildet. Es folgt dennoch eine kurze Beschreibung der spezifischen Redux Widgets, welche sich jedoch alle auf die Interaktion mit dem zentralen Store beziehen:

- **StoreProvider**

Der StoreProvider ist ein Schnittstellen-Widget, das allen unterliegenden Widgets Zugang zum Store ermöglicht.

- **StoreBuilder**

Der StoreBuilder holt sich den aktuellen Zustand von dem in der überliegenden Hierarchie am nächsten platzierten StoreProvider und bietet eine builder-Funktion um die fallspezifischen Informationen zu verwenden.

- **StoreConnector**

Der StoreConnector trägt die Aufgabe, den von dem StoreProvider übergebenen Store durch eine converter-Funktion in ein ViewModel umzuwandeln, welches die benötigten fallspezifischen Informationen in sich hält und durch die builder-Funktion diese Informationen kontextspezifisch verwendet. Weiterhin wird der StoreConnector bei jeder Änderung des Zustands neu generiert und somit die durch das ViewModel repräsentierten Informationen aktualisiert.

5.2. Architekturbeschreibung

Im folgenden Kapitel werden wichtige Details zur Architektur hinsichtlich des Datenflusses innerhalb der Software sowie der zusammenhängende Aufbau der Model-Komponenten anhand eines UML-Klassendiagramms beschrieben.

5.2.1. Übersicht des Datenflusses

Die folgende Abbildung 5.1 soll in vereinfachter Form verdeutlichen, wie die verschiedenen Komponenten der gewählten Redux-Architektur hinsichtlich des Aufbaus und des Datenflusses innerhalb der entwickelten Software zueinander in Beziehung stehen.

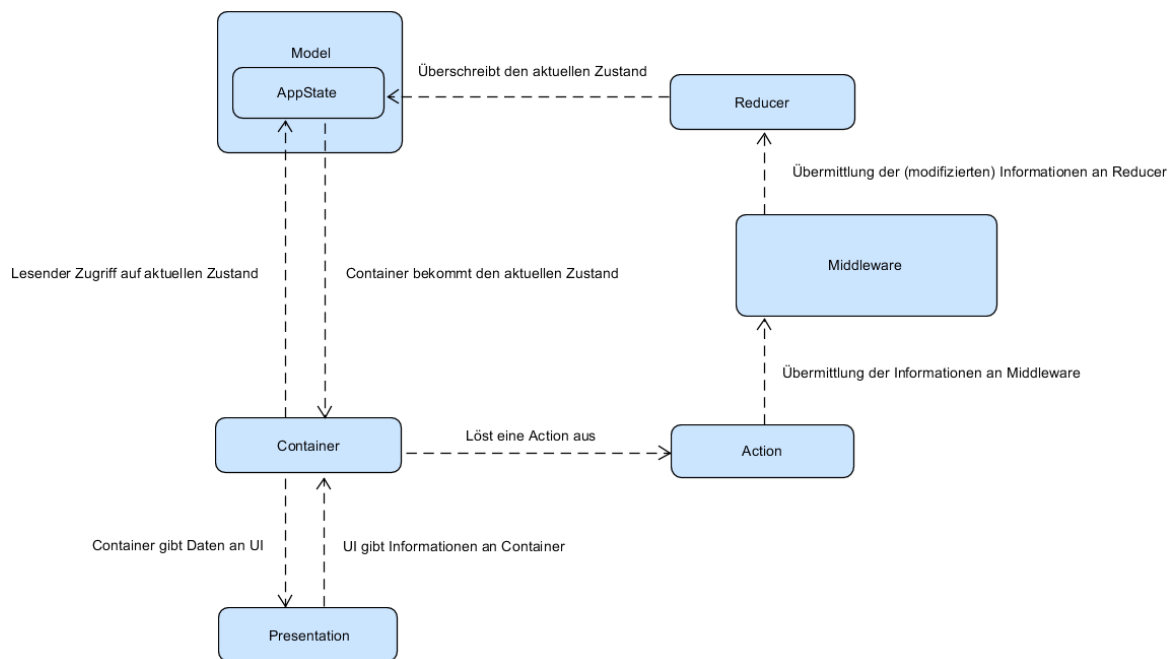


Abbildung 5.1.: Datenflussübersicht der Softwarearchitektur

Die Model-Komponenten stellen den aktuellen internen Zustand des Systems dar und enthalten somit in erster Linie die hierfür verwendeten Objekt-Klassen. Die hierbei für die gewählte Architektur durch ihre Wichtigkeit hervorzuhebende Klasse **AppState**, ist die zentrale Verwaltungsklasse für den gespeicherten Status der Applikation und trägt alle relevanten Informationen in sich, die den aktuellen Zustand des Systems abbilden. Diese Klasse dient also als zentrales Element zur Datenhaltung.

Die Presentation-Komponenten sind allein für die Visualisierung der gegebenen Informationen sowie für die Entgegennahme der Benutzereingaben zuständig, stehen jedoch im engen Austausch mit den Container-Komponenten, welche den aktuellen Zustand der Applikation aus dem **AppState** einlesen und die erhaltenen Informationen an die Presentation-Komponenten

übermitteln. Da nicht immer alle Informationen für die jeweilige Benutzeroberfläche relevant sind, werden diese vor der Übermittlung hinsichtlich der benötigten Inhalte gefiltert und vorbereitet. Weiterhin lösen die Container-Komponenten im Allgemeinen die Actions aus, welche auf den von dem Benutzer getätigten Interaktionen zur Änderung des Systemstatus, wie beispielsweise das Hinzufügen eines Meilensteins, basieren.

Wie bereits in Kapitel 5.1 beschrieben, enthält eine Action-Komponente Informationen über die jeweilige auszuführende Operation, welche von der Middleware und/oder den Reducer-Komponenten aufgegriffen und verarbeitet werden. Das Verarbeiten der Informationen innerhalb der Middleware wird in erster Linie bei Vorgängen gehandhabt, welche sich als zu komplex für die einfach zu haltenden Reducer-Komponenten darstellen. Ein beispielhafter Vorgang wäre hierbei das Überprüfen von Achievements und das zugehörige Auslösen von Notifications. Weiterhin können auch in der Middleware selbst Actions ausgelöst werden, wie anhand des gezeigten beispielhaften Ablaufs in Kapitel 5.3 nachzuvollziehen ist.

Für die eigentliche Änderung des Systemzustands sind jedoch, unabhängig von den in der Middleware getätigten Verarbeitungsschritten immer die Reducer-Komponenten. Diese “wissen“ durch die über die Actions erhaltenen Informationen, welche Aktion der Benutzer ausgeführt hat und folglich auch, welche Information sich an dem aktuell existierenden Zustand der Applikation ändern soll.

5.2.2. Verwendet externe Bibliotheken

Bei der Entwicklung der Software wurden die folgenden externen Bibliotheken verwendet, welche bisher noch nicht genannt wurden:

- **path_provider** [29]
Plugin zur Ermittlung des zielplattform-spezifischen Speicherorts von applikationsbezogenen Daten. Wurde verwendet zur Speicherung von eingelesenen JSON-Dateien aus dem Git-Repository.
- **charts_flutter** [12]
Plugin um Diagramme zu gestalten und anzuzeigen. Wurde verwendet zur Visualisierung des Zeitstrahls auf dem Dashboard.
- **redux_persist** [36]
Middleware zum Speichern und Laden des aktuellen Zustands bei Öffnen und Beenden der Applikation.
- **json_serializable** [22]
Einfache JSON-Serialisierungsmöglichkeiten von komplexen Klassenstrukturen durch Generierung von Code. Wird in Flutter-Dokumentation referenziert [21]. Wurde verwendet zur Serialisierung komplexer Klassenstrukturen (Beispiel: **Milestone** und **Task**).
- **flutter_local_notifications** [13]
Plugin, das das Auslösen von Push-Notifications ermöglicht. Wurde verwendet für das Auslösen von Push-Notification bei Erlangen von Achievements.

5.2.3. Architekturübersicht der Model-Komponenten

Wie in Abbildung 5.2 zu sehen ist, besitzt die bereits angesprochene Klasse **AppState** den gesamten internen Systemzustand größtenteils in Form eines flach ausgerichteten Key-Value Stores. Während die jeweiligen Collections, welche die Objekt-Klassen **Property**, **Achievement**, **Challenge** und **Content** beinhalten, konsequent in einer flachen Hierarchie angelegt sind, weist die Verwaltung von Meilensteinen (**Milestone**-Klasse), inklusive den verschiedenen Unteraufgaben (**Task**-Klasse), eine komplexere Struktur auf. Dies hat sich im Laufe des Entwicklungsprozesses durch die stark zusammenhängende Verbindung der beiden Klassen als weniger aufwändige Alternative zur Verwaltung des Zustands und der Handhabung der diesen Klassen zuzuordnenden **Actions** ergeben.

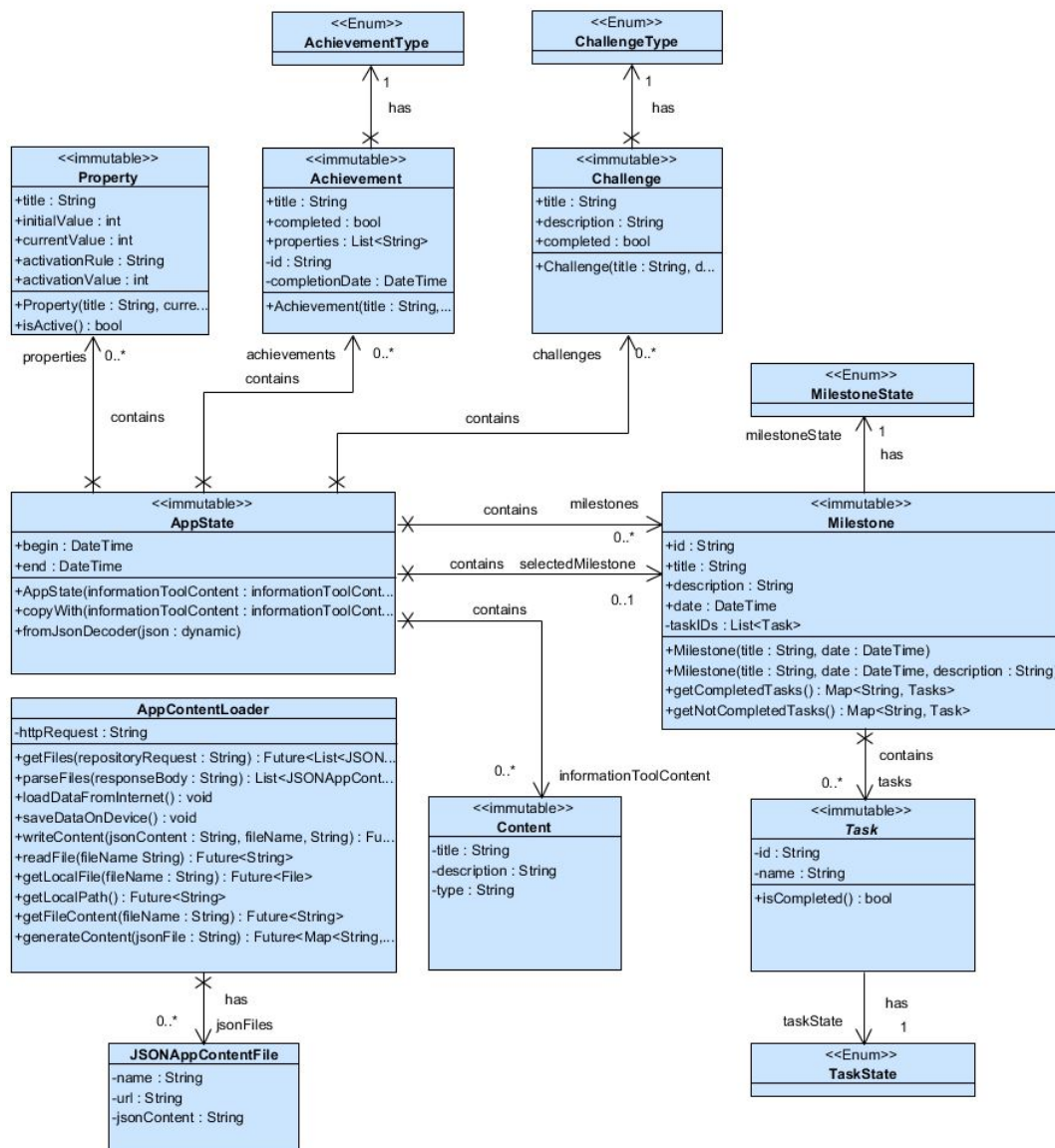


Abbildung 5.2.: Architektur der Model-Komponenten

Beschreibung der Model-Komponenten

Objekte der Klasse **Achievement** stellen die von dem Anwender zu erreichenden Achievements in der Applikation dar und enthalten Referenzen auf die zugehörigen Erfüllbedingungen (**properties**) in Form von textuell angelegten IDs. Ein Achievement kann mehrere Properties enthalten, welche alle erfüllt werden müssen, damit das Achievement erreicht wird. Die in der Applikation eng mit den Achievements verbundenen Herausforderungen werden durch die Klasse **Challenge** dargestellt. Diese erhalten einen Titel, der die Aufgabe darstellt sowie eine Beschreibung. Die Initialisierung der genannten Achievement- und Challenge-Inhalte findet in einer separaten Dart-Datei in Form von statisch definierten Werten statt und wird bei erstmaligem Start der Applikation eingelesen.

Die Content-Objekte stellen den textuellen Inhalt des Bachelor-Guides dar, wobei die Klasse **AppContentLoader** hierbei als Hilfsklasse dient, um die in dem GitRepository liegenden JSON-Dateien einzulesen und den somit heruntergeladenen Inhalt auf dem lokalen Datenträger des Endgeräts zu speichern. Auf Grundlage dieser gespeicherten JSON-Dateien werden die Content-Objekte dann durch die Klasse **AppContentLoader** generiert. Ein Auslesen des Git-Repositories erfolgt nach jedem Neustart der Applikation. Sollte die Verbindung zum Repository fehlschlagen, so wird der vorhandene bereits gespeicherte Inhalt verwendet, also muss zumindest zu einem Zeitpunkt eine korrekte Durchführung des Ablaufs stattfinden, damit die Inhalte angezeigt werden können. Die Struktur der im Git-Repository liegenden JSON-Datei folgt dem in Listing 5.1 angedeuteten Muster:

```
1  [
2    {
3      "id"      : "ROOT_1",
4      "type"    : "ButtonListWidget",
5      "title"   : "Root-Titel",
6      "description" : "",
7      "subsections" : ["SUB_1_1", "SUB_1_2"]
8    },
9    {
10     "id"      : "SUB_1_1",
11     "type"    : "ExpansionPanelWidget",
12     "title"   : "SubTitle_1",
13     "description" : "Beschreibung_1",
14     "subsections" : []
15   },
16   {
17     "id"      : "SUB_1_2",
18     "type"    : "ExpansionPanelWidget",
19     "title"   : "SubTitle_2",
20     "description" : "Beschreibung_2",
21     "subsections" : []
22   }
23 ]
```

Listing 5.1: Struktur: Inhalt der JSON-Datei

Die Klasse `Milestone` stellt im Rahmen der Applikation die zu erstellenden Meilensteine, inklusive den enthaltenen Unteraufgaben (`Task`-Klasse) dar. Wie bereits erwähnt, werden diese nicht, wie der Rest der Zustandsinformationen, flach ausgerichtet im `AppState` gehalten, sondern werden direkt referenziert.

Die detaillierten Beschreibungen der Klassen lassen sich weiterhin unter dem Anhang A.4 vorfinden, wurden jedoch aus Gründen der Übersicht nicht in die Arbeit übernommen.

5.3. Beispielhafter Ablauf einer Nutzeraktion

Im folgenden Kapitel, wird der typische Ablauf einer Nutzeraktion innerhalb der Redux-Architektur an einem fortlaufenden Szenario erläutert. Die hierbei gewählte Beispielaktion umfasst aufgrund der Einfachheit und der gleichzeitigen Repräsentanz des Gesamtsystems das Bearbeiten eines bereits vorhandenen Meilensteins.

Ausgangspunkt des Szenarios

Der Nutzer hat bereits einen Meilenstein angelegt, welcher einen Titel, ein Datum und eine Beschreibung enthält. Der Nutzer entdeckt nun, dass er den angegebenen Meilenstein aufgrund eines Rechtschreibfehlers falsch benannt hat und öffnet das hierfür vorgesehene Fenster zur Bearbeitung eines Meilensteins, woraufhin die zugehörige Container-Komponente die hierfür relevanten Informationen aus der `AppState` Klasse bezieht und an die Presentation-Komponente übergibt und diese dann den vorhandenen Inhalt anzeigt.

EditMilestone-Container

Die Container-Komponente funktioniert durch das Zusammenspiel zweier Klassen, welche zum einen die eigentliche Container-Klasse `EditMilestone` (vgl. Listing 5.2) und zum anderen die zugehörige Klasse `_ViewModel` (vgl. Listing 5.2) darstellt.

Die Converter-Funktion, welche in Listing 5.2 in Zeile 7 zu sehen ist, konvertiert den aktuellen Store in ein `ViewModel`, welches die für die Presentation-Komponente relevanten Daten in der Builder-Funktion in Zeile 10 bereitstellt und diese erzeugt.

```
1 class EditMilestone extends StatelessWidget {
2   final Milestone milestone;
3   EditMilestone(this.milestone);
4
5   Widget build(BuildContext context) {
6     return new StoreConnector(
7       converter: (Store<AppState> store) {
8         return _ViewModel.fromStore(store, this.milestone, context);
9       },
10      builder: (BuildContext context, _ViewModel vm){
11        return AddEditMilestoneScreen(vm.editMilestone, vm.currentMilestones,
12          vm.initialDate, true, milestone);
13      },
14    );
15  }
```

Listing 5.2: Container-Komponente einer Meilensteinänderung

Wie in Listing 5.3 zu sehen ist, werden in Zeile 20 die Informationen aus dem Store geladen und in Zeile 15 die getätigte Action ausgelöst. Letzteres funktioniert durch das Übermitteln einer Callback-Funktion an die Presentation-Komponente (siehe Listing 5.2, Zeile 11 `vm.editMilestone`).

```
1 class _ViewModel {
2     final Milestone currentMilestone;
3     final Function(String, DateTime, String) editMilestone;
4     final List<Milestone> currentMilestones;
5     List<String> currentMilestonesDates;
6     DateTime initialDate;
7
8     _ViewModel(
9         {this.currentMilestone, this.editMilestone, this.currentMilestones,
10          this.currentMilestonesDates}) {
11         // ... Initialisieren von currentMilestonesDates und initialDate
12     }
13
14     factory _ViewModel.fromStore(Store<AppState> store, currentMilestone, context) {
15         return _ViewModel(
16             editMilestone: (String title, DateTime date, String description) {
17                 store.dispatch(new EditMilestoneAction(
18                     currentMilestone, title, date, description));
19                 checkAchievementNotification(context);
20             },
21             currentMilestones: store.state.currentMilestones.values.toList(),
22         );
23     }
```

Listing 5.3: ViewModel von Container-Komponente einer Meilensteinänderung

EditMilestone-Action

Wie in Listing 5.4 zu sehen ist, enthält die in Listing 5.3 Zeile 15 ausgelöste Action `EditMilestoneAction` vier verschiedene Datenfelder, welche den zu ändernden Meilenstein (`milestone`) sowie die neuen Werte `newTitle`, `newDate` und `newDescription`, welche die zu dem vorhandenen Meilenstein zugehörigen Werte überschreiben sollen.

```
1 class EditMilestoneAction {
2     Milestone milestone;
3     String newTitle;
4     DateTime newDate;
5     String newDescription;
6
7     EditMilestoneAction(this.milestone, this.newTitle, this.newDate,
8         this.newDescription);
9 }
```

Listing 5.4: Meilenstein-Action Beispiel

EditMilestoneScreen-Presentation

Bevor die beschriebene Action jedoch gesendet werden kann, muss der Benutzer diese erst einmal auslösen. Wie in Listing 5.5 zu sehen ist, geschieht dies durch das Bestätigen der vom Nutzer eingegebenen Inhalte (vgl. Zeile 4). Darunter zählen in diesem Fall der Titel, das Datum und eine optionale Beschreibung des bearbeiteten Meilensteins.

```
1 void submit(context) {
2
3     if (_formKey.currentState.validate()) {
4         addEditMilestone(_titleInputField.currentState.value, chosenDate,
5             _optionalDescriptionInputField.currentState.value);
6
7         ...
8     }
9 }
10 }
```

Listing 5.5: Methode: Bestätigung einer Meilensteinänderung

AchievementNotification-Middleware

Bei Ausführung des in Listing 5.3, Zeile 18 befindlichen Aufrufs der Top-Level Funktion `checkAchievementNotification()` (siehe Listing 5.6) wird überprüft, ob es nach der Nutzeraktion zu dem Erreichen von Achievements gekommen ist. Der Aufruf dieser Methode muss somit innerhalb der ViewModel-Klassen der Container-Komponenten erfolgen, sofern die jeweilige Action eine für die definierten App-Achievements (vgl. Kapitel 4.2) relevante Bedeutung hat.

Um zu gewährleisten, dass neue Achievements registriert werden, wird vor Ausführung der `SendAchievementNotificationAction` in Zeile 3 - 7 die Action `CheckForAchieveAction()` ausgelöst, welche für das Überprüfen der Achievements hinsichtlich deren zuvor in den Reducer-Funktionen aktualisierten (vgl. Listing 5.8, Zeile 1 - 9) Aktivierungsbedingungen sorgt.

```
1
2 Future<dynamic> checkAchievementNotification(context) {
3     StoreProvider.of<AppState>(context).dispatch(new CheckForAchieveAction(
4         StoreProvider
5             .of<AppState>(context)
6             .state
7             .achievedAchievements[ALL_ACHIEVEMENTS]));
8
9     final action = new SendAchievementNotificationAction();
10    StoreProvider.of<AppState>(context).dispatch(action);
11    return action.completer.future;
12 }
```

Listing 5.6: Methode: `checkAchievementNotification`

Wie in Listing 5.7 zu sehen ist, wird hier eine Abfrage zu der Art der ausgelösten Actions getätigt (vgl. Zeile 2). Sollte es sich hierbei nicht um eine `SendAchievementNotificationAction` handeln, so wird die eingegangene Action sofort weiter an die Reducer-Komponenten geleitet (vgl. Zeile 13). Falls sich die Bedingung jedoch als wahr herausstellt, so wird erst der darauf folgende Ablauf von Befehlen ausgeführt, welcher sich hier auf das Auslösen einer Notification bezieht. Eine Besonderheit innerhalb dieser Komponente ist die Verwendung eines Completers, welcher als Teil der Action Payload übermittelt und bereits in Listing 5.6 “vorbereitet” wurde.

Hinweis: Ein sogenannter Completer ist eine von Dart gegebene Klasse, die es in diesem Fall ermöglicht, eine zukünftig ausgeführte Funktion so zu behandeln, als wäre der Rückgabewert bereits bekannt. Weitere Informationen hierzu sind in der Dart Dokumentation zu finden. [8]

```
1 void notificationMiddleware(Store<AppState> store, dynamic action, NextDispatcher
   next) {
2   if (action is SendAchievementNotificationAction) {
3     new Future.delayed(new Duration(seconds: 1), () {}).then( (result) {
4       if (NEUE ACHIEVEMENTS VORHANDEN) {
5         showAchievementNotification(
6           //... spezifischer Inhalt eines Achievements
7         );
8         action.completer.complete();
9         store.dispatch(new ClearAchievedAction());
10      },
11    );
12  }
13  next(action);
14 }
```

Listing 5.7: Notification Middleware

EditMilestone-Reducer

Nach dem Senden der `EditMilestoneAction` gelangt diese letztendlich an den App-Reducer, welcher wiederum die Top-Level-Funktion weiterer Reducer-Funktionen ist und nach jeder durchgeführten Action den alten Zustand der Applikation überschreibt. Jedes Datenfeld des aktuellen Zustands wird in diesem Rahmen durch die Delegation an untergeordnete Reducer-Funktionen neu generiert. Dabei wird immer der aktuelle Zustand des Systems sowie die in der Action-Klasse enthaltenen Informationen übergeben. In diesem Fall betrifft dies die Methode `updateMilestones` (siehe Listing 5.8) in der überprüft wird, um welche Action es sich handelt. Sollte es sich um keine der explizit abgefragten Actions handeln, so wird der aktuell existierende Zustand unverändert zurückgegeben (vgl. Zeile 20). Sollte die Action jedoch wie in diesem Fall übereinstimmen, so wird anschließend die jeweils definierte Folge von Befehlen ausgeführt und der neue Zustand der jeweiligen Daten zurückgegeben (vgl. Zeile 17).

```
1 Map<String, Property> updateProperties(Map<String, Property> current, action) {
2 ...
3   if (action is EditMilestoneAction) {
4     current = setPropertyValue(current, propApp1.name, 1);
5     return current;
6   }
7 ...
8   return current;
9 }
10
11 Map<String, Milestone> updateMilestones (Map<String, Milestone> current, action){
12 ...
13   if(action is EditMilestoneAction) {
14     Milestone editedMilestone = new Milestone(action.newTitle, action.newDate,
15       action.newDate.toIso8601String(), action.milestone.tasks,
16       action.newDescription);
17     current.remove(action.milestone.id);
18     current[editedMilestone.id] = editedMilestone;
19     return current;
20   }
21 ...
22   return current;
23 }
```

Listing 5.8: Reducer Beispiel

6. Evaluation

In den folgenden Kapiteln werden die unternommenen Evaluationsschritte des Projekts beschrieben und ausgewertet. Hierbei werden die entwickelten Softwaretests, die durchgeführte Nutzerevaluation sowie die abschließende Bewertung des verwendeten Frameworks Flutter aufgeführt.

6.1. Ergebnisse der Softwaretests

Im Rahmen der Softwaretests wird speziell das Testen der Reducer-Funktionen im Kontext der Redux-Architektur als essenziell betrachtet, da diese als zentrale ausführende Einheit der jeweiligen Nutzeraktionen identifiziert werden und somit auch auf die jeweiligen Anforderungen abzubilden sind.

Es wurden verschiedene Interaktionen mit dem System identifiziert, welche durch die folgenden Testfälle im Rahmen der Reducer-Tests abgedeckt werden:

Testgruppe	Testfall	Status	Anforderungen
Milestone Reducer	Add Milestone	Erfolgreich	[Req1.1.1]
	Remove Milestone	Erfolgreich	[Req1.1.2]
	Edit Milestone	Erfolgreich	[Req1.1.3]
	Add Task to Milestone	Erfolgreich	[Req1.2.1]
	Remove Task from Milestone	Erfolgreich	[Req1.2.2]
	Edit Task from Milestone	Erfolgreich	[Req1.2.3]
Challenge Reducers	Check Toggle Challenge State	Erfolgreich	–
Achievement Reducers	Check Challenge Achievement	Erfolgreich	–
Initialization Test	AppContentLoader/Generator	Erfolgreich	[Req2.4]
	GitRequest Test		

Tabelle 6.1.: Übersicht der Reducer-Tests

6.2. Ergebnisse der Nutzerevaluation

Insgesamt wurden 6 Personen einem Test unterzogen, der als zentralen Bestandteil die Nutzung der entwickelten Applikation beinhaltet.

Unter den genannten 6 Personen befinden sich 2 Professoren und 4 Studierende des Studiengangs Informatik/Softwareentwicklung der Fachhochschule Lübeck.

Weiterhin befinden sich 3 der 4 Studierenden in der Phase während der Bachelorarbeit und 1 Studierender am Ende des 5. Semesters vor der Bearbeitung der Bachelorarbeit.

Der verwendete Testablauf setzt sich aus 11 nacheinander gestellten Aufgaben zusammen. Die hierbei gewählten Aufgaben orientieren sich an den verschiedenen Softwarefunktionen der entwickelten Applikation und sehen somit das Verwenden aller Kernfunktionen vor.

Die Reihenfolge der gestellten Aufgaben ist im Rahmen des Tests jedoch nicht fest definiert, sondern wurde je nach Situation frei gewählt.

Jeder Test wurde unter Moderation dieser Aufgaben durchgeführt, wobei hier aufgrund der vergleichbar zu behandelnden getesteten Nutzererfahrung sowie der abschließenden Beurteilung der Tests nicht zwischen Professoren oder Studierenden unterschieden, sondern der jeweils gleiche Testablauf gewählt wurde.

Bei der Durchführung der Tests gab es dennoch einige wichtige Unterschiede.

Zum einen unterlagen die Tests mit den Professoren einer ausführlicher gestalteten Vorstellung und Beschreibung der Features, da diese durch die Erfahrung der Professoren hinsichtlich ihres Nutzens als gut beurteilbar angenommen wurden.

Weiterhin wurde bei den Testdurchführungen mit den Studierenden ein abschließender Fragebogen ausgehändigt, in dem die Testpersonen noch einmal die Möglichkeit hatten weitere Anmerkungen zu der Software festzuhalten.

Dieser Fragebogen sowie der Aufgabenkatalog liegen im Anhang A.5.1 vor.

Der Test wurde auf dem eigenen, dafür vorbereiteten, Smartphone durchgeführt, wobei das Bildschirmgeschehen aller Tests mithilfe einer Software aufgezeichnet wurde, jedoch ist eine der Dateien durch einen unbekannten Datenfehler nicht korrekt aufgezeichnet worden und somit nicht verwendbar.

Des Weiteren wurden nach vorher eingeholter Erlaubnis der Testpersonen ein Audiomitschnitt erzeugt. Dieser Audiomitschnitt wurde von beiden Professoren und insgesamt 3 von 4 Studierenden zugelassen. Es wurden unmittelbar nach jedem Interview Notizen angefertigt, welche im Anhang dieses Dokuments zu finden sind (siehe Anhang A.5.2 und A.5.3).

6.2.1. Zusammenfassung der Testdurchführung mit den Studierenden

Zusammenfassend hat sich ergeben, dass alle Testpersonen die vorgegebenen Aufgaben fast ohne Schwierigkeiten lösen konnten und der Großteil der Aufgaben ohne Hilfestellung des Moderators erfüllt werden konnten. Falls Unklarheiten bezüglich des Lösens der Aufgaben bei den Testpersonen aufgetreten sind, so konnten diese größtenteils selbstständig beseitigt werden.

Im folgenden Verlauf werden die Ergebnisse des Tests, nach dem Ablauf der jeweiligen Aufgaben kategorisiert, dargestellt. Weiterhin werden, falls vorhanden, die Kernaussagen der aufgabenspezifischen Kommentare und Wünsche zur Softwaregestaltung und den Features der Testpersonen beschrieben.

Aufgabe 1 - Anweisungen des Startbildschirms befolgen

Insgesamt haben 4 von 4 Testpersonen diese Aufgabe erfolgreich abgeschlossen. Alle Studierenden konnten den Anweisungen folgen und haben einen Bearbeitungszeitraum für ihre Bachelorarbeit eingetragen und bestätigt, woraufhin sie auf das Dashboard gelangt sind. Es gab keine negativen Anmerkungen zur Softwareumsetzung dieser Funktionalität.

Aufgabe 2 - Beschreibe die Elemente des Dashboards

Alle 4 Testpersonen konnten die auf dem Dashboard befindlichen Elemente, also deren Bedeutung und Nutzen korrekt deuten.

Aufgabe 3 - Informiere dich über die Anmeldung der Bachelorarbeit.

Es konnten 3 von 4 Testpersonen ohne Schwierigkeiten Informationen über die Anmeldung einer Bachelorarbeit durch die Nutzung der Applikation gewinnen.

Eine der Testpersonen hatte große Schwierigkeiten, die Sektion "Bachelorarbeit Guide" als solche zu identifizieren und musste durch Hilfestellung des Moderators geleitet werden. Ab diesem Punkt gelang es der Testperson jedoch problemlos, sich über das Thema zu informieren.

Aufgabe 4 - Erstelle zwei Meilensteine

Alle 4 von 4 Testpersonen konnten diese Aufgabe erfolgreich lösen, wobei jedoch 2 von 4 Personen fälschlicherweise als erste Reaktion auf den Dashboard-Zeitstrahl getippt haben, da dieser in die unmittelbare Verbindung mit den zu erstellenden Meilensteinen gebracht wurde.

Aufgabe 5 - Was ist passiert? Begib dich zum Achievement-Bildschirm.

Als Reaktion auf das Erstellen von Meilensteinen wurde ein Achievement erhalten, was durch eine Push-Notification verdeutlicht wurde. Insgesamt haben 3 von 4 Testpersonen diese Frage korrekt beantworten können. Eine der Testpersonen hat das durch eine Push-Benachrichtigung verdeutlichte Achievement nicht als solches wahrgenommen und war verwirrt. Nach einer Erklärung hat diese Person verstanden was passiert ist und hat sich wie die anderen Personen erfolgreich zu dem Achievement-Bildschirm begeben.

Aufgabe 6 - Füge den Meilensteinen Unteraufgaben hinzu.

Alle 4 von 4 Testpersonen haben diese Aufgabe abschließen können, wobei 3 von 4 Personen Schwierigkeiten hatten, die gegebene Android-Tastatur nicht nur als Eingabewerkzeug, sondern auch das erforderte Betätigen mithilfe des Enter-Buttons zu identifizieren. Dies führte dazu, dass die Testpersonen einen Bestätigungsbutton auf dem Bildschirm gesucht haben. Es wurde der Wunsch nach einem Feedback bei Hinzufügen von Aufgaben geäußert, um Kenntnisse über den erfolgreichen Abschluss des Vorgangs zu gewinnen.

Weiterhin führte die Darstellung der ein- und ausklappbaren Aufgabenliste vermehrt zu Irritationen, da offenbar das Gefühl vermittelt wurde, dass die eingetragenen Aufgaben verschwunden seien.

Aufgabe 7 - 10 Allgemeine Operationen ausführen

Aufgabe 7 bis Aufgabe 10 befassen sich mit weiteren allgemein in der Software zu tätigen Benutzerinteraktionen, welche aufgrund der gleich ausfallenden Ergebnisse im Folgenden zusammengefasst werden:

- Aufgabe 7 - Das Bearbeiten von Meilensteinen (Datum ändern)
- Aufgabe 8 - Die Navigation innerhalb der Software
- Aufgabe 9 - Das Abschließen von Meilenstein-Unteraufgaben
- Aufgabe 10 - Das Abschließen definierter Herausforderungen

Alle Testpersonen konnten diese Aufgabe ohne Schwierigkeiten abschließen.

6.2.2. Zusammenfassung der Testdurchführung mit den Professoren

Insgesamt wurden alle im Rahmen der Testdurchführung gestellten Aufgaben auch von den Professoren erfolgreich gelöst. Der Schwerpunkt der Tests lag hierbei jedoch auf der Betrachtung der einzelnen Features aus der Meta-Perspektive, welche sich durch die Erfahrung der Professoren auszeichnet und die Beurteilung der Softwareumsetzung hinsichtlich des Inhalts vorsieht.

Im folgenden Verlauf ist eine nach den Features der Software kategorisierte Zusammenfassung der Ergebnisse zu finden:

Kategorie 1 - Der Startbildschirm bei erstmaligem Ausführen der Applikation

Es wurden von einer der beiden Testpersonen verschiedene Anmerkungen zu dem Bildschirm geäußert, auf dem die Datumseingabe für den Bearbeitungszeitraum der Bachelorarbeit erfolgt. Zum einen wurde angemerkt, dass es sich bei dem Datum um das amerikanische Datumsformat (MM/DD/YYYY) handelt und dies zu Verwirrungen führt. Weiterhin wurde angemerkt, dass Informationskästchen mit Inhalten zur Bearbeitungszeit der Bachelorarbeit auf dem Bildschirm platziert werden könnten.

Kategorie 2 - Bachelorarbeit Guide

Es liegen keine konkreten Anmerkungen zu dieser Funktion vor.

Kategorie 3 - Meilensteinverwaltung

Grundsätzlich wurden alle das Meilenstein-Tool betreffende Aufgaben erfolgreich gelöst, jedoch kam es bei dem Bestätigen von Unteraufgaben aufgrund der nicht erkennbaren Bestätigungsmöglichkeit in beiden Testdurchführungen zu Schwierigkeiten.

Die Darstellung dieser Option wird jedoch vielen Geräten in unterschiedlicher Weise visualisiert, weshalb es sich hierbei wohl um eine Frage der Gewohnheit handelt. Dies ist insofern wichtig, da alle Testdurchführungen auf einem für die Testpersonen fremden Gerät durchgeführt wurden.

Kategorie 4 - Achievements

Bei Verwendung der Achievement-Funktion wurde angemerkt, dass die zusammenhängenden Achievements und Herausforderungen gegebenenfalls nicht trennscharf genug gehandhabt werden. Es wird gewünscht, dass klassische Gamification-Badges Einzug in die Applikation erhalten.

Kategorie 5 - Herausforderungen

Die Herausforderungen wurden als äußerst sinnvoll bezeichnet, da sie bei richtiger Gestaltung die verschiedenen folgenden Checklisten-Eigenschaften erfüllen können:

- Man kann überprüfen, ob man alle relevanten Inhalte einer Bachelorarbeit bedacht hat.
- Man wird auf Dinge hingewiesen, die man sonst nicht rechtzeitig bedacht hätte.
- Herausforderungen können einen auffordernden Charakter haben.
- Gute Ergänzung zum individuellen Meilenstein-Tool durch stärkere inhaltliche Vorgabe.
- Orientierungsmöglichkeit für Studierende.

Weiterhin wurden die folgenden Wünsche für eine gute Umsetzung von Herausforderungen betont:

- Das selbständige Erinnern an Herausforderungen könnte eine nützliche Funktion sein.
- Verbindung zu anderen Features der Applikation (Bachelorarbeit Guide) sollte gegeben sein, da die Herausforderungen nicht zu isoliert vorliegen sollten und das Gesamtpotential der Software genutzt werden sollte.
- Die korrekte Formulierung der Herausforderungen ist essenziell, da es nicht sinnvoll wäre, wenn man nur Achievements bekommt, wenn man etwas angefangen hat, sondern auch wenn man etwas fertigstellt.
- Eine Mechanik, um Selbstbetrug zu verhindern, wäre wünschenswert.

Kategorie 6 - Dashboard

Das Dashboard und der auf diesem befindliche Zeitstrahl wurde im Rahmen der Tests als sehr sinnvoll eingeschätzt, da die folgenden Aspekte bei den Studierenden gefördert werden könnten:

- Motiviert Studenten dazu, sich überhaupt einen Plan zu machen und sich folgende Fragen zu stellen:
 - Welche Meilensteine habe ich?
 - Wie lange brauche ich?
- Die einfache Möglichkeit der Übersicht und das Fördern folgender Fragen:
 - Wie weit bin ich gerade?
 - Bin ich früh oder zu spät dran?

Weiterhin wurde die Anmerkung gemacht, dass die Auswahl von Meilensteinen auf dem Zeitstrahl erleichtert werden könnte, wenn die Meilensteine über eine andere Visualisierung dargestellt würden, da diese zu klein sind und gerade bei mehreren Meilensteinen nicht auswählbar sein könnten.

Kategorie 7 - Allgemeine Anmerkungen

Eine der Testpersonen hat die Idee geäußert, einen initialen vorbildlichen Meilensteinplan zu erstellen, an dem sich die Anwender für ihre eigenen Zwecke orientieren können.

6.2.3. Ergebnis der Nutzerevaluation

Zusammenfassend lässt sich aus der Evaluation mit den Professoren und den Studierenden ein konkreter aktueller Stand ableiten, welcher als Orientierungspunkt für die weitere Entwicklung der Software dient.

Der Software-Prototyp gilt als noch nicht reif für den realen Einsatz, da er sowohl hinsichtlich der Usability, als auch der inhaltlichen Umsetzung im Rahmen der Tests als ausbaufähig und unfertig erscheint.

Die folgenden Punkte dienen als Konkretisierung der größten zu leistenden Änderungen oder Erweiterungen der Software:

- Das Zeitstrahl-Element auf dem Dashboard sollte durch eine geeignetere Visualisierung ersetzt oder zwecks besserer Bedienbarkeit bei mehreren eingetragenen Meilensteinen entsprechend angepasst werden.
- Achievements sollten stärker von den Herausforderungen differenziert sein und durch eine stärkere der Gamification entsprechenden Visualisierung, wie beispielsweise Badges statt Farben, verfügen.
- Herausforderungen sollten mit den jeweiligen Informationen des Bachelorarbeit Guide verbunden werden, da sie zu isoliert voneinander vorliegen und das mögliche Potential der Funktion nicht genutzt wird.

Des Weiteren sollen die von den Testpersonen angesprochenen Details im Umgang mit der Software berücksichtigt und eine Lösung geeignet umgesetzt werden, worauf eine weitere Testiteration durchgeführt werden soll.

6.3. Bewertung des Frameworks Flutter zur Entwicklung von mobilen Applikationen

Im Laufe der in dieser Arbeit durchgeführten Entwicklung einer Applikation mit dem Framework Flutter haben sich sowohl subjektive Erfahrungen als auch objektive Fakten ergeben, die im folgenden Verlauf unter der Fragestellung, inwiefern sich Flutter für die Entwicklung von mobilen Applikationen eignet, bewertet werden.

Die Wahl der Bewertungskriterien orientiert sich zum einen an den im Rahmen einer Softwareentwicklung bezeichnenden Eigenschaften eines Frameworks. Dazu gehören die Qualität der Dokumentation, die Komplexität des Erarbeitungsaufwandes, die Wahl der zu verwendenden Programmiersprache, die Handlichkeit der allgemeinen Anwendung, die Performance-Eigenschaften und der Support sowie die zugehörige Community-Aktivität des Frameworks. Zum anderen werden die Framework-spezifischen Merkmale mit in den Bewertungsumfang einbezogen, um die klaren, den Anwendungsfall betreffenden, Stärken des Frameworks abzubilden.

6.3.1. Einführung und Dokumentation

Das Framework Flutter weist eine ausführlich gestaltete und sehr gut strukturierte Online-Dokumentation vor, welche besonders auf die für Einsteiger wichtigsten und interessantesten Funktionen mit ausführlichen Beispielanwendungen und Schritt-für-Schritt Anleitungen eingeht. Besonders wird dies durch die Hilfestellungen der zielplattform-spezifischen **Get Started Anleitung**[18] verdeutlicht.

Hierbei werden ausführliche Anleitungen zur Einrichtung und Installation der benötigten Softwarekomponenten für die verbreiteten Betriebssysteme Windows, MacOS und Linux gegeben. Nach Abschluss der Installationsschritte wird der Flutter-Neueinsteiger im zweiten Teil des **Get Started** Programms mit dem Erstellen einer Beispielapplikation unter detaillierter Anleitung und einer abschließenden Tour durch das Flutter-Framework in die App-Entwicklung entlassen und sollte die benötigten Grundlagen zur selbstständigen Weiterbildung durch die Dokumentation erlernt haben.

6.3.2. Verwendete Programmiersprache

Mit **Dart**[10] erhält man bei der Entwicklung mit Flutter eine Google zugehörige, moderne, ebenfalls gut dokumentierte und für die mobile Entwicklung optimierte Programmiersprache, die eine objektorientierte Programmierung unterstützt.

6.3.3. Handhabung und Anwendung

Durch die Handhabung der Flutter-Widgets, welche in einer Baumstruktur organisiert sind, die die Grundbausteine einer jeden Benutzeroberfläche darstellen, können Entwickler im Allgemeinen einen sehr intuitiven Umgang erwarten, welcher weiterhin durch die Dokumentation unterstützt wird. Weiterhin gestaltet es sich aufgrund der einfach gehaltenen Grundstrukturen als sehr einfach, in kurzer Zeit eine lauffähige Minimalsoftware zu erstellen, was erneut für eine sehr gute Erlernbarkeit der Funktionsweise von Flutter spricht.

Flutter enthält von Haus aus eine große Vielfalt an Widgets, welche dem Entwickler große Freiheiten bei der Gestaltung der Software lassen und hinsichtlich der Plattformunabhängigkeit viel Arbeit abnehmen.

Es kann jedoch je nach Anwendungsfall abseits des reinen Oberflächendesigns schnell zum Erreichen dieser Grenze kommen, da viele Services, wie beispielsweise Push-Notifications, nicht von Flutter unterstützt werden. Im Rahmen dieser Arbeit wurde hierfür ein externes Plug-In verwendet, wobei zusätzliche zielplattform-spezifische Konfigurationen zwingend notwendig waren.

Als nicht sehr komfortabel hat sich das Ausführen auf einem iOS Endgerät herausgestellt, sofern man keinen Zugriff auf ein MacOS-Gerät oder -Betriebssystem hat. Dies lässt sich zwar nicht auf das Framework zurückführen, sondern auf die Philosophie des Unternehmens **Apple**, sollte jedoch trotzdem an dieser Stelle Erwähnung finden.

Es ist zwar möglich, eine virtuelle Maschine mit einem MacOS-Betriebssystem einzurichten, jedoch gestaltet sich das eventuell sehr umständlich und nicht sehr praktikabel. Da jedoch bei der Entwicklung einer plattformübergreifenden Applikation das Ausführen auf allen Zielplattformen aus Testzwecken vorgesehen sein sollte, erfährt man als Entwickler unter Umständen Unannehmlichkeiten.

6.3.4. UI-Performance

Da Flutter-Applikationen nativ kompiliert ausgeführt werden, sollte es laut Dokumentation auf einfache Weise möglich sein, bei der Ausführung von Software konstante 60 Bilder in der Sekunde zu erreichen[42]. Weiterhin bietet Flutter den sogenannten **Flutter Performance Profiler**[14], der es ermöglicht, die Performance-Eigenschaften der Applikation bei Laufzeit zu Überwachen.

Da die entwickelte Software jedoch nicht die Komplexität aufweist, bei der potenzielle Einbrüche der Performance zu erwarten wären, kann dieser Aspekt nicht überprüfbar bewertet werden.

6.3.5. Support und Community

Da sich Flutter noch in der Beta-Phase der Entwicklung befindet, ist mit einigen Problemen beim Einsatz zu rechnen, jedoch kann bei Nachverfolgung der im Git-Repository erfolgenden Fortschritte und Meilensteine im Allgemeinen davon ausgegangen werden, dass kontinuierlich an regelmäßigen Updates und Softwareverbesserungen gearbeitet wird.

Subjektiv betrachtet hat sich die Verwendung von Flutter im Falle dieser Arbeit als sehr stabil und zuverlässig geäußert.

Aufgrund des derzeitigen hohen und weiter steigenden Interesses an dem Framework ist eine sehr aktive Community zu erwarten, welche auch zur Lösung von Verständnisproblemen, dem Einholen von Ratschlägen und zum einfachen Interessenaustausch als Anlaufstelle herangezogen werden kann.

6.3.6. Framework-spezifische Features

Das wohl ausschlaggebendste Feature von Flutter ist die sogenannte **Hot Reload Funktion**. Es wird während der Laufzeit ermöglicht, innerhalb von Millisekunden die UI-Komponenten neu zu generieren ohne den inneren Zustand des Systems zurückzusetzen. Dies sorgt für eine extrem flexible Gestaltung und Änderung des Obeflächen-Designs während der Laufzeit, was nicht nur zur Einarbeitung in das Framework besonders nützlich ist, sondern auch bei der direkten Entwicklung mit Kunden oder den späteren Anwendern. Beispielsweise könnten Kunden bei Präsentation der Software Änderungswünsche äußern, welche je nach Aufwand der Änderung direkt umgesetzt werden können. Dies könnte bei den typischerweise häufigen Zwischenpräsentationen im Rahmen von agilen Softwareprojekten zu einer Senkung von Kommunikationsbarrieren sorgen.

6.3.7. Zusammenfassung der Bewertung

Zusammenfassend lässt sich sagen, dass sich Flutter ausgesprochen gut für die zielgruppennahe Front-End-fokussierte Entwicklung von mobilen Applikationen nutzen lässt. Die größten Stärken von Flutter liegen ganz eindeutig in den schnell und effizient zu erreichenden Ergebnissen, was Oberflächendesign betrifft. Die Kunden und späteren Nutzer können unkompliziert in die Entwicklung und Veränderung der Oberfläche miteinbezogen werden, was sich in der Entwicklung von mobilen Applikationen stark anbietet.

Allerdings muss darauf hingewiesen werden, dass Flutter aus der eigenen Natur heraus kein Alleskönner ist, was sich durch die Präsentation des Frameworks seitens Google in der Öffentlichkeit erst auf den Zweiten Blick erahnen lässt, da der Konsument der Dokumentation oder der verschiedenen veröffentlichten Vorträge von dem vermeintlichen Überschuss an Vorteilen überwältigt wird. Beispielhaft hierfür ist das Handhaben der in Kapitel 6.3.3 angesprochenen Push-Notifications, da diese Funktionalität nicht durch das Framework Flutter gegeben ist. Dieser Umstand kann möglicherweise zu einer erhöhten Komplexität hinsichtlich der Wartbarkeit der Software führen, da sich somit auf eigene Verantwortung von Googles Versprechen einer plattformunabhängigen Entwicklung entfernt wird.

Flutter wird im Rahmen dieser Diskussion als sehr geeignet für die Entwicklung mobiler Applikationen bewerten, da das Framework viele Funktionen bietet oder vereinfacht, welche ihre Stärken besonders in der benutzernahen Softwareentwicklung hinsichtlich der schnell zu generierenden prototypischen Fortschritte und Änderungen von Oberflächen-Konzepten zeigen. Mit der Programmiersprache Dart enthält Flutter weiterhin eine optimierte und mächtige Programmiersprache, welche im Gegensatz zu der fokussierten Front-End-Ausrichtung von Flutter auch ein breites Spektrum von Möglichkeiten für die Back-End-Entwicklung bietet.

7. Zusammenfassung

Im folgenden zusammenfassenden Kapitel werden die erreichten Ergebnisse sowie die während der Arbeit aufgetretenen Schwierigkeiten und Erkenntnisse resümiert. Außerdem wird ein Ausblick der Software gegeben und die im Laufe der Arbeit gesammelten Ideen zur Weiterführung des Projekts kurz beschrieben.

7.1. Die wichtigsten Erkenntnisse der Arbeit

Die Untersuchung dieses Problembereiches brachte im Allgemeinen das Ergebnis hervor, dass Studierende und Professoren im Kern sehr ähnliche Erwartungen und Ansichten bezüglich dem Bearbeiten von Bachelorarbeiten haben und das eigentliche Problem in der Ausführung des Projekts selbst liegt. Studierende sehen sich hierbei zum ersten Mal mit einem umfangreichen und derart komplexen Projekt konfrontiert, welches sie in dieser Form an ihre Grenzen bringt. Die hierbei auftretenden Schwierigkeiten lassen sich größtenteils auf die fehlende praktische Erfahrung mit solchen Projekten zurückführen, da diese Erfahrungen in einem solchem Umfang vorher nicht im Studium behandelt werden können. Die Durchführung einer wissenschaftlichen Arbeit sowie die zu einem solchen Projekt zugehörige durchdachte Zeitplanung, stellt somit im Laufe der Bachelorarbeit die erste komplexe Lernerfahrung und große Herausforderung dar.

Das größte identifizierte Problem hierbei ist, dass sich Zeitplanung nur durch eigens gesammelte Praxiserfahrung und nicht durch das theoretische Aneignen von Fachwissen oder Besuchen des an der Fachhochschule Lübeck angebotenen Bachelorarbeit-Seminars erlernen lässt. Da die im Studium durchlaufenen Projekte meist in Gruppenarbeit verlaufen und zudem, besonders aus diesem Grund, sehr überschaubar für den Einzelnen ausfallen, kommt es unter Umständen nicht zu einem vorherigen Lernprozess dieser Größenordnung. Die somit erste **Forderung** von Zeitmanagement mündet offenbar in einer **Überforderung** bei der Bearbeitung der Bachelorarbeit.

Hinsichtlich der fachspezifischen Kompetenzen der Informatik gibt es auffallend wenig Schwierigkeiten, da Studierende im Laufe des Studiums an der Fachhochschule Lübeck größtenteils die für die Bachelorarbeit benötigten Methoden, Werkzeuge und Fähigkeiten durch die zahlreichen und vielseitigen Module theoretisch und auch praktisch vermittelt bekommen haben.

Das Ergebnis der Nutzungsevaluation brachte hervor, dass die Softwarelösung durchaus Potential zur Qualitätssteigerung der Bachelorarbeiten von Studierenden haben kann. Besonders die enthaltene Funktion der Herausforderungen als Leitfaden sowie der Bachelorarbeit-Guide als Informationsquelle stoßen auf hohen Zuspruch seitens der Professoren und der Studierenden. Um eine ausreichend fundierte Aussage über den Nutzen der Software zu ermöglichen, muss jedoch ein Praxis-Test durchgeführt werden, welcher sich im Optimalfall auf mehrere Iterationen von Bachelorarbeit-Studiengängen bezieht.

7.2. Schwierigkeiten bei der Bearbeitung

Als persönliche größte Schwierigkeit betrachte ich das Zeit- und Aufwandsmanagement des Gesamtprozesses, da die zugehörigen Teilprozesse auf eine unvorhergesehene hohe Komplexität gewachsen sind. Konkret zähle ich hierzu besonders das Durchlaufen der Anforderungsanalyse, welches durch die in diesem Projekt detaillierte Vorgehensweise der Durchführung und Auswertung der Interviews und Befragungen, einen enormen Arbeitsaufwand eingenommen hat. Das gewählte Vorgehensmodell führte jedoch auch zu einem sehr detaillierten Bild des Problembereiches, was bei der Entwicklung dieser Applikation als essenziell betrachtet wird.

Durch diesen betriebenen Aufwand kam es somit bei den darauf folgenden, nicht parallelisierbaren Teilprozessen zu einer starken Verschiebung im Zeitplan, was wiederum zu einer Minderung der Gesamtqualität des Projektausgangs geführt hat. Dies wurde vor allem in der Evaluationsphase als enorme zu bewältigende Schwierigkeit wahrgenommen, weshalb die Auswertung des Nutzens der Applikation unter Zunahme weiterer Gründe als nicht vollwertig betrachtet werden kann. Die weiteren Gründe hierfür sind die eingeschränkten Möglichkeiten zur Nutzungsevaluation, welche durch den Rahmen des Projekts innerhalb der gegebenen Bearbeitungsfrist als nicht ausreichend umsetzbar angesehen wird. Der hierfür zu investierende Aufwand wäre ein groß angelegter Einbezug von Studierenden und Professoren, welcher bestenfalls über mehrere Iterationen von Bachelorarbeiten durchgesetzt werden müsste, um die gewünschte Aussagekraft zu erhalten.

7.3. Erfüllung der Aufgabenstellung

Bei der Durchführung des Projektes wurde die Erfüllung der Aufgabenstellung als zentrales Projektziel angesehen. Hierbei gestaltete es sich jedoch aus Zeitgründen schwierig, die angestrebte Qualität in voller Gänze zu erreichen und es mussten bezüglich der Vollständigkeit einige Abstriche gemacht werden.

Im Verlauf des Kapitels werden die erreichten Ziele beschrieben und die nicht vollständig umgesetzten Lösungen begründet. Die dabei gewählte Kategorisierung setzt sich aus den aus der Aufgabenstellung hervorgehenden Aufgabenfeldern der Software zusammen.

7.3.1. Unterstützung bei Bearbeitung der Bachelorarbeit

Um den Studierenden Hilfestellungen in Form von Richtlinien und Ratschlägen zu ermöglichen, wurde einerseits die Bachelorarbeit-Guide Funktion implementiert, welche als Nachschlagewerk zu Rahmenbedingungen, Hinweisen und Bearbeitungsdetails dienen soll.

Der Inhalt dieser Softwarefunktion wird durch das Einlesen einer JSON-Datei aus einem Git-Repository ermöglicht. Diese Schnittstelle sorgt dafür, dass diese Inhalte beliebig verändert oder erweitert werden können, ohne einen Eingriff in den Programmcode zu erfordern.

Weiterhin liegen die bereits eingefügten Wissenssegmente bei Ende dieses Projekts aufgrund des zu betreibenden Zeitaufwandes nur in beispielhafter Form vor und gelten vor allem hinsichtlich der Methodenübersicht als nicht vollständig.

Wie bereits erwähnt kann der Inhalt jedoch beliebig erweitert werden, was konkret bedeutet, dass beispielhafte Gliederungen und alle gewünschten Hinweise, Übersichten und Anleitungen

zu fachspezifischen Methoden und Werkzeugen zur Anforderungserhebung, Nachweisführung und/oder Aufbereitung von Ergebnissen beliebig verändert oder hinzugefügt werden können.

Ein weiterer wichtiger Bestandteil der Unterstützung der Studierenden liegt in der Definition von Herausforderungen. Diese Herausforderungen sollen einen Leitfaden für die Studierenden darstellen, welcher so allgemein wie nötig und so detailliert wie möglich gewählt werden soll. Die Idee hierbei ist, dass die Studierenden sich an dem, durch die Herausforderungen definierten, Rahmen und Ablauf einer Bachelorarbeit orientieren können, um von Anfang an einem konkreten roten Faden folgen zu können und nicht "einfach drauf los" zu arbeiten. Es ist von essenzieller Bedeutung, dass die Inhalte so gewählt werden, dass sie die Inhalte einer themenspezifischen Lösung innerhalb einer Bachelorarbeit nicht beeinflussen, da dies einen zu großen Eingriff in die Prüfungsleistung der Studierenden bedeuten würde und dies sich durch einen manipulativen Charakter sogar negativ auswirken könnte.

Hinsichtlich der Erweiterbarkeit ist es bei den Herausforderungen und Achievements im Gegensatz zu dem Bachelorarbeit Guide zum Ende des Projekts nicht möglich, diese software-extern zu bearbeiten oder hinzuzufügen, jedoch lässt sich auch diese Funktion bei mehr Investition von Zeit und Aufwand nachträglich in das Projekt integrieren.

7.3.2. Unterstützung bei Zeitplanung

Um die Studierenden bei der Zeitplanung der Bachelorarbeit zu unterstützen, wurde eine Softwarefunktion implementiert, welche es den Studierenden ermöglicht, Meilensteine und Teilaufgaben individuell zu verwalten und diese durch eine geeignete Oberflächenpräsentation in Form eines Zeitstrahls als permanenten Begleiter und somit auch als Erinnerungsmöglichkeit innerhalb der Applikation zu visualisieren. Eine konkrete Lösung durch interaktive Erinnerungen, welche beispielsweise mittels Push-Benachrichtigungen ausgelöst werden könnten, ist zum Ende des Projektes nicht umgesetzt worden. Dies lässt sich jedoch durch die Nutzung der von dem **Local Notification Plugin** gegebenen Scheduled-Notification Funktion nachträglich in das Projekt integrieren, da die vorhandenen Informationen der Meilensteine zur weiteren Verarbeitung hierfür bereits vorhanden sind.

7.3.3. Motivation der Studierenden

Um Studierende bei der Durchführung der Bachelorarbeit zu motivieren, wurde ein Gamification-System implementiert, welches den Studierenden bei Erfüllung von Herausforderungen und Benutzung der Applikation ein positives Feedback vermitteln soll. Dies geschieht bei Erfüllen von Herausforderungen durch das Auslösen von Notifications, welche den Herausforderungen entsprechend humorvolle und lobende Texte enthalten. Diese Texte bilden gemeinsam mit sich durch die Wertigkeit unterscheidenden Medaillen den Inhalt der sammelbaren Achievements. Inwiefern diese Strategie ihre Wirkung zur Motivationssteigerung beitragen kann, ist als sehr Zielgruppen-spezifisch anzusehen und sollte weiterhin untersucht werden. Bei einigen Personen wirkt das Feedback laut der Evaluation durchaus positiv auf die Motivation, bei anderen jedoch nicht.

7.4. Fortführung des Projekts

In diesem Kapitel werden die Fragen und Ansätze, welche im Rahmen dieses Projekts nicht untersucht oder realisiert werden konnten, angesprochen und beschrieben.

7.4.1. Ausblick auf weitere Untersuchungen

Im Laufe der Anforderungsanalyse wurde im Rahmen der Interviews oft über die zeitliche Konstellation und Abfolge der an der Fachhochschule Lübeck verfügbaren Module diskutiert, da zum jetzigen Zeitpunkt kein konkretes Modul zur Vorbereitung auf das wissenschaftliche Arbeiten angeboten wird. Eine Problemuntersuchung hinsichtlich des Bedarfs an einem solchen Modul oder einer Alternativlösung wird nach Abschluss dieses Projekts empfohlen.

7.4.2. Ausblick auf Softwarefunktionen

Der im Rahmen der Bachelorarbeit entwickelte Software-Prototyp enthält bereits unterschiedliche Funktionen, welche zur Lösung der auftretenden Probleme dienen sollen. Die hierbei entwickelten Software-Inhalte verfügen im Rahmen einer prototypischen Entwicklung über eine nicht ausreichende Qualität zum Einsatz in einer realen Umgebung. Es wäre zwingend notwendig die vorhandenen Funktionen besonders hinsichtlich der Usability und Wirkungsweise zu überarbeiten oder zu erweitern.

Im folgenden Verlauf werden einige konkrete Ideen zur Erweiterung oder Überarbeitung aufgeführt:

- **Erweiterung der Software für andere Studiengänge**

Die entwickelte Software wurde nur für den Studiengang Informatik/Softwaretechnik der Fachhochschule Lübeck entwickelt. Es wäre denkbar, zu untersuchen, inwiefern andere Inhalte von Studiengängen und Hochschulen in die Software integrierbar sein können.

Denkbar wäre eine studiengangsspezifische Auswahl bei erstmaligem Öffnen der Software.

- **Externe definierbare Erinnerungen**

Eine mögliche Funktion wäre es, durch das externe Absenden von Benachrichtigungen an Studierende client-seitige Push-Benachrichtigungen auszulösen. Dies könnte beispielsweise in Form von Newsletter-Benachrichtigungen über neue Bachelorthemen sowohl von internen Betreuern, als auch externen Firmen geschehen, die auch in der Software selbst angezeigt werden.

- **Auswahl von Sprachpaketen**

Eine wichtige Funktion, welche bis zum Ende des Projekts nicht realisiert wurde, ist die Auswahl von verschiedenen Sprachen, da auch viele Studierende aus dem Ausland die Fachhochschule Lübeck besuchen.

- **Eine geeignetere Darstellung von Informationen im Bachelorarbeit-Guide**

Zum Ende des Projekts liegen die vorbereiteten Informationen in einer sehr rudimentären Fassung vor. Im Rahmen der Entwicklung des Prototyps ist lediglich das Darstellen

von Buttons, welche auf den nächsten Bildschirm leiten sowie die Darstellung des Inhaltes durch ausklappbare Oberflächen-Elemente gegeben. Hier können weiterhin vielfältigere Darstellungsmöglichkeiten wie Grafiken oder eingebettete Videos ihren Platz finden, welche die Übermittlung von Informationen auf eine höhere und zeitgemäßere Qualitätsebene bringen könnten.

- **Dynamische Wochenberichte**

Ein gutes Feature, um dem Benutzer zu zeigen was dieser diese Woche geschafft hat, könnte durch abschließende Berichte einer Woche umgesetzt werden. Hierbei könnte dem Benutzer durch eine Übersicht gezeigt werden, welche Aktionen er an welchem Tag durch Nutzen der Meilensteinplanung getätigt hat. Beispielsweise wären hierfür das Erfüllen von eigens erstellten Unteraufgaben der Meilensteine oder das abschließen eines Meilenstein als Ganzes. Weiterhin könnte dem Nutzer hier auch angezeigt werden welche Erfolge er hinsichtlich der Herausforderungen und Achievements geschafft hat. Auf diese Weise könnte der bereits vorhandene Softwareinhalt genutzt werden und ohne großen Erweiterungsaufwand zu einer Qualitätssteigerung der Benutzererfahrung führen.

7.5. Fazit

Im Rahmen dieses Projekts wurde eine mobile Applikation entwickelt, die Studierenden die Möglichkeit bieten soll, durch eine wegweisende Vorbereitung den Prozess der Bachelorarbeit zu überblicken und zu durchlaufen. Hierbei wurde der Fokus unter anderem auf Features zur Verbesserung von Zeitmanagement und dem Bereitstellen eines Informationskanals zu themenrelevanten Inhalten gelegt. Ein weiterer Teil der Software sind die gamifizierten Inhalte, welche eng im Kontakt zu den Checklisten-artigen Herausforderungen stehen. Diese Inhalte erfüllen den Zweck, jeden geschafften Schritt in Richtung des großen Ziels mit einem positiven Feedback zu belohnen und somit das strukturierte Organisieren des Projektverlaufs zu fördern und die Studierenden zu dem Erreichen von Fortschritten zu motivieren.

Es hat sich hierbei gezeigt, dass erst durch die gezielte und kontinuierliche Einbindung der Beteiligten sowie eine genaue Auseinandersetzung mit dem Problembereich die benötigten Kenntnisse zur Entwicklung einer organisierten Projektlösung aufgebaut und erweitert werden können. Dieser Prozess ist vor allem bei der Entwicklung von "benutzerzentrierten" Applikationen, wie in diesem Projekt, in seiner Wichtigkeit nicht zu vernachlässigen und insofern mit ernstzunehmender Präzision durchzuführen.

Abbildungsverzeichnis

1.1. Online-Umfrage: Zusammenfassung der Umfrageergebnisse	4
2.1. Interest Graph nach Bartle [5]	10
2.2. Flutter High-Level Systemarchitektur [16]	11
2.3. Darstellung (iOS und Android): Plattformunabhängige Entwicklung von Applikationen [41, What is Flutter?]	12
2.4. Beispiel: Aufbau von Widgets [7]	13
2.5. Flutter Widgets Architektur [45]	13
2.6. Flutter Rendering [16]	14
3.1. Darstellung der Analysestrategie	17
3.2. Eingrenzung der Bartle-Benutzerklassen	33
4.1. Benutzeroberfläche Meilensteinverwaltung	40
4.2. Benutzeroberfläche Guide	41
4.3. Benutzeroberfläche Herausforderungen	43
4.4. Benutzeroberfläche Achievements	44
4.5. Benutzeroberfläche Navigations-Menü	45
4.6. Benutzeroberfläche Dashboard	46
4.7. Visualisierung der Achievements	50
5.1. Datenflussübersicht der Softwarearchitektur	55
5.2. Architektur der Model-Komponenten	57
A.1. Oberflächenprototyp: Dashboard	135
A.2. Oberflächenprototyp: Guide	136
A.3. Oberflächenprototyp: Achievements	136
A.4. Oberflächenprototyp: Aufgaben	137
A.5. Oberflächenprototyp: Sonstiges	137

Tabellenverzeichnis

2.1. Übersicht der Spiel-Design-Elemente [6]	7
3.1. Übersicht der Interessengruppen	16
3.2. Einbezug der Studierende in Wahl der Gamification-Strategie	31
3.3. Erwartung der Beteiligten an die Applikation	32
4.1. Abdeckung der Produktfunktionen	48
6.1. Übersicht der Reducer-Tests	64

Listings

5.1. Struktur: Inhalt der JSON-Datei	58
5.2. Container-Komponente einer Meilensteinänderung	59
5.3. ViewModel von Container-Komponente einer Meilensteinänderung	60
5.4. Meilenstein-Action Beispiel	60
5.5. Methode: Bestätigung einer Meilensteinänderung	61
5.6. Methode: checkAchievementNotification	61
5.7. Notification Middleware	62
5.8. Reducer Beispiel	63
A.1. Datenfelder der AppState-Klasse	138

Inhalt der beigelegten CD

Das Grundverzeichnis der beigelegten CD enthält den folgenden Inhalt:

Dokumentation/

Dokumentation des Projekts in Form eines Latex-Projekts.

Anhang/

Die im Laufe des Projekts entstandenen Daten.

Anforderungsanalyse/

Ergebnisse, Auswertungen und Audio- sowie Videomitschnitte der Anforderungsanalyse.

Daten/

Erstellte oder gesammelte Daten der in der Software eingesetzten (oder dafür vorgesehenen) Inhalte.

Evaluation/

Ergebnisse, Auswertungen und Audio- sowie Videomitschnitte der Evaluation.

Software/

Android Studio Projekt der entwickelten Software.

A. Anhang

A.1. Befragung der Beteiligten

A.1.1. Transkripte der Interviews mit den Professoren (Interview I)

Interview I

1
2 **Frage: Ihr Fachgebiet wird auf der Internetseite der Fachhochschule Lübeck als “Data-**
3 **Management, Software Engineering” angegeben. Wenn ein Studierender Sie als Betreuer für**
4 **die Bachelorarbeit wünscht und mit einem Themenvorschlag zu Ihnen kommt, der nicht Ihrem**
5 **Fachgebiet entspricht, gibt es dann Themen bei denen Sie den Studierenden an Ihre Kollegen**
6 **weiterleiten?**
7
8 Antwort: Offen bin ich grundsätzlich schon, aber wenn ich das Gefühl habe, dass ich aus meiner
9 fachlichen Kompetenz das Thema nicht gut betreuen kann, würde ich denjenigen weiterdelegieren.
10 Also Rechnernetze wäre zum Beispiel ein Thema, wo ich sagen würde, es gibt noch Andreas
11 Hanemann, der das wahrscheinlich besser betreuen kann, weil es eben dann auf tieferer
12 Protokollebene stattfindet. Das was ich jetzt so mache ist eben HTTP Protokolle und darüber. Und
13 wenn es um Protokolle darunter geht, bin ich nicht mehr der richtige Ansprechpartner.
14
15 **Und andersherum gefragt, was sind typische Arbeiten, die man bei Ihnen schreibt und wie**
16 **sind sie im Allgemeinen aufgebaut? Es gibt ja zum Beispiel Arbeiten, bei denen die**
17 **Entwicklung einer Software im Vordergrund steht.**
18
19 Ich versuche das mal zu strukturieren.
20
21 Also das wäre aus meiner Sicht der erste Typ: Der Studierende entwirft und implementiert einen
22 Softwareprototyp. Aus meiner Sicht ist es immer ein Prototyp bei einer Abschlussarbeit und ich
23 erwarte kein fertiges Produkt, was nach Abschluss der Arbeit direkt in Betrieb genommen werden
24 kann. Und es geht ja auch nicht darum, jetzt vollständig möglichst viele Use-Cases zu
25 implementieren, sondern es geht mehr um den Entwurf einer Architektur, die sich vielleicht im Laufe
26 der Zeit auch weiterentwickeln kann, und man sollte dann immer Grundlegend zeigen, dass man in
27 der Lage ist, als Studierender ein paar Use-Cases zu implementieren und dann werden ja meistens
28 einige rausgesucht und die werden dann in der Arbeit dargestellt.
29
30 Dann haben wir den ganzen typischen Softwareentwicklungszyklus mit dabei, bei Anforderungen
31 muss man gucken, woher kommen die eigentlich? Hat der Student überhaupt die Aufgabe die
32 Anforderung zu erheben oder sind die Anforderungen bereits
33 erhoben worden aus einem externen Umfeld? Wenn er es im
34 Umfeld einer Firma oder einer anderen Organisation macht, dann sind die Anforderungen manchmal
35 schon gegeben, dann muss er sie nur noch verstehen und durchdringen und kann sie vielleicht
36 nochmal infrage stellen. Also es kann seine Aufgabe sein, die Anforderungsanalyse, muss es aber
37 nicht.
38
39 Dann, die Entwurfsphase sollte auf jeden Fall seine Aufgabe sein. Es kann auch sein, dass er nicht
40 auf der grünen Wiese aufsetzt, sondern es kann ja schon Software existieren, die erweitert werden
41 soll. Dann gibt es eine Ist-Architektur und eine Soll-Architektur.
42 Oder er fängt auf der grünen Wiese an, dann gibt es eine komplett neue Architektur.
43 Dann kommt die Implementierungsphase, da will man natürlich auch sehen, dass der Student in der
44 Lage ist, ein Stück Software zu produzieren, und dann noch das Thema Testing, er sollte sich auch
45 Gedanken zu dem Testing machen. Thema Inbetriebnahme spielt meistens keine Rolle. Das wäre so
46 der erste Typ von Arbeit, wo es aus meiner Sicht schon darum geht, dass der Student ein
47 Softwareprojekt an sich organisieren kann und das auch technisch im Griff hat. Sowohl Entwurfs-, als
48 auch Implementierungs- und Testphase. Um was es da inhaltlich geht - welche Technologie - das
49 spielt eigentlich gar keine Rolle.
50

Dann der zweite Typ von Arbeit, wären empirische Arbeiten, wo meistens irgendwas evaluiert wird. Da kommen natürlich auch viele Unternehmen, mit dem Ziel, den Studenten auf so eine Problemstellung anzusetzen. Zum Beispiel: "Es gibt eine neue Technologie und wir wollen mal gucken wie die sich eigentlich Verhält im Vergleich zu bisher gesetzten Technologien", dann ist meistens die erste Aufgabe, sich ein Kriterienkatalog zu überlegen. Wie kann man es nachher ausmessen? Und dann kommt danach der eigentliche Experimentaufbau und die Durchführung dieser Messung, die auch wieder ganz unterschiedlich ausgeprägt sein kann. Es kann Fragebögen geben, so wie jetzt in dieser Situation, wo man Leute befragen muss. Es gibt natürlich auch die Möglichkeit, mit Onlinefragebögen eine größere Stichprobe zu sammeln. Dann gibt es auch Fallstudien, wo ich mir die Technologien anschau - Technologie A und Technologie B für den gleiche Use-Case - und muss mir vorher Kriterien überlegt haben. Und am Ende ist dann natürlich die eigentlich Evaluation der Beitrag des Studierenden - die Ergebnisauswertung des Experiments. Das wäre die zweite Klasse.

Und da jetzt die Frage, ob die dritte Klasse mit da rein gehört? Das ist manchmal schwer zu trennen. Die könnte man auch mit zur zweiten Klasse zählen.

Das ist so in dem Bereich Datenanalyse/Machine-Learning. Manchmal gibt es Arbeiten, wo dann öffentlich ausgeschriebene Wettbewerbe im Bereich Datenanalyse angegangen werden. Also zum Beispiel über „Kaggle“ oder jetzt hatte ich eine Arbeit mit diesem „Spotify Million Playlist Dataset“. Das heißt, da ist eine Datenanalyseaufgabe - die sind natürlich dann offen vom Ergebnis her - und aus meiner Sicht geht es darum, dass der Student sich erst mal explorativ den Datensatz anschaut.

Ach ihr habt so etwas Ähnliches auch gemacht. Nicht als Abschlussarbeit, sondern in dem Wahlpflichtmodul.

Ja, genau.

Also erst mal explorativ den Datensatz anschauen und dann überlegen, mit welchen Machine-Learning Verfahren könnte ich denn im Kontext dieser Aufgabenstellung zu guten Ergebnissen kommen. Dann hat man in der Regel ja auch so ein Benchmarking, wo man das Ergebnis dann hoch lädt und dann ein Feedback bekommt.

Hier ist das Lernziel aus meiner Sicht, dass der Studierende zeigen soll, dass er die Machine-Learning Algorithmen versteht und auch zur Anwendung bringen kann. Bei einer sehr guten Arbeit wendet der Studierende nicht nur einen Ansatz an, sondern mehrere und vergleicht diese. Da hat man teilweise auch Herausforderungen aufgrund der Größe der Datenmenge. Das sind dann so diese Benchmarkinganwendungen. Es gibt auch so Datenbankbenchmarks, da ist das das Gleiche. Das hatte ich auch schon im Wahlpflichtmodul. Da war es dieser Big-Data-Benchmark. Das könnte man natürlich auch in die Abschlussarbeit reinbringen. Da hat man am Ende dann auch ein Evaluationsergebnis, aber der Kontext und die Rahmenbedingungen sind durch so ein Wettbewerb meistens enger gesteckt.

Eigentlich sind es diese zwei Arten (von Arbeiten). Einmal konstruktiv, irgendetwas erschaffen oder analytisch irgendetwas evaluieren/analysieren.

Was wir kaum haben (im Studiengang Informatik/Softwareentwicklung): In der BWL zum Beispiel, gibt es auch reine Literaturarbeiten

Dankeschön. Sie haben es ja bereits angerissen, was ihre Erwartungen betrifft. Aber was sind Ihre allgemeinen Erwartungen an den Bacheloranden?

Ich erwarte von einer Bachelorarbeit erst mal keine neue Erkenntnis. Das würde ich von einer Dissertation erwarten - wirklich neue Erkenntnisse. Und die Masterarbeit ist irgendwo dazwischen. Ich

erwarte aber, dass der Studierende sich selbstständig - nicht so stark begleitet, wie in den Praktika in den Semestern davor - in ein Thema einarbeitet, in dem er sich vielleicht vorher noch gar nicht so gut auskannte. Da erwarte ich eigentlich strukturiertes Vorgehen des Studenten und dass er sich am Anfang so was wie einen Zeitplan macht und Meilensteine überlegt. Ob er das dann mit einem Projektmanagement Tool macht ist mir eigentlich ganz egal. Er kann es auch einfach auf einen Zettel schreiben. Aber das merkt man ja schon relativ früh - bei den ersten beiden Treffen - dann, ob der Studierende sich einen Plan über die gesamte Laufzeit gemacht hat oder ob das einfach so laufen lässt.

Je nachdem sehe ich das als meine Aufgabe, ihn häufiger zu mir zu bitten oder weniger. Aber meistens lasse ich ihn relativ frei laufen. Beispielsweise die beiden chinesischen Studenten kommen einmal in der Woche, weil man das Gefühl hat, dass die das brauchen. Bei den Online-Studenten habe ich sogar das Gefühl, dass sie noch selbstständiger sind. Aber das ist jetzt ein subjektiver Eindruck.

Was da nachher das Ergebnis betrifft, sehe ich die Arbeit immer in zwei Teilen. Es sollte einen Literaturteil geben, wo der Studierende es geschafft hat, das, was er auf technischer Ebene implementiert hat, auch zu vergleichen mit verwandten Ansätzen - Related Work. Zu versuchen, das auf eine höhere Abstraktionsebene zu bringen und grundlegende Muster zu erkennen, diese Muster zu beschreiben - z.B. Architekturmuster, Entwurfsmuster, die nicht nur die nächsten zwei Jahre eine Rolle spielen, sondern länger schon eine Gültigkeit haben.

Und dann eben auch möglichst einschlägige Quellen verwenden. Damit meine ich, sich nicht nur auf allgemeine Softwaretechnik-Lehrbücher zu beziehen, sondern dass man in diesen Softwaretechnik-Lehrbüchern Unterkapitel findet - zum Beispiel zum Thema Requirements Engineering - und sich daraufhin spezielle Bücher zu dem Thema Requirements Engineering anschaut und aus denen zitiert. Und dann kann ich natürlich immer noch einen Schritt weitergehen und sagen, dann finde ich am Ende der Kapitel der Lehrbücher Verweise auf wissenschaftliche Papiere und gucke mir die auch noch an und suche mal nach den Primärquellen. Und das unterscheidet eine bessere Arbeit von einer, die nicht so gut ist - dass man die Suche nach den Primärquellen und die Diskussion des Themas erkennt. Dass so eine Arbeit im Grundlagenteil nur an einer Quelle orientiert ist und man eigentlich nichts anderes getan hat, als diese Quelle umzuschreiben oder zu übersetzen, sondern dass da mehrere Quelle ernsthaft eingeflossen sind und manchmal gibt es ja auch Aussagen in Quellen, die sich widersprechen. Und das wäre dann die Literaturarbeit.

Der andere Teil wäre der praktische Teil. Hier ist es Aufgabe des Studierenden und des Betreuers, zu Anfang zusammen die Komplexität so zu wählen, dass es nicht zu viel oder zu wenig ist. Wenn es zu wenig ist, kann man sagen: "Das, was du gemacht hast, war zu trivial und führt dann auch nicht zu einer guten Note", und wenn es zu viel ist, dann kommt es entweder nicht zu einem runden Ende oder es ist alles nur so angerissen. Das heißt, dann muss der Betreuer mit seinem Erfahrungswissen ein bisschen steuern. Der Studierende aber auch, vor allem wenn es eine externe Arbeit aus einem Unternehmenskontext ist. Dann fällt es mir manchmal schwer, es von Anfang an richtig einzuschätzen - wie viel Aufwand sich da wirklich verbirgt.

Das Ergebnis soll natürlich am Ende laut Aufgabenstellung gelöst sein - dafür hat man ja die Aufgabenstellung, wo drinnen steht, welche Funktionalitäten abgedeckt werden sollen - und zum anderen sollen natürlich externe Anforderungen auch abgedeckt werden. Die Anforderungen kann man ja aufschreiben im Rahmen der Arbeit.

Das wäre so der praktische Teil. Und da gibt es durchaus ja Unterschiede, je nach Art der Arbeit. Wenn jemand das Gefühl hat, der Praxisteil war nicht so ergiebig, oder etwas einfacher und war schnell fertig, der hat natürlich die Möglichkeit, das im Literaturteil wieder auszugleichen. Grundsätzlich würde ich sagen, die Empfehlung ist 50/50, aber es gibt auch Schwankungen von 30/70 und 70/30 und das ist auch absolut ok.

Darauf aufbauend wäre eine Frage, wo denn häufig Probleme auftreten? Gibt es Probleme seitens der Studierenden, die immer wieder auftreten? - Stichwort Zeitmanagement.

Es gibt Studierende, die neigen dazu alles auf den letzten Drücker zu machen. Das ist natürlich nicht gut und das wissen die Studierenden meistens auch selbst. Es ist von Vorteil, wenn man sich alle 2 oder 3 Wochen mit dem Betreuer trifft und dann eine Vorversion hat, oder einen Prototypen, den man vorstellen kann. Es ist immer besser über irgendwelche Artefakte zu sprechen, als über das was man noch tun möchte und welche Probleme einen davon abhalten, dass man das noch nicht getan hat. Also eine Empfehlung an die Studierenden wäre: "Wenn man einen Termin mit seinem Betreuer eingeht, am besten immer die Entwicklung zeigen können".

Beim ersten Termin erwarte ich immer eine Gliederung und 3 oder 4 Einträge im Literaturverzeichnis als Hauptquellen. Und beim nächsten Treffen sollte ich dann immer was Neues zeigen können und nicht noch mal das gleiche. Ein paar Einträge im Literaturverzeichnis hin und her schieben geht innerhalb von 5 Minuten - das meine ich damit nicht. Und wenn der Studierende sich hauptsächlich mit Lesen beschäftigt hat, kann man trotzdem in den Unterkapiteln zeigen, welche Quellen man eigentlich zitieren möchte.

Beispiel: "Thema A nehme ich aus Kapitel 5 von dem Buch; Thema B aus einer anderen Quelle; ...".

Also auch wenn man die Zeit nur zum Einarbeiten genutzt hat, kann man das auch zu Papier bringen.

Dann ein Thema: wissenschaftliches Arbeiten an sich. Und da müssen wir als Fachhochschule, speziell auf diesen Studiengang (Informatik/Softwareentwicklung), uns mal an die Nase fassen, da wir wenig vorarbeiten.

Es gibt bei uns in der Form keine Übung in Form einer wissenschaftlichen Seminararbeit oder Hausarbeit, die sich als Übung für die Bachelorarbeit eignet. Ich kenne das von einer anderen Hochschule - da sind 2 Seminararbeiten vorher Pflicht - und dann kommt natürlich auch tendenziell ein besseres Ergebnis in der Abschlussarbeit dabei raus. Weil der Studierende dann genau das schon mal gemacht hat und erkennt, woraus diese Literaturarbeit eigentlich besteht. Es ist ja nicht die Aufgabe, wenn man ein scheinbar relevantes Paper mit 20 Seiten bekommt, dass man diese 20 Seiten im Detail durchliest und alles versteht.

Es geht ja meistens darum die Idee des Papers zu verstehen - Literaturscreening gehört ja auch dazu - und die paar Seiten, die wirklich relevant sind, für das was man tut, zu lesen und zu verstehen. Ein Lehrbuch liest man ja auch nicht von vorne bis hinten.

Der praktische Teil - Die Beschreibung was man gemacht hat - gelingt da meistens besser, aber sich mit mehreren Quellen auseinanderzusetzen und diese ineinander fließen zu lassen gelingt nicht immer.

Das nächste ist dann der wissenschaftliche Anspruch, Struktur und der Versuchsaufbau. Dass ich mir vorher klar mache, was das Ergebnis meiner Arbeit ist und wie der Weg dahin aussieht, dass man messbare Kriterien braucht und welche Fragen man stellen muss um diese Kriterien dann zu erfüllen. Da gibt es ja diesen GQM Ansatz, was vielleicht auch ein Vorschlag für das Bachelorseminar wäre:

Goal Question Metric

Goal	- Man überlegt sich Ziele die man erfüllen will.
Question	- Fragen, um diese Ziele zu erfüllen.
Metric	- Und die Antworten auf die Questions sind meistens messbare Metriken.

Diese Methode ist natürlich auf einem sehr hohen Abstraktionsniveau aber grundsätzlich vielleicht mal nicht schlecht, sich das als Student zu fragen: wie kann ich das eigentlich quantifizieren? Im Evaluations-Kapitel steht drin, was gut geklappt hat und was nicht geklappt hat, aber besser wäre es, wenn das durch Zahlen belegbar ist, aber das gelingt (den Studenten) nicht immer.

207
208 Ich fasse nochmal zusammen:

- 209 - Zeitmanagement
- 210 - Wie funktioniert eigentlich wissenschaftliches Arbeiten? Literaturrecherche?
- 211 - So etwas wie ein strukturierter Experimentaufbau, vor allem bei dem zweiten Typ von Arbeit.
- 212 Bei dem Ersten Typ habe ich eigentlich Anforderungen und ich muss belegen, dass diese
- 213 erfüllt werden.

214
215 **Sie haben es ja bereits angerissen mit Ihrem Gedanken, dass die Hochschule kein Modul**
216 **anbietet, welche die Studierenden auf diese Problematik vorbereitet. Wie würden Sie diese**
217 **Probleme versuchen zu lösen, wenn Sie könnten?**
218

219 Das ist ja erst mal kein grundsätzlich neues Thema. Als ich Student war, hatte ich wissenschaftliches
220 Arbeiten im ersten oder zweiten Semester und es ist total demotivierend. Da geht nachher kein
221 Student mehr hin, man kann auch schlecht eine Prüfung in Form einer Klausur machen und es hat
222 aus meiner Sicht auch keinen Sinn, das so früh zu machen. Ich finde das hier schon gut, dass es das
223 Bachelorarbeit Seminar gibt - es findet natürlich im 6. Semester statt, das heißt es ist parallel zur
224 Bachelorarbeit. Vielleicht würde es Sinn machen, wenn es im 5. Semester liegen würde, aber auch
225 nicht viel früher. Das, was man in so einem Kurs an Theorie im Frontalunterricht machen kann, ist in
226 der Regel nicht sehr spannend und schwierig zu gestalten. Also würde ich versuchen diesen Teil
227 möglichst kurz und prägnant zu halten und stattdessen den Studenten selbst eine Art Mini-Übung zu
228 einer wissenschaftlichen Arbeit in Form einer Seminararbeit schreiben zu lassen. Und dann eine
229 Form der Arbeit, die sich ganz stark auf den Literaturteil bezieht. Das heißt, man bekommt ein oder
230 zwei Quellen vorgegeben zu einem bestimmten Verfahren oder einer mathematischen Methode und
231 soll die dann beschreiben und an einem Beispiel dann auch zeigen, in Form einer Präsentation am
232 Ende. So würde ich das machen, das haben wir auch häufig schon diskutiert. Das kann dann
233 genauso heißen wie jetzt auch und ist vielleicht nicht nur einem Kollegen zugeordnet, sondern jeder
234 Student sucht sich, wie bei einer Bachelorarbeit auch, einen Betreuer. Vielleicht kann man auch einen
235 Pool aufbauen an Themen. Alle Kollegen dürfen dann 5 Themen angeben und die werden dann unter
236 den Studierenden irgendwie verteilt.

237 Das muss natürlich vom Aufwand reduziert sein, da es ja nur um das wissenschaftliche Schreiben
238 und die wissenschaftliche Literaturarbeit - der Umgang mit Quellen - geht.

239 Das könnte man machen, aber man darf nicht vergessen, dass es sich dabei "nur" um die
240 Bachelorarbeit dreht. Früher gab es das Grund- und das Hauptstudium und am Ende nach 10
241 Semestern musste man erst nachweisen, dass man das wissenschaftliche Arbeiten beherrscht und
242 da hatte man natürlich zwischendurch auch Hausarbeiten auf dem Weg dahin. Jetzt könnte man ja so
243 argumentieren, dass die Masterarbeit eigentlich der Part ist, bei dem Sie das zeigen müssen und die
244 Bachelorarbeit ist die Übung, aber nun hören halt viele nach der Bachelorarbeit auf. Das darf man
245 nicht vergessen. Die Bachelorarbeit muss aus meiner Sicht nicht perfekt sein, sondern ist nur eine
246 Vorübung zur Masterarbeit.

247
248 Mein Gestaltungsvorschlag wäre also, das Seminar aus dem 6. Semester in das 5. Semester zu
249 verschieben und dann so einen Themenpool aufzubauen. Bei der Bachelorarbeit erwarte ich ja vom
250 Studenten dann, dass er sich auch selbst ein bisschen um das Thema kümmert. Hier würde ich
251 sagen, der Student soll sich nicht selbst darum kümmern, sondern es werden geeignete Themen aus
252 unterschiedlichen Bereichen der Informatik zusammengestellt, sodass jeder sich was aussuchen
253 kann. Die Themen könnten sich dann natürlich auch wiederholen.

254
255 Es gibt ja auch im Internet Bachelorarbeiten, die man durchlesen kann, man muss dann natürlich
256 auch selbst in der Lage sein, eine schlechte Bachelorarbeit von einer guten zu unterscheiden, aber
257 ich glaube, das wird ja auch im Bachelorseminar so gemacht und das klappt ja auch erstaunlich gut.

Ein Student kann ja eine schlechte Bachelorarbeit von einer guten unterscheiden, und deshalb kann man ja auch seine eigene Arbeit relativ gut einsortieren in diese Bandbreite. Was mir auch auffällt ist, dass sich viele Studierende nicht rechtzeitig mit der Themenfindung beschäftigen. Vielleicht sollte man da einen Hinweis geben, das kann ich nicht so ganz nachvollziehen, weil es ja eigentlich bekannt ist, das man in dem 6. Semester diese Abschlussarbeit schreibt.

Das ist auch etwas, was ich von vielen Studierenden gehört habe. Die Studierenden wissen meistens gar nicht was eigentlich "erlaubt" ist, bezüglich der Themenwahl und aus meiner Sicht tun sich viele sehr schwer damit, sich zu informieren. Vielleicht haben die Studierenden Berührungängste oder einfach Angst mal nachzufragen, weil sie denken, dass sie dann einen schlechten Eindruck hinterlassen?

Ja, da sehe ich ganz klar eine hohe Schuld bei den Studierenden. Im 6. Semester hat man im Laufe des Studiums viele Personen kennen gelernt. Wie haben sehr viele professionelle Lehrkräfte an der Fachhochschule und einer sollte wohl dabei sein, zu dem man schon ein gewisses Vertrauen gefunden hat und bei dem man sich auch traut zu fragen. Und man kann ja wirklich jeden Fragen. Wenn man sich nicht für Rechnernetze interessiert, aber das Gefühl hat, dass der jeweilige Kollege nett ist, dann kann man ja hingehen und einfach mal nach dem Prozess hinter der Bachelorarbeit fragen - dann wird man sicherlich auch Antworten bekommen.

Was wir ja nicht wollen - und da glaube ich, für mehrere Professoren zu sprechen - ist, das ganze hier noch weiter zu "verschulen". Dafür gibt es ja letzten Endes auch das Curriculum. Da stehen die Module drin, da steht die Abschlussarbeit im 6. Semester drin. Man kann auch in der Prüfungsordnung lesen was eigentlich alles zulässig ist - es liest nur keiner.

Genau da liegt das Problem, wer und ob jemand die Schuld dafür trägt, ist ja erstmal zu vernachlässigen. Es existieren ja auf jeden Fall große Probleme bei diesem Prozess. Meine Hoffnung mit der Applikation ist es zum einen, auch die Aufmerksamkeit und den Fokus frühzeitig auf Dinge zu lenken, die immer wieder falsch gemacht werden, sodass dieser Prozess bei den Studierenden vielleicht früher ausgelöst wird.

Die Idee von dieser App finde ich grundsätzlich nicht schlecht - das es erstmals überhaupt ein neues Medium ist oder ein neuer Kanal, um Informationen zu erhalten. Scheinbar reicht es nicht, diese Informationen auf die Website zu schreiben.

Viele Studierende wissen, dass diese Infomaterialien existieren, aber teilweise finden sie sie nicht oder suchen nicht danach. Ein Ansatz wäre es beispielsweise, diese Informationen mit der Fachhochschule zusammen aufzubereiten oder durch Verlinkungen zugänglicher zu machen.

Ja. Das Eine ist das Organisatorische, was aus meiner Sicht vielen Studierenden fehlt. Es gibt ja auch die Möglichkeit, sich von externen Unternehmen die Aufgabenstellungen zu holen. Das machen aus meiner Sicht erstaunlich wenige, obwohl es sehr viele Unternehmen gibt, die händierend Informatikstudenten suchen. Was man überlegen könnte ist, ob man als zusätzlicher Kanal mit einer solchen App auch konkrete (externe) Abschlussarbeiten angeben könnte. Oder ein Pool von möglichen Themen "triggert" vielleicht ein paar Studierende an. Das gibt es seitens unserer Hochschule nicht wirklich. Zumindest nicht koordiniert. Professor Kratzke hat auf seiner Website auch Themen, die er gerne bearbeitet hätte, aber ansonsten hat das glaube ich keiner. Ich hab das auch mal gemacht, aber ich glaube, nur ein Student hat sich daraufhin gemeldet - also auch relativ wenig Reaktion darauf. Könnte man ja drüber nachdenken, ob man von der Hochschule aus so einen Themenpool angeht, aber da braucht man natürlich auch jemanden, der den pflegt - nicht nur von

externen, sondern auch von internen Themen. Jeder ist natürlich bereit, seine eigenen Themen zu pflegen, aber jemand muss natürlich auch das Gesamtsystem pflegen.

Ich hab da ja immer Angst. Man baut eine App, die dann für ein Jahr "cool" ist und ab dann eigentlich durch neuen Content lebt. Wo kommt dieser weitere Content her und wie könnte der aussehen?

Wie würden Sie Ihre Einstellung zu einer solchen App beschreiben?

(lacht) Das hängt vom Endergebnis der App ab, aber grundsätzlich würde ich das begrüßen, dass ein neuer Kanal existiert, über den sich die Studierenden informieren können.

Was ich begrüßen würde ist, wenn es eine Plattform geben würde, wo ich jedes Semester 3-4 Themen eintragen könnte. Das sollte nicht zu aufwändig sein. Irgendwie nur so Vorschlag für eine Überschrift und eine kurze Beschreibung der Idee des Themas. Da ist mir ganz egal wie das Technisch gelöst ist - idealerweise natürlich was browserbasiertes. Dann erscheint das was ich da eingetragen habe in dieser App, die für Studierende gedacht ist.

Das würde natürlich nicht funktionieren, wenn da nur meine Vorschläge drin stehen, sondern natürlich müssten da auch Vorschläge von anderen drinnen stehen. Sowohl intern und extern. Für externe Unternehmen gibt es das ja schon. So was wie „abschlussarbeiten.de“, wo die Unternehmen Praktika oder Abschlussarbeiten inserieren.

Das ist natürlich deutschlandweit. Wenn man dann den Filter auf Lübeck stellt ist ja nichts mehr da. Wenn man dann doch nach Informatik filtert ist noch weniger da. Oder es gibt nur ein prominentes Unternehmen in Lübeck.

Abschluss des Interviews

A.1.2. Transkripte der Interviews mit den Professoren (Interview II)

Interview II

Frage: Ihr Fachgebiet wird auf der Internetseite der Fachhochschule Lübeck als “Rechnernetze, Web-Technologien” angegeben. Wenn ein Studierender Sie als Betreuer für die Bachelorarbeit wünscht und mit einem Themenvorschlag zu Ihnen kommt, der nicht Ihrem Fachgebiet entspricht, gibt es dann Themen bei denen Sie den Studierenden an Ihre Kollegen weiterleiten?

Antwort:

Üblicherweise liegen die Themen schon in diesen Bereichen, aber wenn Studierende selbst ein Thema vorschlagen und von mir betreut haben wollen dann ist das schon mal etwas weiter ausgelegt. Aber sonst, in manchen anderen Fällen, wenn ich denke, dass ein Kollege oder eine Kollegin besser geeignet ist, werden die Studierenden auch weitergeschickt.

Bei der Zweitbetreuung habe ich das bisher recht Großzügig betrachtet. Das läuft ja oft auch etwas unterschiedlich, wenn zum Beispiel der Studierende schon einen Zweitbetreuerwunsch äußert. Der Zweitbetreuer beschäftigt sich ja nicht während der Arbeitszeit mit der Abschlussarbeit, sondern erst zum Ende der Arbeit, Richtung Kolloquium. Es ist aus Zeitgründen nicht möglich, dass zwei Personen kontinuierlich die Arbeit betreuen.

Vielleicht könnten Sie auch mal darüber nachdenken, ob die Applikation nur für Bachelorarbeiten ausgelegt ist oder nicht vielleicht auch für Masterarbeiten.

Für Masterarbeiten haben wir hier ja 6 Monate Zeit und manche Themen müsste man dann noch vertiefter betrachten.

Was ich eigentlich sagen wollte: es müsste bei dem Ablauf der App ja so sein, dass der Studierende den Prozess selbst steuert. Ich mache das Grundsätzlich so, dass ich am Anfang erkläre, dass es um eine selbstständige wissenschaftliche Arbeit geht und ich den Studierenden nicht alles hinterhertragen möchte. Die App müsste auch so programmiert sein, dass die App diesen Prozess auch unterstützt. Das ist bei den deutschen Studierenden nicht so das Problem, während chinesische Studierende sich an das selbstständige Arbeiten erst gewöhnen müssen.

Haben Sie schon eine Idee entwickelt?

Ja, die Idee bisher liegt so vor, dass Studierende selbständig Meilensteine mit Unteraufgaben planen, erstellen und verändern müssen und seitens der App Erinnerungen ausgelöst werden.

Beim Ablauf habe ich das oft so gemacht: man macht gewisse Zwischenversionen und ich kommentiere dann, was besser/anders gemacht werden muss. Und dann erwarte ich beim nächsten Feedback, dass diese Kommentare auch umgesetzt wurden und neuer Inhalt zum weiteren Kommentieren enthalten ist. Ich möchte nicht zweimal die gleichen Kommentare geben müssen.

Die Arbeit wird im Laufe der Zeit immer umfangreicher und man fragt sich als Dozent dann irgendwann: “Was soll ich denn nun genau lesen?”.

Es wäre hilfreich, wenn dann in der Mail steht oder markiert ist, was genau neu ist oder nochmal gelesen werden sollte, damit man nicht erst suchen muss. Bei einigen Texten weiß man sonst nicht, ob es nur um Kleinigkeiten geht oder sich das überhaupt lohnt.

Eine weitere Sache ist die Sache mit den 3 Monaten. Wenn man angemeldet ist, dann gilt folgendes: die Zeit läuft für 3 Monate und man hat somit ein gewisses Enddatum, was nicht verlängert werden kann, außer man wird krank, oder es ist eine externe Arbeit und die Firma kann die nötige Hardware nicht liefern, die man benötigt. Bei Gründen, die man nicht vertreten kann, kann verlängert werden,

aus Selbstverschulden jedoch nicht, z.B. bei zu optimistischen Zeitplan. Aber man hat ja noch vor der offiziellen Anmeldung die Möglichkeit sich vorzubereiten.

Für die App wäre es also nicht schlecht, wenn man den Hinweis auf das eben Gesagte geben würde. Es kam schon oft vor, dass die Studierenden noch am letzten Tag an der Bachelorarbeit gearbeitet haben.

Noch ein wichtiger Punkt ist: es ist ein wesentlicher Unterschied, ob man allgemein schmierpapiermäßig etwas aufschreibt oder ob es ein guter Text (gutes Deutsch, klar formuliert, sauber zitiert, formatierte Bilder) ist. Das wirkliche Aufschreiben kostet immer viel Zeit. Es ist ganz wichtig darauf hinzuweisen. Ein wichtiger Tipp ist hier (Verweis auf studentische Vorlagen zum wissenschaftlichen Arbeiten), von Anfang an nach diesen Vorlagen zu arbeiten. Nicht erst in vielen verschiedenen Dokumenten Sachen festzuhalten oder provisorischen Inhalt zu verfassen. Bei relevanten Literaturstellen sollten gleich alle Quellen sauber angegeben werden und auch von Anfang an überlegt werden, wie man was dokumentiert. Vorher nur in Stichpunkten arbeiten und wenn klar ist, dass die Struktur so bleibt, dann ausformulieren.

Man muss halt überlegen, wie man das mit der App macht. Bringt man die Studierenden schon zu Anfang in eine gewisse Reihenfolge? Schon am Anfang mal überlegen - komplette Arbeit? Ich würde es nicht empfehlen Kapitel 1 fertig zu stellen, und dahinter steht noch gar nichts.

Und sonst die Frage mit der Gliederung. Ich habe in der Vorlage eine Art Standardgliederung, die meines Erachtens nach auf möglichst viele Bachelorarbeiten passen kann. Es gibt unterschiedliche Themen. Ist es ein Programmierthema, wo viel umgesetzt werden muss, dann ist der Text vielleicht nicht so lang. Oder ist es ein vergleichendes Thema, dann ist natürlich mehr Text und weniger praktische Arbeit da?

(Persönliche Einschätzung zu typischen Informatikthemen) Ein Tipp zur Gliederung wäre noch, dass man klar erkennen kann, was der Student eigentlich gemacht hat und was allgemeine Kenntnis aus der Literatur ist.

Häufig habe ich so Themenstellung, die heißen so: "Konzeption und prototypische Implementierung von ...". Da kann man dann am Anfang darstellen, wie die Aufgabenstellungen und was die Anforderungen an das Ziel sind. Dann guckt man in der Literatur was andere schon gemacht haben und was man davon erben kann. Dann macht man ein Konzept-Kapitel und ein Umsetzungskapitel. Und dass es "prototypische Entwicklung" heißt, hat den Hintergrund, dass man sich da selber nicht so in Schwierigkeiten bringt. Wenn man schwerwiegende Programmierfehler hat, sieht das sonst vielleicht etwas blöd aus. Dann kann man immer noch sagen, dass die Umsetzung eher eine Fleißarbeit ist, das Konzeptionelle aber enthalten ist. Das ist aber eher eine persönliche Einschätzung zu typischen Informatikthemen. Aber wenn das andere Themen z.B. aus ITD sind, dann würden die das natürlich anders machen.

Zu den Programmierthemen hat man typischerweise die Anforderungen, erstmal das Thema "Anforderungen" klar aufschreiben, dazu dann MUSS/KANN-Kriterien abwägen und mit Literaturwerken vergleichen. Dann das Konzept ausarbeiten und die folgende Umsetzung erarbeiten. Bei einer professionellen Softwareentwicklung würde man ja nicht einfach was "loshacken". Das wäre ja nicht die professionelle Herangehensweise. Sondern erst muss man sich ja was konzeptionell überlegen und das bei der Umsetzung ggf. sonst nur zum Teil umsetzen.

(Zusammenfassung der Antwort durch den Interviewenden)

Ein bisschen wollte ich noch auf das Thema mit einem Firmenbetreuer eingehen. Wir haben häufig den Fall, dass die Studierenden eine Bachelorarbeit in einer Firma erstellen. Die Firma hat andere Interessen als die Fachhochschule. Die Firma hat oftmals die Situation, ein konkretes Problem zu

lösen, wo vielleicht schon ein spezielles Konzept für vorgesehen ist und somit den Fokus eher auf der praktische Umsetzung liegt. Das ist aber nicht ganz der Anspruch an die Bachelorarbeit. Bei der Bachelorarbeit ist es schon wichtig, dass man da verschiedene Lösungsmöglichkeiten anschaut und die sinnvoll miteinander vergleicht. Das ist aber keine Programmieraufgabe wo das Konzept schon vorhanden ist und Sie es nur umsetzen sollen. Das wäre zu wenig. Eine richtige konzeptionelle Arbeit wird hier von den Studierenden erwartet - soll heißen, THEORIE + Praxis. Da steht unter Umständen der Fokus auf dem frühzeitigen Dialog zwischen Betreuer der Firma und Betreuer der Fachhochschule, um frühzeitig unterschiedliche Erwartungen zu identifizieren.

(Schilderung der Erwartung der Studierenden an das Thema Bachelorarbeit, Bezug auf Übergang von bisherigem Studiumsverlauf zu Bachelorarbeit. Studierende haben oft Unsicherheiten überhaupt zu verstehen, was eine Bachelorarbeit ist und was erwartet wird und wo die Freiheiten liegen)

Ganz andere Situation als in den Studienmodulen, da in den Modulen oft der Gedanke existiert, dass der Code sowieso nicht weiterverwendet wird, sondern nur Lerngegenstand ist. Gerade auch in den Softwaretechnikprojekten sieht das anders aus. Da geht es auch darum, den Code den Nachfolgern sauber zu hinterlegen und zu pflegen. Und es ist im Allgemeinen viel offener als in den Modulen.

(Unterschied nochmal zusammengefasst)

(Die Brücke von Studium zur Bachelorarbeit aus Sicht der Studierenden - Problem: Übergang zu Hart?)

Sie würden lieber noch ein anderes Modul dazwischen haben, in dem Sie schon mal viel offener in Berührung zu solchen Themenstellungen kommen oder vielleicht sogar das Softwaretechnikprojekt vorziehen?! Wir haben derzeit das Problem, dass wir nur 6 Semester haben. Früher mit dem Diplom, mit den 4-5 Jahren, konnte man deutlich mehr machen. Es ist jetzt nicht so einfach ein Auslandssemester zu integrieren. Da soll sich jetzt einiges tun, dass man ein sogenanntes Mobilitätsfenster reinbringt, und somit in einem Semester ausschließlich Wahlpflichtfächer hat und so leichter mal ins Ausland gehen kann.

Ich habe selbst mal in meinem Softwaretechnikprojekt gefragt, ob Interesse besteht, Auslandssemester innerhalb des Bachelorstudiengangs zugänglicher zu machen und so Erfahrungen (besonders bezüglich eigenständiges Arbeiten) zu sammeln. Von den 10 Personen hat nur einer sich gemeldet und ausgesagt, dass er nicht so viel Wert drauf legt. Die anderen haben gar nichts gesagt. Man weiß nicht ganz genau woran das liegt - vielleicht an den organisatorischen Schwierigkeiten? Oder an der Finanzierung? Ob die Studierenden nur schnell das Studium abschließen wollen und Geld verdienen wollen? Oder es einfach nicht nötig ist, sowas vorzuweisen, da die Arbeitslage sehr gut ist? (Vorteil derzeit nicht erkennbar)

Man kann die Bachelorarbeiten in Firmen machen oder man kann sie intern in der Fachhochschule machen. Früher war es sehr üblich extern zu machen. Jetzt, wo die Fachhochschule viel Forschung vor Ort hat, ist es sehr üblich interne Bachelorarbeiten zu machen.

(Fokus wird auf Applikation gelegt, was die Aufgabe der Applikation sein soll. Fokussieren der eigentlichen Idee und die Idee des Bachelor-Seminar vielleicht vorzuziehen)

Ich finde es nicht so gut, dass die Leistungspunkte so „Kuddelmuddel“ sind (Hier sind 8 CP, da sind 5 CP), das gibt immer Probleme mit diesen Anerkennungsverfahren. In anderen Studiengängen, z.B. ITD, hat jedes Modul grundsätzlich 5 Leistungspunkte oder, bei dem großen Projekt, ein Vielfaches davon (10 CP). Das ist so ein bisschen einfacher, als wenn jedes Modul seine Leistungspunkte hat.

Hier (für Softwaretechnik) wurde auch schon einiges herumgeschoben. Es ist vorgesehen, dass man halt ungefähr auf 30 CP pro Semester kommt. Wenn man das Bachelorseminar in das 5. Semester zieht, haben wir im letzten Semester nur noch 25 CP. Dann müssten wir wieder ein anderes Fach einbringen und wenn das ein Fach mit Inhalten zu z.B. formalen Sprachen wäre, dann wären diejenigen benachteiligt, die eine Bachelorarbeit in diesem Gebiet schreiben wollen.

(Abschweifen vom eigentlichen Thema)

Was halten Sie von der Idee der App, haben Sie Anregungen, sehen Sie Chancen oder sogar Risiken?

Günstig finde ich es, dass die Personen an den Zeitplan erinnert werden. Denn mit dem Zeitmanagement hapert es manchmal ein bisschen. Für manche sind es vielleicht unnötige Erinnerungen, aber für andere ist das, denke ich, nicht verkehrt.

Es bleibt die Gefahr, dass wenn die App einen Fortschrittsbalken anzeigt und der bei 100% liegt, die App ja nicht den tatsächlichen Fortschritt prüfen kann und man (der Studierende) sich somit denkt, dass "alles toll" ist, obwohl man zum Beispiel alles vollgeschrieben hat, was jedoch (qualitativ) nicht so toll ist. Vielleicht könnte man ja auch inhaltliche Fragen einbringen. Zum Beispiel wenn man Grafiken hat (Entsprechen die Grafiken einem Standard?). Diagramme wurden schon in eigenen Notationen geschrieben, was zu Fehlern und Missverständnissen führt - ein Zustandsdiagramm, welches fehlerhafte Wege zugelassen hat. Wenn man da eine vernünftige Notation eingehalten hätte, wäre es deutlich besser gewesen.

Man könnte in die App vielleicht eine Checkliste integrieren, die diese Fragen an bestimmten Stellen abfragt. Eine Mögliche Frage wäre, ob bekannte Notationen verwendet wurden oder: "Keine Abbildung soll als selbsterklärend betrachtet werden", dass man so ein paar Punkte als Checkliste herausnimmt. Oder: „Bei jeder Quelle alle Angaben aufgeführt?“. Wenn es eh schon passt und das abgefragt wird, dann macht das ja nichts.

Es wäre natürlich auch nicht gut, wenn diese Abfragen erst kurz vor der Abgabe kommen und der Studierende dann merkt, "Oh, Mist! Ist ja gar nicht...", und alle 20 Grafiken neu überarbeiten muss. Und natürlich sollte diese Abfrage auch nicht dazu führen, dass die Studierenden, "wie bei Softwareinstallationen", sie einfach mit dem immer wieder aufeinanderfolgenden Bestätigen wegklicken. Also sollte es auch eher eine übersichtlich ausgelegte Auswahl an Fragen bleiben. Ich denke, da müssen Sie am besten überlegen in welcher Reihenfolge das ganze kommt. Wenn die erste Grafik kommt, dann sollte die Frage auftreten.

A.1.3. Transkripte der Interviews mit den Professoren (Interview III)

Interview III

Frage: Ihr Fachgebiet wird auf der Internetseite der Fachhochschule Lübeck als "Informatik" angegeben.

Ja, dass ich auf der Webseite der Fachhochschule nichts weiter angegeben habe liegt daran, dass ich da zwei Links habe, wo ich die Informationen regelmäßig pflege. Der Vorteil liegt dabei, dass ich diese Informationen selbst pflegen kann. An der Webseite der Fachhochschule kann ich nichts ändern. Deswegen habe ich damals gebeten, nur die Links auf meine CoSA Profilseite und noch ein weiteren anzugeben. Die aktuellen Sachen passieren auf diesen externen Seiten. Ich will nicht ständig dafür sorgen, dass irgendwas aktualisiert wird. Wenn Sie da auf der CoSA Seite schauen, werden Sie mehr Informationen finden.

Wenn ein Studierender Sie als Betreuer für die Bachelorarbeit wünscht und mit einem Themenvorschlag zu Ihnen kommt, der nicht Ihrem Fachgebiet entspricht, gibt es dann Themen, bei denen Sie den Studierenden an Ihre Kollegen weiterleiten?

Ja, ich betreue relativ breit gestreut. Ich unterrichte ja auch in mehreren Studiengängen. Ich unterrichte bei ITD genauso viel wie in Informatik zum Beispiel. Dadurch bin ich gewohnt, Arbeiten zu betreuen, die eigentlich gar nicht meinen fachlichen Kern betreffen. Aber es ist so: bei Bachelorarbeiten gibt es eben ganz schön viele Dinge, die eben einfach "bachelorarbeitstypisch" sind. Die kann man dann beurteilen, auch wenn es nicht exakt das eigene Fachgebiet ist. Bei einer Promotion muss das das eigene Forschungsgebiet sein. An der Fachhochschule arbeiten wir, denke ich, auf einem Level, bei dem 70 bis 80 Prozent der Professoren das Wissen haben sollten. Wir sind ja nicht alle als Lehrstuhlinhaber eines Spezialgebiets berufen, sondern so ein bisschen sind wir schon Generalisten. Insofern kommt es auch vor, dass ich Studenten an andere Professoren verweise. Der Grund ist aber meistens, dass ich nicht noch mehr Leute betreuen kann, aber es kam auch schon einmal vor, dass es aus fachlichen Gründen war. Das war dann ein Bereich, wo ich mich wirklich nicht so gut auskannte und dann Kollege X oder Kollege Y empfohlen habe. Das kommt vor. Aber dafür gibt es ja auch dieses Vorgespräch. Bei mir läuft es normalerweise so, dass Leute mich dann ansprechen, weil sie gerne die Bachelorarbeit bei mir schreiben wollen und dann treffen wir uns, um Interessenschwerpunkte zu besprechen und am Ende des Gesprächs ist dann für beide Seiten klar, ob das sinnvoll ist oder ob das Thema vielleicht jemand anders betreuen sollte.

Lassen sich Bachelorarbeiten grob nach Typen kategorisieren? Was sind da Ihre Erfahrungen? Stichwort - "entwickelnd" oder "vergleichend".

Ja. Man muss da auch nochmal unterscheiden zwischen internen und externen Arbeiten. Das kann auch noch große Unterschiede machen.

Es gibt die entwickelnde externe Arbeit. Das ist ganz klar eine Kategorie, welche bedeutet, dass die Person, die eine Arbeit schreibt, in einer Firma sitzt oder zumindest für eine Firma primär arbeitet. Das bedeutet auch, dass das eigentliche Ziel, das dabei rauskommt von dem Unternehmen gesteckt wird und in dem Fall sehe ich mich als Betreuer eher in der Position eines Projektleiters oder Projektmanagers. Ich mische mich da dann sehr selten in die inhaltlichen Ziele ein, solange ich das Gefühl habe, dass die Arbeit das Niveau einer Bachelorarbeit erfüllt. Also wenn unsere Leute dann als Arbeitssklaven eingesetzt werden, muss ich das verhindern. Aber wenn die dann genau etwas brauchen, werde ich nicht nein sagen. Also in diesem Fall ist meine Rolle eine ganz spezielle Rolle, weil man dann mit der Firma klären muss, welche Punkte in die Arbeit reinkommen, weil es eine Bachelorarbeit ist. Das sind oft Dinge, die der Firma eigentlich völlig egal sind. Also ich möchte dann natürlich trotzdem in der Arbeit sehen, warum dieser (ein bestimmter) Weg gegangen wurde und welche Alternativen es gegeben hätte. Das ist den Firmen meistens ganz egal, weil sie das eine Ergebnis haben wollen - und das soll dann auch funktionieren. Aber die Leute, die wir hier in die Welt entlassen, sollen schon in der Lage sein, zu entscheiden, zu begründen, abzuwägen und "ingenieurmäßig" zu denken. Da sollten sie auf jeden Fall sagen können, warum sie sich für zum Beispiel genau ein Framework entschieden haben. Das sind so Aspekte, die den Firmen oft egal sind und nicht im Vordergrund stehen. Da ist das Besondere der Arbeit dann, das man versuchen muss was in die Arbeit hineinzubringen, was keine Praktikantenarbeit ist, sondern den Rahmen einer Bachelorarbeit erreicht. Das ist also insgesamt eine spezielle Sache, die auch nicht mit allen Firmen gleich läuft, aber im Mittel würde ich sagen, sind meine Erfahrungen deutlich positiv. Die meisten Firmen sehen das ein, dass die Arbeit nicht identisch mit dem Produkt ist, was letzten Endes entsteht.

Dann gibt es natürlich die interne Bachelorarbeit, welche bei mir den Schwerpunkt meistens im Softwaretechnik Bereich legt. Ich erwarte in diesem Fall, dass die Leute mit der Methodik arbeiten, die sie hier (an der Fachhochschule Lübeck) gelernt haben. Da kommt dann ein Produkt heraus, von dem man die Eigenschaften kennt, wo man sorgfältig geplant hat und die Anforderungen erhoben hat. Es ist dann wirklich ein Softwareentwicklungsprozess, den die Leute da dann einmal durchlaufen und mir geht es dann meistens nicht primär um die Qualität des Ergebnisses, sondern um den Weg. Da liegt dann der Unterschied zu der Entwicklungsarbeit in Firmen. Da muss dann ein bestimmtes Ergebnis herauskommen und man versucht dann unter den bestimmten Randbedingungen, das dann hinzubekommen. Wenn es intern passiert, habe ich das Gefühl, dass die Leute dann mehr das Interesse daran haben, was den fachlichen Kern ausmacht und wenn man dann am Ende feststellt, dass man das Ziel nicht vollständig erreicht hat und dann erklären kann, warum das passiert ist, dann kann das trotzdem eine sehr gute Arbeit sein. Der Fokus liegt also ganz klar auf der Entwicklung.

Dann gibt es natürlich auch die vergleichenden Arbeiten. Beispielsweise vier prominente Webframeworks, worüber sich die Gemeinden streitet, welches nun das Beste ist. Eben eine vergleichende Studie zu führen - so was kann man natürlich auch machen. Das betreue ich aber eher selten, was aber daran liegen könnte, dass die Leute, die mich fragen, offenbar an solchen Sachen nicht so ein Interesse haben.

Was ich häufig habe - besonders im Bereich Künstliche Intelligenz - ist so eine Art Machbarkeitsstudie. Das wäre noch ein dritter Typus. Zum Beispiel, wie gut die Gesichtserkennungsdienste von Amazon, Microsoft oder anderen funktionieren.

Sind die schon so gut, dass man mit einer einfachen Drohnenkamera auf 10 Meter Entfernung Personen erkennen kann? Das war tatsächlich mal der Inhalt einer Arbeit.

Auch hier war es nicht Ziel der Arbeit, ein fertiges Produkt zu entwickeln. Es war im Kern auch keine vergleichende Studie, sondern es ist mehr die Frage: "Ist die Technologie so weit entwickelt, dass man so was daraus ausbauen könnte?". Das Ergebnis war eben nur ein Prototyp, mit dem dann verschiedene Untersuchungen gemacht wurden. Das wäre dann auch so ein Typus von Arbeit - eben die technische Machbarkeit - die dann in technische Grenzbereiche vordringt, wo der Betreuer vorher meistens auch nicht weiß, ob das funktionieren wird oder ob man auf Hindernisse stößt. Und in

diesem Fall geht es dann ja wirklich weniger darum ein Produkt als Ziel zu erreichen, sondern darum, eine Frage zu beantworten. Man würde am Anfang vielleicht Anforderungen aufschreiben, aber mehr mit dem Hintergrund, ob die sich erfüllen lassen oder nicht. Dadurch kann das eine sehr anspruchsvolle Studie werden, weil man da natürlich auch sehr viel testen muss. Man muss ja sicherstellen, dass es nicht deswegen nicht funktioniert, dass man den Prototyp falsch gebaut hat. Das wäre also auf jeden Fall ein Typus von Arbeit, den es gibt.

Im Studiengang ITD gibt es da ja noch ganz andere Dinge. So was wie eine Studie über die Bedienbarkeit, Usability oder User Experience. Da kommt dann auch die Frage auf, ob man etwas entwickeln kann, dass bisher schon existiert, aber "nicht spannend" genug für die Leute wirkt. Da kann durchaus auch viel Technik für ins Spiel kommen. Da wäre das Ziel auch eher eine Entwicklung oder manchmal auch eine Machbarkeitsstudie, aber mit einem nicht nur technischen Ziel. Ich vermute aber mal, dass Ihr Fokus eher auf Informatik-Arbeiten beschränkt ist?

Genau, aber es ist im Sinne der Erweiterbarkeit ja durchaus interessant. Diese Studien, die Sie angesprochen haben (Machbarkeitsstudien, etc.) gehen ja schon eher in eine forschende Richtung.

(zögern) Forschend wäre vielleicht etwas übertrieben. Ich glaube Machbarkeit trifft es am besten. Da gibt es wenig zu erforschen, sondern man guckt ja nur, ob etwas funktioniert. Aus meiner Sicht ist das eher Entwicklung. Entwicklung heißt ja nicht, dass man wie ein Handwerker nach einem bekannten Rezept etwas zusammenbaut, sondern eher das Rezept erstmals zu finden. Von Firmen wird es immer gerne euphemistisch als Forschung bezeichnet. Sobald die was machen, wo nicht jeder sofort sagen kann ob es geht, nennen die es Forschung (lacht). Aber es ist eigentlich nur etwas mehr als Handwerk.

Unsere Absolventen sollen in der Lage sein für Firmen zu überprüfen, ob es zum Beispiel möglich ist, einen bestimmten Dienst mit einer Sprachsteuerung auszustatten. Das heißt aber nicht gleich, dass man einen Forscher braucht, (lacht) sondern dann braucht man einen Ingenieur, der mit Technologien umgehen kann und weiß, wie man so was testet und ausprobiert. Und wenn er dann herausfindet, dass es funktioniert, dann ist das ja kein Forschungsergebnis, sondern dann hat er die Machbarkeit nachgewiesen. Eine Bachelorarbeit sollte eigentlich auch nicht weiter gehen. Dafür gibt es ja dann die Masterarbeit - um zu versuchen, der Welt eine neue Erkenntnis zu eröffnen. Wenn man methodisch weiterentwickelt, dann forscht man. Das ist aber nicht die Aufgabe einer Bachelorarbeit. Aber trotzdem geht es natürlich schon manchmal so ein bisschen in diese Richtung. Das sind dann häufig Leute, die vorhaben noch einen Master zu belegen, weil man so die Grenze zur Forschung tatsächlich ankratzen kann. Die Leute kriegen dann dadurch so ein Gefühl dafür, was man sich fragt und wo diese Fragen herkommen.

Es ist schwierig das richtige Wort dafür zu finden. Forschung ist es natürlich nicht, aber Recherche wäre dann auch zu wenig.

Sie haben es bereits etwas ausgeführt, aber ich würde gerne nochmal fragen, was Ihre Erwartungen an die Studierenden sind, die Sie betreuen?

In der Bachelorarbeit soll man ja Eigenständigkeit nachweisen, deswegen finde ich es sehr wichtig, dass man sich natürlich regelmäßig trifft und dann alles bespricht, aber im Vordergrund sollten dann da die Arbeitsergebnisse stehen. Also aus meiner Sicht ist eine Bachelorarbeit immer noch eine Art Lehrveranstaltung. Die Leute schreiben eine solche Arbeit zum ersten Mal, also kann ich nicht davon ausgehen, dass sie alles wissen. Aber Sie sollten bereits im Stande sein, eigene Zwischenergebnisse zu erarbeiten, sodass wir dann darüber diskutieren können und ich auf dieser Basis dann weiterhelfen kann. Das heißt, dass das im Studium vorher vermittelte dann einfach schon mal da ist und die Leute

dann auch selber mal auf die Idee kommen, irgendwo nachzuschlagen und sich Dinge anzulesen und selbst merken, welche Dinge sie in Erfahrung bringen müssen. Da möchte man die Studierenden dann nicht jedes Mal darauf hinweisen was sie machen sollen. Die Leute sollten möglichst herkommen und schon Ideen im Kopf haben und sich Rückkopplung einholen - bin ich auf dem richtigen Weg? So stelle ich mir das vor und in der Regel läuft das auch so.

Also sollte es so sein, das man sich trifft um eher die Qualität der Ergebnisse zu reflektieren und so zu überlegen wo die geleistete Arbeit zu wenig oder vielleicht sogar zu viel war?

Genau, auch das kann sehr wichtig sein. Bei der Bachelorarbeit verrennen sich manchmal Studierende in irgendwelchen Seitenzweigen, die sie vielleicht ganz spannend finden, und dann kommt die Gefahr auf, das große Ziel aus den Augen zu verlieren. Dann überlegt man da, wie man die Kurve bekommen kann. Solche Dinge wären die Dinge, über die man (bei einem Treffen) am liebsten spricht.

Was ich von vielen Studierenden gehört habe ist, dass sie zu Anfang extrem viel Zeit in die Grundlagen gesteckt haben und durch Feedback auch darauf hingewiesen worden, das es vielleicht zu viel ist.

Das ist keine Seminararbeit, wo man wochenlang Grundlagen darstellt. Die Grundlagen stellt man ja nicht dar, um zu zeigen, dass man die Grundlagen beherrscht, sondern Grundlagen sollen in der Bachelorarbeit ja dargestellt werden, um zu sehen worauf die Arbeit basiert. Ich empfehle oft, dass man sich für das Grundlagenkapitel erst mal nur Skizzen macht, zum Beispiel durch Stichwortartige Inhalte. Richtig schreiben kann man erst, wenn man den Kern der Arbeit im Griff hat, weil man erst dann weiß, was man in den Grundlagen erklären muss und was eben nicht. Das ist auch so ein Punkt.

Was ich auch wahrgenommen habe ist, dass viele Studierende die Arbeit sequenziell angehen, obwohl sogar auch im Bachelorseminar darauf hingewiesen wurde, dies nicht zu tun. Da wären wir dann schon bei dem nächsten Thema. Diese Probleme, die auftreten und vielleicht sogar immer wieder auftreten. Welche Probleme treten denn immer wieder auf?

Es kommt häufig vor, dass Leute Probleme mit dem Schreiben haben und das ist diesen Leuten eben vorher nicht wirklich bewusst, weil in unserem Studium wenig vorkommt, wo man lange Texte schreiben muss. Meist sind es eher kurz gefasste Dinge, und das etwas Längeres vorkommt ist eher selten. Es fällt vielen Leuten schwer den Fokus beim Schreiben zu behalten - die driften oft weg. Unter anderem war das auch der Anlass, weswegen wir im 5. Semester im Fach Intelligente Systeme angefangen haben diese Berichte einzuführen. Das war nicht von Anfang an so. Ich glaube Sie mussten das auch schon machen oder?

Ja.

Da merken dann alle bei der ersten Aufgabe einen Bericht zu schreiben, weil niemand genau weiß, was da jetzt eigentlich reingehört. Und bei der Bachelorarbeit stellen sich dann die Fragen in einem ganz anderen Umfang - Was muss da eigentlich rein? Ich hoffe dadurch zu erreichen, dass die Leute zumindest für einen Typus von Arbeit ein Gefühl dafür entwickeln, was es Wert ist reingeschrieben zu werden und was nur "Füllmasse" ist.

Da passiert dann eben das, was Sie auch schon gesagt haben. Man spricht die Gliederung ab und dann fangen die Leute meistens an, von vorne zu schreiben, was oft keine gute Idee ist. Es gibt einige, die bekommen das hin, aber die haben wohl eine literarische Begabung. (lacht) Denen floss das aus der Feder. Den meisten Informatikern fließt das eben nicht aus der Feder und da empfehle ich dann eben, dass sie die Arbeit an allen Stellen wachsen lassen. Man skizziert alles was man

haben will, hat den großen Rahmen und fängt dann an, an mehreren Stellen zu arbeiten. Im Grundlagenkapitel hält man erst mal nur fest, was da rein muss und welche Punkte wichtig sein könnten. Dann fängt man eigentlich mit dem Teil an, welcher einem am leichtesten fällt, nämlich zu beschreiben, was man wirklich gemacht hat. Und während man das beschreibt fällt einem oft ein, was man jetzt gerne so sagen würde und was man vorher eigentlich in den Grundlagen beschreiben muss. Der Inhalt entsteht also so ein bisschen aus der Arbeit selbst. Ich glaube, dass es unmöglich ist das linear entstehen zu lassen, wenn man zum ersten Mal so eine Arbeit schreibt. Diesen großen Plan und diese Weitsicht hat man einfach nicht. Mir selbst fällt es immer noch schwer. Wann immer ich etwas abfasse mache ich das immer noch in diesem Stil. Ich überlege mir zuerst, was ist denn das wichtigste überhaupt an dem ganzen Ding, danach wird mir dann klar, wie ich da heranzuführen muss. Von wo muss ich kommen? Was muss ich erklären, damit ich den Leser dann da habe, wo ich ihn haben will, wenn der interessante Teil kommt.

Ich glaube das ist eine Sache, die heutzutage sowieso wichtig ist. Die Leute haben immer weniger Geduld zu lesen. Wenn zehn Seiten Grundlagen kommen und es dann spannend wird, ist es oft heikel und deswegen ist es wichtig so zu schreiben, dass der Leser von Anfang an das Gefühl hat, dass alles was kommt wichtig ist. Das ist sowieso wichtig. Dieses zielgerichtete Schreiben zu lernen. Das ist dann eben nicht dieser klassische Seminar-Stil, zu zeigen was man alles recherchiert hat und der Leser sich dann fragt, wozu er das alles erzählt bekommt. Deswegen nenne ich das oft am Anfang als eine Option, einfach mal darüber nachzudenken, ob das so der richtige Weg ist und empfehle es, dass die Arbeit an mehreren Stellen wächst. Die Einleitung, denke ich, sollte man als allerletztes schreiben.

Die Arbeit soll am Ende eben wie aus einem Guss wirken und eben nicht so, dass sie sich im Laufe der Zeit entwickelt hat. Am Ende weiß man, wie man die Einleitung verfassen muss, damit sie zur Arbeit passt. Das sind Dinge, die man dann wirklich steuern kann. Damit kann man wahrscheinlich schon mal viel Gutes tun.

Viele haben eben genau dieses Problem. Studierende arbeiten vor sich hin und bekommen immer mehr Angst es dann zu schreiben, sie versuchen dann ganz am Anfang anzufangen, wissen nicht was sie dann schreiben sollen und damit wird die Hürde dann immer höher.

Viele Leute fangen zu spät an. Das wäre auch noch so ein Punkt. Sie unterschätzen dann die Zeit, die man wirklich braucht, um wirklich das, was man nachher druckt und abgibt, entstehen zu lassen. Viele sehen dann im Vordergrund, beispielsweise bei der Entwicklung, die Software entstehen zu lassen und beißen sich dann daran fest, um sich vielleicht sogar selbst mit Arbeit zu beschäftigen, die einen davon abhält mit dem Schreiben anzufangen.

Wenn man es schafft die Leute dazu zu bekommen von Anfang an das Gerüst der gesamten Arbeit vor sich zu sehen, würde es glaube ich so manche "Fast-Katastrophen" verhindern, wo Leute erst drei Wochen vor der Abgabe anfangen den Text zu schreiben. Das kann eigentlich nicht gut gehen, wenn man nicht gerade eine Ausnahmebegabung hat, was Schreiben angeht. Deswegen ist es glaube ich gut, in die Breite zu gehen. Sich klar zu machen, dass es nicht nur um die Software geht, die man entwickelt, sondern dass es letzten Endes hauptsächlich um die Darstellung geht und man deswegen immer die Arbeit als Ganzes im Auge behalten muss, ohne sich am Text fest zu beißen. Da kann man wahrscheinlich viel Gutes mit tun, mit dem, was Sie da vorhaben.

Lustigerweise wird es ja alles immer wieder erwähnt. Es gibt das Bachelorseminar, wo auch viele Tipps und Hinweise gegeben werden, aber trotzdem machen alle die gleichen Fehler. Es ist sehr interessant warum es denn nicht funktioniert. Vielleicht kommen die Studierenden mit der eigenen Vorstellung in die Quere. Einleitung und Grundteil hören sich an wie etwas, wo man ansetzen könnte.

Ich glaube es ist sehr schwierig bei so etwas wie der Bachelorarbeit die Ratschläge in einem Paket zu geben und dann zu glauben, dass es die Leute verinnerlicht haben und so anwenden werden. Ich

glaube, dass so etwas nur funktionieren kann, wenn es arbeitsbegleitend stattfindet. Zu verstehen warum etwas eine gute Idee ist verinnerlichen die Leute erst, wenn sie das Problem wirklich haben. Das ist so ähnlich wie mit dem Projektmanagement. Da kann man sich natürlich hinstellen und betonen, was die üblichen Fehler sind, aber eins ist klar: wenn man die Leute auf ein Projekt loslässt, werden sie alle genau diese Fehler machen. So funktionieren Menschen irgendwie nicht. Es gibt Dinge, die sind nicht lehrbar indem man sie erklärt, sondern sie sind nur lehrbar, indem man es die Leute machen lässt und sie dabei begleitet. Es gibt auch andere Dinge, die man sehr gut so lehren kann. Man erklärt etwas und es macht klick weil sie verstanden wurden. Aber es gibt einfach Dinge, die man wirklich erst versteht, in dem Moment in dem man sie lernt und das ist in vielen Bereichen so. Das kennt man auch aus dem Sport. Wenn man den Kindern im Training erklärt, was man im Spiel machen soll, hören alle Kinder sehr aufmerksam zu, aber im Spiel machen alle genau den einen Fehler, wobei man vorher stundenlang erzählt hat, dass sie es auf keinen Fall so machen sollen. Ich hab einige Jahre als Jugendtrainer im Fußball gearbeitet und daher weiß ich, dass es die echt Situation ist, in der es plötzlich Klick macht.

Ich glaube, dass es bei der Bachelorarbeit auch so ist. Erst wenn die Leute an ihrer eigenen Bachelorarbeit sitzen und die Zeit tickt ist die Aufnahmebereitschaft da, zu erkennen wo da der Sinn liegt. Deswegen glaube ich, es könnte eine tolle Sache sein, wenn das mit ihrer Arbeit gut läuft, da das Ziel ja ist, die Leute genau im Prozess zu begleiten. Das wird ja kein Tutorial, wo man ein Video anguckt und dann werdet ihr alle tolle Arbeiten schreiben. Genau das kann nicht funktionieren, meiner Ansicht nach. Man braucht etwas, was einen genau in der Situation begleitet, wo man das Problem hat. genau da muss man ansetzen und dann, glaube ich, kann man eben solche Sachen verhindern. Dann braucht man sich auch gar nicht mehr fragen, warum die Leute das nicht verstanden haben, sondern ich glaube, dass viele der Probleme gar nicht mehr auftreten werden, wenn man sie in dem Moment richtig begleitet. Das wäre so meine Hoffnung.

Deswegen würde ich gar nicht so viel darüber brüten woran das liegt, sondern ich glaube es ist eher diese Trennung. Die Leute sind einfach noch nicht in dieser Situation und es ist, wie Leuten das Schwimmen zu erklären, bevor sie das erste Mal das Wasser berühren.

Da kann man auch nicht einfach sagen, dass es einfach ist sich an der Oberfläche zu halten und dass man eigentlich kaum etwas dafür machen muss. Das verstehen die Leute alles, aber in dem Moment wo sie im Wasser sind, ist die Situation eine völlig andere und die fangen dann an herum zu hampeln und vergessen dabei völlig, dass man eigentlich gar nichts machen muss. Deswegen braucht man eben die Person, die am Beckenrand steht und guckt was da passiert.

Also würden sie dieser Applikation, oder der Idee, positiv gegenüberstehen? Oder als sinnvoll ansehen?

Die Idee ist eine sehr sinnvolle. Ob es dann im ersten Anlauf gleich funktioniert weiß man nicht. Sie probieren es ja eben zum ersten Mal. (lacht)

Manchmal scheitern Dinge ja auch an Kleinigkeiten. Aber als Ansatz überhaupt klingen die Dinge für mich sehr viel vielversprechender, als vorher die Einweisung zu machen und dann zu hoffen, dass die Leute es richtig machen. Da sehe ich die Chancen auf jeden Fall hier viel größer, das so was (die Leute) voranbringen kann.

Ich sehe das als sinnvoll an. Zumindest so sinnvoll, das es sich lohnt, da mal ein Versuch zu machen.

Das Problem ist ja in meinem Fall, dass es natürlich auch meine Bachelorarbeit ist, und ich ...

Ja, sie haben natürlich die Probleme, die jeder hat, und zusätzlich erwartet man Wunder. (lacht)

Genau. Die Zeit reicht letzten Endes ja gar nicht aus, um das Konzept ordentlich auszuwerten und zu sehen, ob es funktioniert. Man müsste ja eigentlich mindestens eine ganze Iteration erst mal ausprobieren, um zu sehen, ob es überhaupt einen Effekt hat.

Haben Sie sonst noch Anmerkungen oder Ideen, die sie noch mitteilen wollen?

Nö, vielleicht nur eine sehr profane Anmerkung. Wenn man schon mal so etwas macht wäre es vielleicht nicht verkehrt, wenn man in das, was letzten Endes das Tool wird, die Formalien aufnimmt. Auch das steht überall und wird zig-fach erklärt, aber auch das erkläre ich immer nochmal den Bacheloranden - wie es mit den Formalien aussieht, wie es mit der Anmeldung aussieht und wie es mit dem Kolloquium läuft. Es wäre schön, wenn man dann so einen Infoteil mit drin hätte. Die Hoffnung ist natürlich, dass die Leute es am Ende annehmen und nutzen wollen. Wir hätten dann eine Stelle, wo sie freiwillig hingucken. Wenn man jetzt sagt, das steht doch in irgendwelchen Lernraum-Kursen, dann sagen alle, dass sie wissen, dass es irgendwo steht, aber (es ist zu aufwändig zu finden).

Wenn es aber jetzt eine Stelle gibt, von der man weiß, die Leute die Interesse haben, werden sich diese Stelle angucken und da dann diese Sachen stehen, dann wäre das Klasse.

Das Besondere ist ja, dass ich in diesem Fall ja auch in die Zielgruppe falle. Und da spiegelt sich meine eigene Erfahrung auch wieder und deckt sich mit den Erfahrungen, die ich von anderen Studierenden gehört habe. Jeder weiß, dass es irgendwo steht, aber man weiß gar nicht so richtig was und wo man es suchen muss.

Wo muss ich suchen, wie heißt das Ding, was ich suche und warum soll ich überhaupt ein Formblatt suchen, wenn man nicht mal weiß, dass man es überhaupt ausfüllen muss. Das ist im Moment total diffus verteilt und das an Orten, wo die Leute aus eigenem Antrieb gar nicht gucken und deswegen wäre das eine geeignete Stelle um diese Dinge zu bündeln oder zumindest die Links zu finden. Es muss ja nicht nochmal da dupliziert sein. Aber das eine Stelle da ist, wo einem das Tool dann auch gleich sagt, welche Dinge alle wichtig sind und wo man die Hinweise findet und was für Hinweise man gar nicht findet, die dann aber explizit genannt werden. Das fände ich klasse.

Ansonsten fällt mir eigentlich nichts weiter ein. Ich denke es wird Dinge geben, die man gut generisch vorbereiten kann, man sollte sich aber auch bewusst sein, dass Arbeiten immer auch einen individuellen Charakter hat und das es Dinge gibt, die man nicht so allgemein abhandeln kann. Das ist, glaube ich, sonnenklar. Deswegen kann der Versuch nicht der sein, es in seiner Gesamtheit abzuhandeln, sondern eher - und ich glaube auf der Spur sind Sie auch - was die Grundprobleme sind. Was ist allen gemeinsam? Wo kann man ansetzen, um Alle voran zu bringen. Man sollte auf jeden Fall vermeiden die Leute mit irgendetwas zu gängeln. Also das man zu sehr versucht die Leute in irgendwelchen Spuren zu halten, die dann nicht zu jeder Arbeit passen. Aber mehr als so eine diffuse Äußerung kann ich auch nicht machen, weil ich das auch nicht weiß. Das muss man dann mal ausprobieren und mal gucken, wie das dann ist. Da muss man einfach Mut haben.

Ja, mir ist auch klar, dass man nicht so einfach den Weg gehen kann und die App "fragen" kann, was man machen soll und dann weiß was man braucht und machen muss. Aber ich bin auf jeden Fall dabei, zu untersuchen wo die Probleme liegen, um von da aus dann die Schwachstellen zu finden und zu gucken, was man verbessern kann.

Ja, da gibt es dann ja viele Formalien, die vielleicht gar nicht so kompliziert sind, bei denen man unterstützt werden kann, wie das Erstellen eine Literaturverzeichnisses. Das man diese Sachen dann schon mal sammelt und dann eben schon mal so eine Liste hat. Also es gibt da ganz viel auf der handwerklichen Ebene, wo so ein Tool auf jeden Fall unterstützen kann - jeden, ganz egal wie die Arbeit ist - aber auch bei den Sachen, die in Richtung der Art der Vorgehensweisen gehen, gibt es sicherlich auch allgemeine Sachen. Ich glaube, da haben sie schon exakt die richtigen Fragen gestellt.

Für mich klang es so, als wären Sie mit Ihren Fragen auf der richtigen Spur.

A.1.4. Transkripte der Interviews mit den Professoren (Interview IV)

Interview IV

1

2 **Frage: Ihr Fachgebiet wird auf der Internetseite der Fachhochschule Lübeck als „praktische**
3 **Informatik“ angegeben. Wenn ein Studierender Sie als Betreuer für die Bachelorarbeit**
4 **wünscht, und mit einem Themenvorschlag zu Ihnen kommt, der nicht Ihrem Fachgebiet**
5 **entspricht, gibt es dann Themen, bei denen Sie den Studierenden an Ihre Kollegen**
6 **weiterleiten?**

7

8 **Antwort:**

9 Ja, da gibt es Themen. Alle Themen, die ich nicht selber fachlich kompetent erschöpfen und betreuen
10 kann. Die sollten schon in meinem fachlichen Bereich liegen - den setze ich breit an - aber
11 angenommen da kommt jetzt jemand mit der Anforderung "Spielprogrammierung mit dem Fokus auf
12 künstlerischer Gestaltung", dann gibt es hier kompetentere Betreuer.

13 Wenn ich einen Betreuer ausmachen kann, der deutlich kompetenter wäre hier an der FH, würde ich
14 erstmal sagen: "gehen Sie mal dorthin und fragen Sie mal, wenn nicht, kommen Sie wieder zurück,
15 dann schauen wir weiter.". Also da wird jetzt keiner komplett weg geschickt, nur nochmal rumhören,
16 ob es einen vielleicht besseren Betreuer gibt.

17

18 **Also würden Sie schon sagen, dass man hier etwas abgrenzen kann?**

19

20 Ja, das hängt immer vom konkreten Thema ab, das ist nicht so fest als Grenze definiert. Die ist
21 natürlich weit - ein Graubereich. Eine Variante wäre noch, als Zweitbetreuer einen Kollegen aus
22 einem anderen Fachbereich hinzuzuziehen.

23

24 **Kann man bei Bachelorarbeiten, die Sie betreuen, grob gesagt "eine Art" identifizieren?**

25

26 Die Mehrheit hat eine entwickelnde Natur. Das ist aber daraus begründet, dass Studierende etwas
27 entwickeln wollen. Die wenigsten kommen zu mir und sagen, sie wollen irgendwas vergleichen und
28 einen Bericht schreiben. Das habe ich nur in extremen Ausnahmefällen erlebt.

29 Die Aufgaben die ich vergeben habe, lassen sich im allgemeinen immer auf ein verteiltes System
30 abbilden.

31

32

33 **Wo liegen aus Ihrer Sicht die Prioritäten bei dem Bearbeiten der Bachelorarbeit. Was**
34 **verlangen Sie von dem Studierenden? Wo liegen für Sie die Punkte bei der Bachelorarbeit, wo**
35 **Sie sagen würden: "Das ist der Inhalt der Bachelorarbeit"?**

36

37 Methodische Vorgehensweise, Systematische Vorgehensweise unter Anwendung der Kenntnisse die
38 die Studierenden im Laufe des gesamten Studiums gelernt, gehört und praktiziert haben. Angefangen
39 bei der Anforderungsanalyse bis hin zum Test.

40

41 Die Vorgehensweise, die den üblichen Modellen entspricht, die ausgiebig diskutiert wurden. Also nicht
42 einfach anfangen zu codieren. Das wäre weit weg von unserem Studiengang
43 "Informatik/Softwaretechnik". Entsprechend dem Stand der Kunst, den wir hier lehren.

44

45 Die Herangehensweise an das Problem. Wie nähere ich mich dem Problem? Wie zerlege ich das
46 Problem, weiß ich, was eine Anforderungsanalyse ist? Ein Pflichtenheft?

47

48 **Das geht ja jetzt schon mal in die Richtungen der Erwartungen an den Studierenden. Was**
49 **erwarten Sie denn von dem Studierenden im Allgemeinen?**

50 **Dazu zählt zum Beispiel die Kommunikation.**

51

Entscheidend ist wirklich die Vorgehensweise, wie ich vorhin bereits betont habe, welche sich am Softwarelebenszyklus orientiert. Dass man vergleichende Analysen durchführt, wenn es notwendig ist. Dass man seine Entscheidungen begründen kann - sei es die Auswahl der Programmiersprache, die Auswahl der Softwarearchitektur - im Vergleich zu denen, die man nicht verwendet hat, und die Bewertung gegenüber den Alternativen.

Und zum Stichwort Kommunikation? Was erwarten Sie, wenn der Studierende zu Ihnen kommt. Nach dem Motto "So ich bin jetzt zu Ihnen gekommen aber ich hab noch nichts aufgeschrieben. Was soll ich tun?"

Dann trete ich mit dem Studierenden in die Aufgabenfindungsanalyse ein und das schon bei dem genauen Formulieren der Aufgabenstellung. In den seltensten Fällen bekommt der Studierende eine fertige Aufgabenstellung, da Sie das ja im späteren Berufsleben auch machen müssen: Aufgaben definieren, abgrenzen und Anforderungen schreiben.

Die erste Phase ist es, die Aufgabenstellung zu formulieren.

Das kann durchaus lange dauern, dass wir eine Aufgabenstellung haben. Das geht über mehrere Treffen hinweg. Darauf lege ich letzten Endes den Fokus, dass dort genau auch darauf geachtet wird, was der Studierende von sich selbst letzten Endes fordert.

Ist das erfüllbar? Wie kann er es bewerten?

Und was die Kommunikation angeht, dass er sich regelmäßig meldet und nicht wartet bis er kontaktiert wird. Er MUSS selbst proaktiv den Kontakt zu seinem Betreuer suchen. Dass er, wenn er Probleme hat, auch möglichst schnell den Kontakt zu mir sucht. Das verstehe ich auch unter selbständigem Arbeiten.

Gibt es denn typische Probleme die während der Arbeit mit den Studierenden (immer wieder) auftreten?

Ja, die gibt es ganz naturgemäß. Das ist halt die erste größere Arbeit, die die Studierenden schreiben.

Bei der Programmierung gibt es interessanterweise nicht so typische Probleme, die sich wiederholen, eher bei der Ausfertigung der Arbeit.

Zum Beispiel die Forderung, dass Aussagen begründet werden müssen unter Bezugnahme auf anerkannter Fakten, Publikationen oder selbst durchgeführte Analysen. Das sind Erkenntnisprozesse, die fast jeder durchmacht. Bei der Aussage "Das Programm ist schnell": warum ist es schnell, in welchem Sinne ist es schnell; oder es wird einfach nur gesagt, die eine Programmiersprache ist besser geeignet als die andere - Warum? Aus welchen Gründen? Was sind die Kriterien?

Da haben fast alle Studierenden ihre Probleme, bis hin zu Details und Fakten, das ordnungsgemäße nachvollziehbare (seitenweise) Zitieren von Quellen, oder im Internet Permanent Links, die nachvollziehbar sind. Dann erkennt der Studierende erst, warum er die Arbeit so gestalten muss - damit sie für einen Nachfolgenden komplett nachvollziehbar ist.

Das ist glaube ich das, was zum Schluss rauskommt - warum schreiben wir die Bachelorarbeit in der Art und Weise, wie wir sie schreiben? - Weil sie eine wissenschaftliche Arbeit ist, die in all Ihren Begründungen nachvollziehbar sein muss für einen externen Beobachter. Das ist eine neue Erkenntnis, die so noch nicht erarbeitet werden konnte im Studium.

Aber auch das Projektmanagement, das Selbstmanagement, das Zerlegen in Arbeitspakete. Das bekommen einige Studierende besser hin als andere Studierende. Das Schätzen von Zeitaufwand ist generell eine große Schwierigkeit bei Softwaretechnik-Projekten in der Praxis. Da gibt es auf jeden Fall Unterschiede. Das wurde im Studium ja auch schon stellenweise behandelt - das merkt man auch.

Wichtiger Rat, den ich immer mitgebe: "Beginnen Sie mit der Dokumentation direkt während der Arbeit von Anfang an - sonst wird es später sehr schwer und kommt an dieser Stelle dann zu Blockaden"

Aber, was gut funktioniert ist das Verständnis für den Softwareentwicklungszyklus, das kriegen sie gut hin. Alle wissen, es geht mit der Anforderungsanalyse los, über die Entwurfsphasen bis hin zum Testen. Beim Testing wird es wiederum schwierig, es kann sehr Aufwändig werden ein System zu testen, gerade, wenn es ein Benutzerinterface hat.

Dann muss ich ja Nutzer davor setzen und eine Nutzeranalyse machen. Das sprengt dann den Rahmen. Aber Testen ist immer ein Problem - vielleicht aus Zeitgründen: nicht früh genug Modul- und Unit-Tests geschrieben.

Ich vermute mal, das liegt daran, dass wir an der Fachhochschule kein Wahlpflichtmodul zu dem Thema Testing anbieten. Das ist insgesamt ein Schwachpunkt, der nicht von den Studierenden verschuldet ist. Da wünsche ich mir auf jeden Fall, dass das gemacht wird.

Können Sie aus Ihrer Sicht den Zeitpunkt konkretisieren, wo der Studierende den größten Erkenntnisgewinn hat - also merkt: "Oh, das ist das was von mir verlangt wird!"?

Ich glaube, dass es zu dem Zeitpunkt ist, bei dem ein bestimmter Umfang an Dokumentation geschrieben ist. Spätestens, wenn die Inhaltsstruktur aufgebaut werden soll. Das ist ja auch eins der Themen, die man mit dem Studierenden bespricht. Wie ist der Aufbau der Arbeit? Die Kapitelstruktur? Ich glaube, zu diesem Zeitpunkt dämmert es. Das ist die erste große Schreibaufgabe. Gar nicht mal bei dem Programmieren, das haben Sie im Laufe des Studiums ja schon häufig gemacht. Spätestens beim ersten Feedback zu einem geschriebenen Text und ich die Punkte nochmal bringe, die ich schon vorher angemerkt habe.

Wobei ich sagen würde, dass sich vieles gebessert hat durch das Bachelorseminar.

Ich muss nicht mehr so viel sagen, was ich früher bei der Diplomarbeit sagen musste. Ich muss zwar alles nochmal sagen, aber nicht alles nochmal neu erklären. Es ist definitiv eine gute Hilfe. Es ersetzt zwar nicht die Beratung - ich muss vieles wiederholen - aber man merkt, die Studierenden haben davon schonmal was gehört.

(Bezug auf die Applikation) Glauben Sie, dass es so möglich ist, Studierende zielgerichteter zu unterstützen und die vielleicht sogar die Fragebereitschaft zu erhöhen?

Auf jeden Fall, viele Beratungsaspekte wiederholen sich im Laufe der Jahre, und da wäre es schon sehr Hilfreich auf die Applikation zu verweisen, weil man somit einen ganz neuen Diskussionshintergrund hätte. Nach dem Gespräch ist der Kontakt ja erstmal unterbrochen für eine Weile, dann kommt der selbstständige Arbeitsabschnitt. Und für den Zeitabschnitt benötigt der Studierende meiner Meinung nach auch nochmal einen Leitfaden, deswegen wäre das in meinen Augen als Web-App sehr sinnvoll, auf die wir als Betreuer auch verweisen können.

Früher hatten wir nur einen Termin, es gab dann zum Ende des 5. Semesters einen Beratungstermin, wo alle Studierenden gekommen sind.

155 Da haben wir kurz vorgestellt, was eine Bachelorarbeit eigentlich ist, die Studierenden konnten
156 Fragen stellen - daraus hat sich auch das Bachelorseminar entwickelt.

157

158 Daher wäre Ihre Web-App an dieser Stelle unbedingt notwendig.

159

160

161 **Sehen Sie Probleme, die durch die Applikation entstehen könnten?**

162

163 Ja, ganz klar die Wartung der Applikation. Sie schreiben Ihre Bachelorarbeit und haben dann Ihre
164 Note und haben einen lukrativen Job. Aber wer kümmert sich dann um die App?

165 Es muss natürlich gewartet werden, sonst ist es am Ende nur ein totes Konstrukt, was keiner mehr
166 nimmt.

167

168 **Sehen Sie Risiken durch die Applikation bei der Nutzung von den Studierenden?**

169

170 Ja, es könnte sein, dass die Studierenden sich festbeißen und bestimmte Punkte aus Ihrem
171 "Leitfaden" als unumstößlich ansehen und somit in den Konflikt mit den Betreuer kommen, wodurch
172 die Studierenden in eine Abwehrhaltung geraten. Es (Leitfaden) sollte also immer noch als
173 interpretierbar - als Ratschlag - gelten.

174

175 Es sollte klar erkennbar sein, dass es nicht so gedacht ist. Es kann ja auch vorkommen, dass es bei
176 der App zu inhaltlichen Fehlern kommen kann, welche durch Wartung und Einsatz erst aufgedeckt
177 werden können.

178

179

180

A.1.5. Transkripte der Interviews mit den Professoren (Interview V)

Interview V

Frage: Ihr Fachgebiet wird auf der Internetseite der Fachhochschule Lübeck als "Software Ergonomie, Mensch-Maschine-Interaktion, Softwaretechnik" angegeben. Würden Sie sagen, dass die Bachelorarbeit üblicherweise in diesen Fachgebieten stattfinden oder halten Sie das eher offener?

Antwort: Überwiegend ja, allerdings ist das Themenspektrum - gerade auch in der Medieninformatik, wo ich viel lehre, und auch in ITD relativ breit - und da kommen da auch mal Fragestellungen, die dann eher am Rande liegen. Aber klar! Die Hauptexpertise liegt ja dann in dem Bereich und wenn Anfragen zu Themen kommen, mit denen ich eher gar nichts zu tun habe, würde ich die dann auch eher ablehnen, wenn jetzt nicht zwingende Gründe dafür sprechen, wie zum Beispiel, dass derjenige niemand anderen findet oder ganz verzweifelt ist. Üblicherweise liegen die (Themen) in diesem Themenschwerpunkt.

Es ist ja eher schwierig einen konkreten Typus von Bachelorarbeiten zu definieren - was ich meine ist so was, was in der Softwareentwicklung eine "entwickelnde" Arbeit ist. Kann man Bachelorarbeiten aus Ihrer Sicht (grob) kategorisieren?

Das sind verschiedene Typen zum Kategorisieren. Zum einen Softwareentwicklungsprojekte, wo im Mittelpunkt steht möglichst auch funktionsfähige Prototypen - meistens sind es ja Prototypen - aber funktionsfähige Applikationen zu entwickeln. Auch dann durchaus den Fokus auf der technischen Umsetzung. Häufig aber auch mit Designaspekten dabei.

Dann gibt es eher die design-lastigen Arbeiten, die kommen häufig aus ITD, wo es nicht darum geht, funktionsfähige Prototypen zu entwickeln, sondern verschiedene Interfacedesigns oder Design-Alternativen auszuprobieren und dann zu evaluieren, zum Beispiel hinsichtlich Ihrer Wirkung, was kommt besser an? Dann gibt es auch Arbeiten, die in Richtung Corporate Design oder Ähnliches gehen, wo beispielsweise ein neuer Webauftritt entwickelt wird, mit neuem Logo, mit neuen Informationsmaterialien, wie zum Beispiel ein Flyer oder Ähnliches, für eine Organisation beispielsweise. Das wären so die ITD-Arbeiten. Teilweise sind es eben auch wissenschaftlich geprägte Arbeiten, wo es dann auch eine ausgedehnte Literaturrecherche gibt und man mit Hilfe von empirischen Methoden, wie beispielsweise Fragebogen- oder Laboruntersuchungen, bestimmte Fragestellungen beantworten möchte.

Sie sind nun schon auf die Aufgabenstellung im Detail eingegangen. Könnten Sie auch etwas zu den Erwartungen an die Studierenden im Allgemeinen sagen? Was soll der Studierende im Rahmen seiner Bachelorarbeit leisten?

Also wichtig ist ja bei einer Bachelorarbeit zum einen, dass es ein eigenständiges größeres Projekt ist, das man in einer überschaubaren Zeitspanne bearbeitet. In den anderen Projekten, die sie im Studium haben, sind das dann ja häufig noch Gruppenarbeiten. Auch wenn da schon größere Projekte bearbeitet werden, macht man das selten alleine. Kommt auch ab und an mal vor, je nach Studiengang. In der Bachelorarbeit haben Sie zum ersten Mal die Herausforderung, dass sie allein verantwortlich eine Fragestellung entwickeln, die Methoden planen, das Vorgehen planen, die Umsetzung in die Tat umsetzen und das entsprechend verschriftlichen.

Diese Eigenständigkeit ist ein Merkmal von (diesen) Arbeiten. Es ist in meinen Augen nicht die Aufgabe des Betreuers jeden Schritt vorzugeben und zu sagen, wie Sie das machen müssen, denn dann wäre es keine eigenständige Arbeit mehr. Klar, als Betreuer begleitet man das, gibt

Hilfestellungen, gibt Ratschläge und Feedback, aber letzten Endes liegt die Verantwortung für die Umsetzung immer beim Bearbeiter selber.

Das andere, was ich wichtig finde, ist, dass es eine wissenschaftliche Arbeit ist. Das kann von der Herangehensweise unterschiedlich sein. Je nachdem, ob ich einen Schwerpunkt in der Softwareentwicklung habe oder ich wirklich eine wissenschaftliche Fragestellung untersuche. Nichtsdestotrotz ist die Vorgehensweise so, dass ich mir nicht einfach selbst was aus den Fingern sauge, sondern mir eine sinnvolle Grundlage erarbeite, anhand von Literaturrecherche oder eben auch indem ich mir verwandte Systeme anschau und gucke, wie da bestimmte Dinge umgesetzt sind. Also dass ich mir erst einmal eine Grundlage lege und systematisch mein Vorgehen plane und das auch transparent und überprüfbar umsetze. Dass ich jeden einzelnen Schritt auch begründen kann und dass das nicht einfach so aus dem Bauch heraus passiert. Das ist unabhängig davon, wie die Fragestellung ist und unabhängig davon, wie die konkrete Methodik ist. Das kann sehr unterschiedlich sein. Es ist halt eine Grundvoraussetzung für das wissenschaftliche Arbeiten, dass ich begründet und nachprüfbar vorgehe.

Was ist denn wenn ein Prototyp entwickelt werden soll und dieses Ziel nicht erreicht wird, der Weg dahin aber korrekt und begründet dokumentiert ist. Liegt der Fokus dann eher auf dem Weg dahin?

Das kann passieren, denn das ist letzten Endes die Natur von Wissenschaft. Ich stelle Hypothesen auf oder habe bestimmte Vermutungen oder Fragestellungen, die ich sicherlich auch gut begründen muss. Die können ja auch nicht aus der Luft kommen. Es kann sich aber trotzdem herausstellen, dass die Hypothese falsch war. Das macht dann nicht unbedingt das Ergebnis wertlos, denn auch daraus kann ich dann ja eine Erkenntnis ziehen. Insofern, wenn ich ein Vorgehen plane, das gezielt und begründet umsetze, und am Ende merke, aus bestimmten Gründen habe ich die falsche Vorgehensweise gewählt, kann ich daraus jede Menge Lernerfahrung ziehen und das ist auch eine wertvolle Erkenntnis für andere Leute, die dann dieselben Fehler nicht noch einmal machen müssen oder von einer anderen Grundlage starten können. In diesem Sinne ist das dann keine gescheiterte Arbeit.

Gibt es denn Fehler und Probleme bei der Bearbeitung der Bachelorarbeit, die immer wieder auftreten? Wo liegen allgemeine Probleme, gerade in Bezug auf die Erwartungen, die Sie angesprochen haben.

Also ich glaube ein grundsätzliches Problem - nicht nur bei Bachelorarbeiten, sondern auch bei umfangreichen Projektarbeiten - ist die mangelnde Zeitplanung. Das haben Sie auch schon häufig beobachtet im Studium, vielleicht nicht bei sich selber, aber bei anderen, dass man Dinge gerne nach vorne schiebt, gerade wenn das nicht so in kleine Häppchen verteilt ist, wo ich genau weiß (wann ich welche Aufgabe habe). Dann schiebt man das gerne vor sich her und am Ende kommt man dann nicht mehr mit der Zeit aus, die man noch hat.

Was ich auch empfehle ist zum Beispiel frühzeitig mit dem Schreiben anzufangen. Das wird zwar unterschiedlich gehandhabt. Ich glaube aber, dass es nicht hilfreich ist, wenn ich zweieinhalb Monate was tue und in der verbleibenden zwei Wochen alles nochmal schnell aufschreiben und dokumentieren will, was ich gemacht habe. Das ist zum einen viel mehr Aufwand, weil ich mir vieles nochmal neu erschließen muss, wenn ich zum Beispiel eine Literaturarbeit gemacht habe, und das dann zwei Monate liegen lasse, dann muss ich im Grunde nochmal alles lesen, wenn ich das dann aufschreiben will. Also dieses zeitnahe Verschriftlichen und gutes Dokumentieren ist ein Erfolgsfaktor, beziehungsweise kann umgekehrt dazu führen, dass man sehr viel mehr Arbeit hat.

Dann natürlich auch unklare Absprachen. Das passiert häufiger mal, wenn die Bachelorarbeiten in Unternehmen geschrieben werden. Das man sich dann vielleicht nicht genau geeinigt hat, was die Erwartungen sind oder dass das Unternehmen vielleicht auch unrealistische Erwartungen an die Bachelorarbeit hat. Es ist wie gesagt eine wissenschaftliche Arbeit, es kann eben sein, dass da was anderes herauskommt als man eben denkt. Dann noch, das (von dem Unternehmen) teilweise sehr stark eingegriffen wird und man somit gar nicht mehr diese freie Methodenwahl hat, die man eigentlich braucht, sondern dass das Unternehmen dann eben vorgibt, wie es gemacht werden soll oder Ergebnisse umsetzen will, wobei es gute Gründe geben würde es anders zu machen. Dann ist man als Bachelorarbeiter, der vielleicht auch ein Angestelltenverhältnis in der Firma hat, in einer schlechten Position, weil man kaum die Freiheit hat, sich beispielsweise gegen den Chef zu stellen. Dann muss man sehr behutsam Lösungen finden, wie man das dann zumindest auch nochmal reflektieren kann in der Arbeit, wo eben Entscheidungen anders getroffen werden mussten, als man es getan hätte. Manchmal ist es eben auch so, dass Kunden sich für eine Designlösung entscheiden, die eigentlich schlecht ist. Wo man dann selber sagt, dass es eine schlechte Lösung ist, der Kunde aber darauf besteht, dass es so gemacht werden muss, hat man in so einem Vertragsverhältnis nicht die Möglichkeit es anders zu machen. Das sind Dinge, die muss man dann reflektieren und das kann dann schon zu Schwierigkeiten im Prozess führen.

Würden Sie sagen, dass Unternehmen den Bacheloranden bewusst (als billige Arbeitskraft) auf den Praxisteil lenkt, wobei der interne Betreuer ja eher den ganzen Weg sehen möchte?

Genau, das kommt vor. Das sollte man dann versuchen während des Betreuungsprozesses zu lenken, indem man sich zum Beispiel regelmäßig Teilergebnisse vorlegen lässt, zumindest die Gliederung sehr gut bespricht und dann eben auch darauf hinweist, wenn die Grundlagenrecherche nicht so ausgeprägt ist, wie das der Fall sein sollte, damit man da auch möglichst noch gegensteuern kann. Es gibt aber auch durchaus auch Bachelorarbeiter, die sich drei Monate nicht melden und die einfach in ihrem Unternehmen vor sich hin werkeln und letzten Endes eine mehr oder weniger fertige Arbeit vorlegen. Wenn das dann zwei Wochen vor Abgabe ist, hat man auch wenige Möglichkeiten noch korrigierend einzugreifen. Dann bringt es auch nichts zu sagen, dass der Grundlagenteil nicht wissenschaftlich genug ist, da man das in zwei Wochen nicht vollends beheben kann. Und das liegt letzten Endes auch in der Verantwortung (des Studierenden). Es ist eine eigenständige Arbeit. Wenn ich schon wochenlang nichts gehört habe, kann ich mal nachfragen, wie der Stand der Dinge ist, aber letzten Endes ist es auch die Verantwortung des Einzelnen, sich zu melden und Unterstützung einzufordern. Es ist nicht meine Aufgabe den Leuten hinterherzulaufen und zu gucken, dass sie jeden Schritt richtig machen. Das kann dann manchmal zu einem unangenehmen Erwachen führen.

In einem solchen Fall, wo das Unternehmen ganz klar andere Interessen hat als der Betreuer, ist die Aufgabe dann auch, ganz klar bei dem Studierenden zu vermitteln oder zumindest eine Grundkommunikation aufzubauen?

Die Aufgabe des Studierenden ist, in erster Linie erst mal für Transparenz zu sorgen und sich Hilfe zu holen. Wenn ich den Eindruck habe, dass es in eine falsche Richtung geht, würde ich durchaus auch anbieten, das ich mich mit dem Unternehmen in Verbindung setze und dort dann auch das Gespräch suche mit den Betreuern vor Ort. Das kann ich aber nur dann tun, wenn ich gezielt darauf angesprochen werde. Meistens kenne ich die Leute ja nicht mal, die da in dem Unternehmen sind und das wäre auch von den Ressourcen her nicht zu leisten, ständig vor Ort irgendwelche Besuche zu machen. Das wird manchmal angefragt oder angeboten, aber das kann ich gar nicht leisten. Die Leute sind in allen möglichen Unternehmen. Dann wäre ich ja nur noch am rumreisen, um dann vor Ort irgendwelche Termine auszumachen. Das ist auch nicht meine Aufgabe. Aber wenn jemand zu mir kommt, mit dem Gefühl, in eine falsche Richtung gedrängt zu werden oder nur eine zu kleine Aufgabe zugeteilt bekommen zu haben. Das versucht man im Vorhinein zu vermeiden. Dass man

gerade bei größeren Projekten sicherstellt, dass derjenige auch wirklich eigenständig etwas bearbeiten kann.

Das kann auch zu Problemen führen - ist tatsächlich auch schon vorgekommen - das dann Dinge in der Bachelorarbeit präsentiert werden, die dann gar nicht von der Person selber erarbeitet wurden. Wir hatten mal einen Fall - da war ich Zweitprüfer - da ist das erst im Kolloquium aufgefallen, weil ich bestimmte Designentscheidungen hinterfragt habe und dann die Person gar nichts dazu sagen konnte. Ich habe mich dann gewundert, und habe die Person darauf angesprochen, dass sie es doch entschieden hat und dann doch begründen müsste, aber dann stellte sich heraus, dass sie es nicht entschieden hat. Sie hat das Design auch nicht erstellt. Das haben dann die Kollegen in der Grafikabteilung gemacht. In der Arbeit wurde es dann aber so präsentiert, als wäre es von der Person erstellt worden, was natürlich hochproblematisch ist. Dann muss man sich fragen, ob es überhaupt noch eine Arbeit ist, die als "bestanden" gewertet werden kann. Das sind eben so ein bisschen die Sachen, wo man vorher drauf gucken muss. Sind das wirklich Eigenanteile? Kann ich das auch ein Stück weit eigenständig bearbeiten? Habe ich da auch Entscheidungsbefugnisse? Wenn ich letzten Endes nur ein Befehlsempfänger bin, dann ist das nicht geeignet für eine Bachelorarbeit. Das sollte man versuchen im Vorfeld zu klären und meistens gelingt das dann auch.

Die Punkte, die Sie bisher genannt haben, zielen ja alle eher darauf ab, dass es bei der Durchführung der Bachelorarbeit Probleme gibt. Gerade was das Thema wissenschaftliches Arbeiten angeht gehen die Erwartungen von Studierenden und Professoren durchaus auseinander. Wie würden Sie denn die fachliche Kompetenz der Studierenden bewerten, also das, was besonders im Studium gelehrt wurde?

Gibt es dort auch Probleme?

Natürlich gibt es da Unterschiede. Es gibt Leute, die sind besser und es gibt Leute, die sind schlechter. Das ist letztlich ganz klar. Insofern haben wir da auch ein Spektrum bei Bachelorarbeiten, wie in jedem anderen Modul auch. Es gibt Leute, die das alles ausgezeichnet machen und es gibt Leute, bei denen hapert das. Aber es ist selten so, dass die Bearbeitung, die Methodik und die Vorgehensweise und die Planung exzellent sind, aber kein fachliches Wissen da ist, um die Aufgabe zu bearbeiten. Das habe ich eigentlich noch nie erlebt. Das geht so ein Stück weit Hand in Hand. Das würde im Grunde ja bedeuten, dass man sich für ein komplett falsches Thema entschieden hat, wo man zum Beispiel in Informatik dann anfängt, ein Design-Thema zu bearbeiten, obwohl man das im Studium eigentlich gar nicht gelernt hat. Oder umgekehrt, jemand der seinen Schwerpunkt im Design gelegt hat, eine Hardcore-Aufgabenstellung aus der Eingebetteten-Systeme-Welt bearbeiten würde und dann feststellt, dass er gar nicht die fachlichen Kompetenzen trägt, um das zu machen. Aber das weiß man ja eigentlich, oder kann es im Vorfeld einschätzen, ob es erfolgversprechend ist oder nicht. Typischerweise suchen sich die Leute ja auch Arbeiten, die ihren Schwerpunkten entsprechen und auch ihren Interessen. Das ist ja durchaus auch ein wichtiger Entscheidungspunkt, wenn ich mich für eine Arbeit entscheide. Dann ist das ja auch etwas, wo ich mich später mit bewerben kann, oder wo ich sagen kann: "Das habe ich erarbeitet" - als Arbeitsprobe. Und insofern geht das selten komplett auseinander, dass jemand komplett exzellent im Vorgehen ist und im Fachlichen schlecht. Umgekehrt kann es durchaus sein, dass man fachlich ordentlich was drauf hat, aber es "schlampig" angeht. Das kann man aber mit ein wenig Selbstdisziplin in den Griff bekommen.

**Betreuen Sie viele Arbeiten aus dem Studiengang Informatik/Softwareentwicklung?
(Anmerkung: Weil eigentlich Studiengangsleiter für Medieninformatik Online gewesen)**

Es ist häufig so ein "Drittmix" gewesen aus den Studiengängen, in denen ich überwiegend gelehrt habe, wie Medieninformatik Online - Bachelor und Master, Informationstechnologie und Design und Informatik, und daraus rekrutieren sich dann auch die Bachelorarbeiten, die ich angefragt bekomme. Mal ein bisschen mehr und mal ein bisschen weniger. Das schwankt immer ein bisschen aber es ist in der Regel eine ganz gute Mixtur.

Bei ITD gibt es doch ein recht frühes Modul, wo sich mit dem wissenschaftlichen Arbeiten beschäftigt wird, oder?

Bei ITD nicht, sondern in der Medieninformatik gibt es "Einführung in wissenschaftliche Projektarbeit". Das ist in der Tat - ganz früh nicht, sondern im 4. Semester – schon mal der Bachelorarbeit vorgeschaltet, wo man sich grundsätzlich schon mal mit solchen Dingen wie Recherche beschäftigt. Wie mache ich ein Literaturverzeichnis? Wie mache ich Literaturangaben? Was bedeutet eigentlich wissenschaftliches Vorgehen?

Achso. Hat das einen großen Einfluss auf die Bachelorarbeit? Merkt man da einen Unterschied? (Zwischen den Studiengängen)

Die Bachelorarbeiten sind insofern schwierig zu vergleichen, weil die Klientel einfach unterschiedlich ist. Wir haben in der Medieninformatik Online sehr viele Berufstätige, die berufsbegleitend studieren und insofern auch länger brauchen. Das ist nicht der Faulheit geschuldet, sondern weil sie geplant weniger Zeit für das Studium aufwenden und die Bachelorarbeiten entstehen dann häufig aus dem Arbeitskontext heraus. Weil es natürlich effektiv ist, wenn ich in einem Unternehmen tätig bin, wo man Projekte teilweise in einer relativ hohen Position betreut, dann ist das auch naheliegend, dass ich so was eben auch für eine Bachelorarbeit suche. Das ist dann auch eine ganz andere Position, als jemand, der ein Praktikum im Unternehmen macht oder eben als Bachelorstudent oder Werksstudent angestellt wird für eine Zeit. Die Projekte sind dann teilweise auch schon ein bisschen komplexer und anspruchsvoller, weil die Leute dann teilweise auch schon 10 Jahre Berufserfahrung haben und häufig mit diesen praxisorientierten Herangehensweisen sehr vertraut sind. Da ist meistens nicht das Problem, das so auszuarbeiten, wie man das in der Praxis dann eben macht. Da ist eher dann die Herausforderung, es wissenschaftlich zu betrachten und ich hoffe, dass dieses Modul dabei hilft, aber ich könnte es jetzt nicht 1:1 mit anderen Studiengängen vergleichen und sagen, dass die Bachelorarbeiten deshalb besser wären. Das ist auf jeden Fall etwas, was ich jedem Studiengang empfehlen würde: das frühzeitig mit einzubringen. Auch für Projektarbeiten kann es sehr nützlich sein.

Also es wäre Ihrerseits auf jeden Fall ein Ansatz, so ein Modul einzuführen, um damit die von Ihnen angesprochenen Probleme anzugehen oder zu lösen? Wie Sie schon gesagt haben, machen Studierende (im Laufe des Studiums) mehrere Projekte - größere Projekte, kleinere Projekte alleine oder auch als Gruppenarbeit. Haben Sie weitere Anregungen, wie man diesen Übergang aus dem "Studium" in die Bachelorarbeit "besser" gestalten könnte oder diese auftretende Kluft zu minimieren?

Ich versuche das teilweise in meinen Lehrveranstaltungen zu tun, indem ich nicht Klausuren als Prüfungsleistung nutze, sondern eben da schon semesterbegleitende Projektarbeiten, die dann oft schon solche Komponenten beinhalten. Sie haben ja auch Module von mir belegt. Es gibt dann ja auch Themen zu wählen, die eine wissenschaftliche Herangehensweise verlangen. Das mache ich nicht zur Pflicht, weil das ja nicht der primäre Inhalt des Moduls ist, aber viele tun das. Dann bekommen sie natürlich auch Feedback zur Herangehensweise und können das schon mal üben. Was bedeutet das eigentlich, eine sinnvolle Quellenrecherche zu machen? Wie kann ich beurteilen, ob die Quelle, die ich gefunden habe, seriös ist? Wie mache ich einen sinnvollen Quellenverweis? Das sind alles Dinge die man so schon mal übt. Alles in einem geschützten Rahmen, weil da noch nicht der Hauptfokus darauf liegt. Es gibt natürlich Feedback, aber es wird nicht erwartet, dass es da schon perfekt läuft. Das finde ich schon relativ wichtig, dass man da ab der Hälfte des Studiums, wenn man in das 4. oder 5. Semester kommt, schon mal etwas anspruchsvollere Aufgaben bearbeitet, und nicht dieses typische Auswendiglernen für eine Klausur und danach hat man alles einfach abgehakt.

(Kurze beispielhafte Beschreibung der im Rahmen der Bachelorarbeit zu entwickelnden Applikation)

Was ich von Studierenden in Interviews wahrgenommen habe ist, dass sie den Zeitplan zum einen völlig unterschätzen. Einer hat sogar schon damit gerechnet, dass es am Ende eng wird, schiebt seine Arbeit nach eigener Aussage trotzdem immer weiter auf.

Da soll die App zum einen ansetzen. Dass der Fokus auf Dinge gelenkt wird, die teilweise im Vorfeld unangenehm zu betrachten sind, aber im Nachhinein den Nachteil ausgleichen. Die Hoffnung ist, dass die Studierenden den Fokus haben, frühzeitig erkennen, wo die Schwierigkeiten liegen. Das "Zeitmanagement-Tool" ist dabei nur ein Aspekt. Darüber hinaus, sammle ich viele Tipps und Hinweise, die Studierende sonst viel zu spät selbst erfahren durch Fehler. Die ganze Idee ist natürlich, dass das begleitend funktioniert.

Wie das Beispiel "Schwimmen lernen". Wenn man vorher erklärt, wie es funktioniert (wie im Bachelor-Seminar) rennen am Ende alle ins Wasser rein und machen die üblichen Fehler trotzdem (wie bei der Bachelorarbeit).

Erkennen Sie bei einer solchen Applikation Chancen oder vielleicht sogar Risiken für die Bearbeitung der Bachelorarbeit?

Ich denke, die Unterstützung beim Zeitmanagement ist essenziell. Da habe ich ja vorhin ausgeführt, dass man Dinge gerne vor sich herschiebt. Unterstützung beim Zeitmanagement würde ja auch ein bisschen "Unterstützung beim Planungsprozess" implizieren. Ich kann ja nur dann ein gutes Zeitmanagement haben, wenn meine Planung gut ist. Wenn ich sage: "Ich brauche drei Wochen für die Literaturrecherche", und habe mich dabei aber grandios verschätzt, dann bringt mir die App aber auch nichts, wenn sie mir sagt, dass der Literaturteil bald dran ist, also wenn die ganze Zeitplanung letzten Endes unrealistisch war. Da könnte man sich überlegen - was ich dieses Semester ausprobiert habe - dass man eine sehr strukturierte Arbeitspaketbeschreibung macht. So, wie man es bei Projekten in der Wirtschaft oder in der Forschung auch machen muss. So, dass ich mir überlege, wie viele Stunden ich in etwas rein stecke, welche Tätigkeiten es konkret sind und man nicht nur so etwas schwammiges wie Literaturrecherche oder Anforderungsanalyse schreibt, sondern mir wirklich konkret überlege, welche Datenbanken ich zum Beispiel anschauen muss. Komme ich da überhaupt heran? Habe ich da Zugang? Wenn ja, habe ich den Zugang auch von Zuhause oder muss ich dafür beispielsweise in die Bibliothek, um mich mit einem Rechner vor Ort einzuloggen? Wie viele Rechercheergebnisse erwarte ich mir denn? Reicht es, wenn ich am Ende 5 einschlägige Publikationen habe oder erwarte ich eher fünfzehn? So was kann ich ja auch mit dem Betreuer besprechen. Wie ausführlich muss alles sein? Mache ich in der Anforderungsanalyse eher Interviews oder Fragebögen? Wenn ja, wie viele Personen muss ich eigentlich befragen? Komme ich an die Leute leicht heran oder ist das total schwierig? Muss ich damit rechnen, dass ich erst mal zehn Leute fragen muss, bevor einer mir zusagt? Das sind ja alles Dinge, die letzten Endes die Komplexität bestimmen und der Unterschied dann darin liegen kann, ob ich eine Woche brauche oder drei Wochen. Insofern denke ich, nicht nur Zeitmanagement, sondern auch eine Unterstützung bei der Planung wäre eine wichtige Sache, indem man zum Beispiel bestimmte Templates vorgibt, die mir bestimmte Informationen abverlangen. Dass ich mir über bestimmte Punkte eben Gedanken machen muss. habe ich schon die Fähigkeiten, die ich dafür brauche? Muss ich mit Einarbeitung rechnen? Wie ist der Umfang konkret zu benennen? Woraus speist sich das? Also das ich wirklich gezwungen bin eine konkrete Planung zu machen und dann wird auch meine Zeitplanung besser sein, weil ich das dann viel besser einschätzen kann. Das ist ja alles, was letzten Endes in der agilen Softwareentwicklung (zum Tragen kommt). Ich muss letztlich möglichst kleine Pakete schnüren, die ich dann gut abschätzen kann. Wenn ich ein Riesenpaket habe, schreibe ich dann vielleicht 10 Personenmonate dran, aber ob das nun realistisch ist, weiß ich nicht. Wenn ich aber sage, dass ich drei Klassen implementieren muss, dann habe ich ein viel handhabbareres Paket letzten Endes, dass

ich überschauen kann. Deshalb glaube ich, kann man relativ viel aus der agilen Softwareentwicklung entnehmen für so eine Bachelorarbeit, denn das ist eigentlich ein sehr gutes Beispiel dafür. Man hat einen begrenzten Zeitraum und ein im Vergleich zu anderen Softwareentwicklungsprojekten kleines Projekt. Dann kann man möglichst viele kleine Arbeitspakete verteilen, die ich dann gut planen und abarbeiten kann und wo man später gegebenenfalls auch gut gegensteuern kann, wenn man merkt, dass man falsch geplant hat. Dann habe ich die Möglichkeit im nächsten Arbeitspaket gegenzusteuern. Wenn Ihre App dabei unterstützen würde, wäre das, denke ich, sehr nützlich.

Erkennen Sie vielleicht auch Risiken, was die Benutzung der App angeht?

Man kann sich natürlich zum einen "tot planen". Das kann natürlich auch passieren, dass ich dann vor lauter Planung nicht zum Bearbeiten komme. Das sollte nicht passieren. Und weiterhin kann die App natürlich nicht die Eigenverantwortung abnehmen. Man kann am Ende nicht sagen, dass "die App mich nicht rechtzeitig erinnert hat und deswegen ist meine Bachelorarbeit nichts geworden". Das ist dann eher so eine Grundhaltung. Die Leute, die das für sich annehmen: „Das ist meine Arbeit, das ist mein Projekt, das ist auch so ein bisschen mein Baby und da stecke ich Herzblut rein.“, das ist, glaube ich, schon erforderlich, um so eine Arbeit über die Runden zu kriegen. Dass man davon überzeugt ist und dass man das will. Immer dann, wenn man gar nicht wirklich weiß, ob einen das Thema so richtig interessiert, ist es schon so ein Alarmsignal. Das ist dann keine gute Voraussetzung. Man muss eben schon ein bisschen für die Sache "brennen" und das kann keine App erzeugen. Das muss aus der Eigenmotivation heraus kommen.

Also sollte die App das natürlich auch klar kommunizieren, was die Aufgabe der App ist.

Genau

A.1.6. Notizen von Interviews mit Professoren (Interview VI)

1 Interview VI – Notizen

2 Frage 1 – Welches Fachgebiet für Bachelorarbeiten?

3 Zu 80% haben die Arbeiten einen Bezug zu den genannten Fachgebieten Eingebettete Systeme,
4 Rechnerarchitekturen, Betriebssysteme und Systemprogrammierung. 20% fallen in andere Bereiche,
5 wie zum Beispiel Benutzeroberfläche oder Simulationen.

6 Frage 2 – Wie sehen diese Bachelorarbeiten aus? / Wo liegen die Prioritäten? / Welche 7 Typen von Arbeiten gibt es?

8 Der erste Typ ist die Entwickelnde Arbeit. Jedoch werden, im Gegensatz zu anderen Betreuern und
9 deren Fachgebieten, mehrheitlich hardware-nahe Themen behandelt.

10 Hierbei liegt der Fokus auf der Testbarkeit (Stichwort Sensoren, Messungen und Messwerte) und
11 dem Aufbereiten und Analysieren sowie der Validierung im Rahmen der Implementierung.

12 Ein weiterer Typ sind konzipierende Arbeiten. Hierbei werden zunächst die notwendigen
13 Anforderungen erhoben, welche Dinge modelliert werden sollen, dann folgt die Konzeptstudie oder
14 auch ein Machbarkeitsbeweis. Es folgt die Evaluation (Stichwort Brauchbarkeit/Technologien).

15 Zu 90% werden interne Arbeiten geschrieben. Externe Arbeiten können durchaus einen
16 Mehraufwand bedeuten.

17 Frage 3 – Erwartungen an den Bacheloranden?

18 Wichtig ist selbstständiges Arbeiten. Dazu gehört, die Arbeit richtig einzugrenzen, dass das Problem
19 richtig identifiziert wird und ein Selbstständiger Lösungsprozess stattfindet sowie die dazugehörige
20 Literaturarbeit.

21 Die Kommunikation sollte vom Studierenden angeregt werden. Der Betreuer bietet seine
22 Unterstützung an, der Studierende muss sie aber auch selbstständig annehmen. Dazu zählt, sich an
23 Einigungen zu halten, sich eigenständig zu melden, aktiv Hilfestellung zu erbitten und Feedback
24 einzuholen. Es ist nicht die Aufgabe des Betreuers, den Studierenden zu erinnern.

25 Aufgabenstellungen werden teils durch Abstracts bereitgestellt, die in gemeinsamer Arbeit detailliert
26 ausformuliert werden. Der Studierende kann gerne eigene Anregungen einbringen oder auch ein
27 eigenes Thema bearbeiten.

28 Frage 4 – Wo treten Probleme auf? / In welchen Punkten mangelt es?

29 Es gibt Probleme dabei, das Problem in Teilprobleme zu zerlegen.

30 Vielen mangelt es an richtigem Zeitmanagement. Es wird realitätsfern geplant, so dass es oftmals z.B.
31 einen tollen Entwurf und eine gute Implementierung gibt, aber nur einen mäßigen Nachweis.

32 Einige Studierende haben Probleme damit, sich eigenständig zu melden, da sie Angst haben, die
33 Erwartungen des Betreuers nicht zu erfüllen.

34 Die Literaturarbeit, also das korrekte wissenschaftliche Arbeiten, sorgt für Probleme. Dazu zählen das
35 korrekte Einbinden von Quellen, Zitaten und Belegen oder auch ein angebrachter Schreibstil und die
36 Rechtschreibung.

37 Die selbstständige Vorbereitung, das korrekte Einschätzen von Arbeitsaufwand, das konstruktive
38 Umgehen mit Kritik sind weitere problematische Punkte. Der Betreuer möchte klar den Unterschied
39 zum letzten Arbeitsstad sehen. Eine Gliederung der nächsten Arbeitsschritte und ein
40 wochenbasierter Meilensteinplan helfen, klare Ergebnisse zu produzieren.

41 Frage 5 – Ansätze um diese Probleme zu lösen?

42 Das Grundlagenkapitel ist eine gute Möglichkeit für den Betreuer, die Arbeitsweise und die
43 Kompetenzen des Studierenden einzuschätzen und sollte daher direkt am Anfang geschrieben
44 werden, auch da es fachlich noch oberflächlich ist. Der Betreuer kann es als Abgleich nutzen, um sich
45 selbst und die Betreuung auf den Studierenden einzustellen.

46 Das Bachelorarbeit-Seminar vermittelt zwar richtige Inhalte, diese kommen jedoch bei den
47 Studierenden nicht an, sie behalten die Inhalte nicht.

48 Frage 6 – Meinung zu Applikation? Einstellungen? Anregungen?

49 Die Applikation könnte die Akzeptanz (die Aufnahmebereitschaft) der Studierenden gegenüber den
50 enthaltenen Informationen bezüglich der Bachelorarbeit steigern.

51 Risiken könnten sein, dass das selbstständige Denken zu sehr von der Applikation abgenommen wird.
52 So wird die eigentliche Arbeit der Prüfungsleistung nicht vom Studierenden selbst ausgeführt. Der
53 Lerneffekt könnte gemindert werden, da partielles Scheitern oder das Erfahren von Grenzen (also
54 das Fehlermachen) eine wichtige Erfahrung für den Studierenden sind.

A.1.7. E-Mail-Befragung der Professoren I

Schriftliche E-Mail Befragung I

1
2

3 **Können Sie Arten/Klassen von Bachelorarbeiten identifizieren?**

4 **(Als Anregung hierfür, gebe ich gerne die typische "entwickelnde" Abschlussarbeit als**
5 **Paradebeispiel, bei der der Softwareentwicklungszyklus im Mittelpunkt steht)**

6 Synthese-Arbeiten vs. Analyse-Arbeiten (z. B. Softwareentwicklung vs. Fallstudie)

7 Interne vs. externe Arbeiten (letztere zusätzlich mit externer Betreuungsperson)

8

9 **Was erwarten Sie von den Bacheloranden? Können sie abgrenzen, was sie von den**
10 **Bacheloranden fordern und demgegenüber, was Sie als Ihre Aufgabe als Betreuer**
11 **bezeichnen?**

12 Erwartungen: Selbständige Einarbeitung in die Problemdomäne, selbständige Auswahl adäquater
13 Methoden und Werkzeuge, strukturiertes Vorgehen, schlüssige und formal-saubere Dokumentation
14 inkl. Erläuterung von Entwurfsentscheidungen

15 Aufgabe als Betreuer: Ansprechperson für fachliche, organisatorische und
16 dokumentationsspezifische Fragen (z. B. Einhaltung von Standards wissenschaftlichen Arbeitens),
17 kontinuierliche Begleitung des Arbeitsfortschritts, finale Begutachtung

18

19 **Können Sie Probleme identifizieren, die seitens der Bacheloranden, immer wieder erneut**
20 **auftreten? Gibt es grundsätzliche Probleme oder Schwierigkeiten bei dem Vorgang/Ablauf**
21 **der Bearbeitung der Bachelorarbeit?**

22

23 Übliche Probleme: mangelhaftes Zeitmanagement, mangelhafte Schwerpunktsetzung (z. B.
24 Verzetteln in unwichtigen Details), mangelhafte Strukturierung der Dokumentation, unangemessener
25 Schreibstil, unzureichende Kritikfähigkeit, formale Fehler (Orthographie, Kommasetzung, Grammatik,
26 Zitierweise, Quellenangaben)

27 **Haben Sie Ideen oder Ansätze, wie man die Studierenden bei dem Prozess der Bearbeitung**
28 **der Bachelorarbeit unterstützen kann?**

29 Nutzung von Werkzeugen zum Projektmanagement, kontinuierliche Begleitung durch regelmäßige
30 Statusgespräche, Bereitstellung von Vorlagen und Musterdokumenten

31

32 **Sehen Sie Risiken oder Chancen bei Anwendung einer solchen Applikation, die die**
33 **Studierenden bei der Bearbeitung der Bachelorarbeit unterstützt?**

34 Chancen: Die App könnte Probleme im Projektmanagement und bei der Gestaltung der
35 Dokumentation (vgl. 3) – je nach Usability und echtem Mehrwert.

36 Risiken: Die App könnte missverstanden werden als Ersatz für die persönliche Betreuung –
37 insbesondere fachliche Aspekte wird eine App naturgemäß nicht abdecken können. Es könnte
38 weiterhin zu einem „Device Mismatch“ kommen, da Bachelorarbeiten üblicherweise nicht an
39 mobilen Endgeräten entstehen – man müsste also immer zwei Geräte bedienen: Notebook/Desktop-
40 PC und Smartphone. Ggf. hätte es Vorteile, die Funktionen der App auf demjenigen Endgerät zur
41 Verfügung zu haben, das primär für die Erstellung der Bachelorarbeit genutzt wird.

A.1.8. Zusammenfassung der schriftlichen Befragung der Studierenden

Ergebnisse der schriftlichen Befragung von Studenten Teil 1

Beschreibung:

Die wörtlichen Aussagen der befragten Studierenden wurden im folgenden Dokument verschiedenen ermittelten Kategorien zugeordnet. Die gewählten Kategorien (**Erwartungen, Einschätzung der größten Probleme und Wünsche für bessere Vorbereitung durch das Studium**) gehen der Fragestellung hervor, während die jeweiligen Unterkategorien durch Sortieren und Filtern von Antworten entstanden sind.

Insgesamt ausgewertete Gespräche mit Studierenden: 17

Erwartungen

Teilen von Erfahrungen

- Ich erwarte, dass er mich bei der Arbeit mit der Firma unterstützt, indem er mir zum Beispiel Dinge sagt, die ich nicht wissen kann. Wie z. B. dass das zeitlich nicht passen wird, oder ein Teil der Arbeit nicht in eine BA gehört. Ich erwarte, dass er seine Erfahrungen mit mir teilt.
- Dass er mich in meiner Bachelorarbeit unterstützt, auch was die Kommunikation mit der Firma betrifft.
- Dass er bei der Themenfindung mögliche Richtungen aufzeigen kann, die für ein geeignetes Thema eingeschlagen werden können.
- Erarbeitung eines sinnvollen Themas
- Es würde mich freuen, wenn er mich über den zeitlichen Ablauf einer BA informiert, also wie viel Zeit man hat, was wann ansteht
- Zudem soll er den Umfang der Arbeit richtig einschätzen können.

Kommunikation

- Erreichbarkeit, dass Fragen schnell beantwortet werden oder man sich treffen kann, wenn was ist und Kritik.
- Die Möglichkeit, überhaupt mal kommunizieren zu können, Treffen bei Erreichen eines neuen Meilensteins.
- Ich erwarte, dass der Betreuer die nötige Zeit aufbringt, um mich betreuen zu können und auch ansprechbar und aufgeschlossen gegenüber Fragen ist.
- Ich erwarte, dass ich mit meinem Betreuer jederzeit über das Projekt reden kann, wenn ich Fragen habe

Feedback

- Und dass er sich mit meinen Sachen auseinandersetzt und nicht nur hofft, dass ich das schon gut mache.
- Bei Fragen Hilfestellung geben.
- Von meinem Betreuer erwarte ich konstruktives Feedback in Bezug auf Ideen, die ich ihm mitteile, außerdem konstruktive Kritik an meiner Dokumentation.
- Grundsätzliches Interesse an der BA des Betreuers, konstruktives Feedback und eine gewisse Diskussionsbereitschaft.
- An den vereinbarten Feedbackpunkten Rückmeldung zu meiner Arbeit geben, mich dabei davor bewahren, überdurchschnittlich dumme Dinge zu tun
- regelmäßiges Feedback über aktuellen Arbeitsstand und das er sagt, wenn was scheiße ist

- Einfach nur Feedback. Quasi eine Person, die Korrektur liest. Also jemand, der auf Unverständlichkeiten hinweist bzw. bei groben schlechten Entscheidungen dazwischenfunkt
- Dass der Betreuer sich genügend Zeit für Fragen/Antworten nimmt und einen nicht ignoriert.
- Dass er sich Zeit nimmt für den Studenten und den Studenten dabei unterstützt eine wirklich professionelle wissenschaftliche Arbeit zu schreiben
- Sinnvolles Feedback während der Entstehung der Arbeit und gegebenenfalls Denkanstöße bei auftretenden Problemen
- Dass er für Fragen offensteht
- Mir würde sehr viel Tipps zu Struktur der Bachelorarbeit helfen.

Eingreifen des Betreuers

- Wenn er erkennt, dass ich was falsch mache, das er mich in die richtige Richtung lenkt
- Weiterhin kann manchmal ein kleiner Schubser, was das Zeitmanagement betrifft nicht schaden denke ich.
- Anregungen für Erweiterungen. Andere Denkweisen anstoßen. Aber auch Einforderungen für Leistungen.
- Dass er von sich aus sagt welche Teile meiner BA er sehen will, also wann er meine ersten Implementierung-Versuche sehen will oder die Gliederung oder ob da erst was Sinnvolles drinstehen soll.
- Dass er die Zielstellung klar definiert

Einschätzung der größten Probleme

Schreiben der Dokumentation

- Unsicher war ich mir beim Gewichten der Ausführungen. Also wie viel muss ich zu einem Abschnitt schreiben und was davon alles mit Quellen belegen.
- Ausführlichkeit der Doku
- Die Doku was drinstehen soll. Also die Gliederung. Seitenanzahl. Rechtschreibung.
- Die Einschätzung vom Umfang der Arbeit
- Eine sinnvolle Doku parallel zum Code zu erstellen und eine gute Gliederung für diese zu finden
- Die größten Problemen in meinem Fall würden sehr wahrscheinlich bei der Dokumentation der Bachelorarbeit liegen. Vor allem - man weiß nie, ob man sich in der richtigen Richtung bewegt (zumindest kann man in dem Fall nicht hundert Prozent sicher sein). Das zweite Problem ist, dass Bachelorarbeit schon eine Art wissenschaftlicher Arbeit ist. Umgangssprache passt in dem Fall nicht.

Zeitmanagement

- Definitiv das Zeitmanagement. Ich muss dringend auspassen das ich das schreiben nicht vergesse und nur Code.
- Das größte Problem wird bei mir darin liegen, einen Zeitplan zu erstellen
- Zeitmanagement
- Zeitmanagement
- ZEIT
- Zeitmanagement wird sehr wahrscheinlich ein Problem sein
- Einschätzung des Umfangs

Fachliche Probleme

- Konkrete Anforderungen
- Weiterhin werde ich vor einem Problem stehen, welches wie folgt aussieht:

Die von mir programmierten HMI-Module sind mit einem Bus mit dem Host verbunden. Sollten jetzt 2 Module ausfallen und ausgetauscht werden, soll die Software des Hosts in der Lage sein, neue Module zu erkennen und mit denselben Aufgaben zu bestücken, wie ihre Vorgänger.

Beispielsweise sind die beiden Module für "Anker rauf/runter" und "Notausstieg auf/zu" defekt, werden ausgetauscht und die Funktionalitäten sollen dann bei den nächsten Modulen an derselben Stelle liegen.

- Hardware zum Laufen kriegen
- sinnvolles Testen
- Unsicherheiten bei den implementation, wenn es nicht so wird wie ich es erhofft hätte

Sonstige Probleme

- Die Kommunikation zwischen Betreuer, Firmenmitarbeiter und mir zu koordinieren. soweit ich weiß, wechselt mein Betreuer in der Firma oder es müsste mit mehreren Personen und dem Betreuer kommuniziert werden. Hier hat mein Betreuer schon gesagt, dass der gern EINE Ansprechperson hätte und nicht mit der ganzen Firma diskutieren müsste.
- Schwer zu sagen, gefühlt alles
- Größtes Problem sehe ich immer noch in der Themenfindung.
- Meine größten Probleme liegen bei der Themenfindung für eine Bachelor Arbeit zu finden

Wünsche für bessere Vorbereitung durch das Studium

Zeitmanagement

- Bessere Vorbereitung im Bereich Planung und Zeitmanagement. Wie lange brauchen verschiedene Sachen wirklich. Welche Probleme treten auf, mehr mit größeren Projekten arbeiten. Das Softwaretechnik Projekt ist dafür sehr gut, nur leider kommt es ganz zum Schluss und auch nur ein Modul. Webtech war zwar ähnlich, aber da hat man wenig mit Management gemacht. Da haben wir meistens "einfach drauf los programmiert".
- Mehr Projektarbeiten evtl. auch bestehende Projekte übernehmen (z.B. SWT-Projekt AutoSail), evtl. Aufgaben in denen man mal so Zeitschätzungen machen soll, beispielsweise, wie lange dauert das Modellieren einer Autonomen Steuerung für das AutoSailProjekt mit anschließender Auflösung, um zu gucken, das man eine völlig falsche Zeitvorstellung hat. Den Studenten auf den Boden der Tatsachen zurückholt und ihm klar machen das nicht alles mal eben so runterprogrammiert wird in 2 Stunden.
- Zeitaufwände besser abschätzen können
- Zeitmanagement ob man noch in der Zeit liegt. Das mit dem Feedback von anderen bachelorstudenten würde Die Motivation sicherlich erhöhen jedoch nur wenn diese auch selber Fortschritte vorweisen.
-

Wissenschaftliches Arbeiten und Schreiben

- Ein Kurs "Wissenschaftliches Schreiben", am besten speziell für Informatiker. Darin können die Unsicherheiten beseitigt werden. Viele haben ja auch Probleme damit, wie Quellcode in den Text untergebracht werden kann. Gibt zwar häufig Tipps dazu (auch im Lernraum) aber trotzdem sind mit einem kleinen Absatz über den Umgang mit Quellcode längst nicht alle Fragen geklärt. Außerdem müssten die Betreuer dann nicht jedes Mal aufs neue Hinweise dazu geben. Dieser Kurs sollte aber vor dem 6. Semester angeboten werden (als NT-WPF?).
- Richtiger Aufbau einer wissenschaftlichen Arbeit, Sinnvolle Quellen angeben und Zitate etc. einbringen
- Mehr Wert auf die Dokumentation in allen Projekten legen. Diese Dokumentation muss aber in allen Modulen einheitlich und gleich sein (zumindest demselben Muster folgen). Vielleicht sollen die Studierenden sich mehr mit Papers beschäftigen (z.B. Studenten erzwingen, so was wie Paper von Google zu Big Table zu lesen, damit man sich an "wissenschaftlichen Styl" gewöhnt.

Informationen zur Bachelorarbeit

- Ich hätte mir gewünscht besser auf die BA vorbereitet zu werden. Sowas wie im 4ten Semester mal eine Infoveranstaltung, was man machen könnte, worauf man achten muss bei dem planen und das man jetzt schon mal anfangen sollte zu gucken. Vielleicht sogar vor und Nachteile und/oder Erfahrungen von BA in einer Firma oder im Ausland. Sowas fehlt komplett!
- Dass man mehr auf die Formalitäten einer Bachelor Arbeit vorbereitet wird (richtiges zitieren, Schriftform etc.)
- Überschaubarkeit des Projektes. Welcher Aufwand ist wirklich notwendig.
- Vielleicht mehr Informationen über das organisatorische (vielleicht hab ich die auch nur noch nicht, weil ich mich da noch nicht mit befasst hab), ansonsten fällt mir da leider nicht viel ein
- Im 5 Semester Einführungsveranstaltungen und Beratung, falls kein Thema gefunden wird

- Eine Infoveranstaltung für die Bachelorarbeit. Also früher, irgendwann im 3/4 Semester. Ein paar grobe Regularien. Also wie das mit Prüfern ist, was ich alles fertig haben muss, ob ich früher abgeben kann, wie funktioniert das Kolloquium...

-

Sonstige Anmerkungen

- Eigentlich nicht wirklich viel mehr, da ich mich mit der Einstellung angemeldet habe, dass es eher allgemein wird und dass ich viele Teilbereiche kennen lerne.

Ich finde dieses Studium ist mehr drauf aufgebaut einen breiten Umfang des Themen Bereichs

Informatik zu geben und die BA nur Mittel zum Zweck ist, den Zettel zu bekommen von daher finde ich es so in Ordnung wie es ist. Halt nur wenn man nicht drauf vorbereitet wird, ist es wichtig das der Mentor möglichst vieles erklärt

- Ich finde, dass alles wichtige gezeigt/beigebracht wurde damit man für die Bachelorarbeit vorbereitet ist.
- Bisher bin ich zufrieden mit dem Inhalt
- Viel mehr verschiedene Design Pattern während des Studiums Kennenlernen und auch durchgehend in den Praktika einsetzen und viel mehr testen von Applikationen während des Studiums
- Da würde mir jetzt nichts konkretes einfallen
- nichts

Ergebnisse der schriftlichen Befragung von Studenten Teil 2

Beschreibung:

Befragung der Studierenden unter Angabe, ob die Studierenden positiv oder negativ auf die Ideen Vorschläge reagiert haben

Motivation durch soziale Einbindung anderer Studierende

Die Antworten lassen sich in drei Kategorien unterteilen, welche sich aus den positiven, negativen oder neutralen Antworten zusammensetzen.

Art der Reaktion	Häufigkeit der Reaktion
Positiv	3
Negativ	12
Neutral	2

Motivation durch das Sammeln von Achievements

Die Antworten lassen sich in drei Kategorien unterteilen, welche sich aus den positiven, negativen oder neutralen Antworten zusammensetzen.

Art der Reaktion	Häufigkeit der Reaktion
Positiv	15
Negativ	1
Neutral	1

Ergebnisse der schriftlichen Befragung von Studenten Teil 3

6. Würdest du die App nutzen um eine der folgenden Zwecke zu erfüllen?

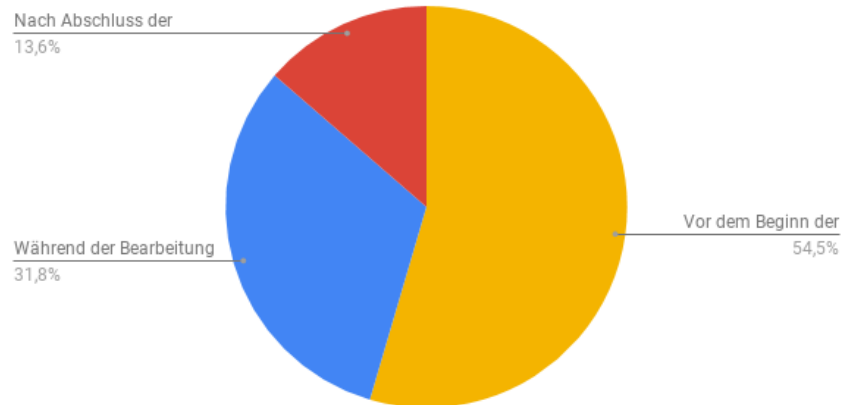
- Strukturierte Informationssammlung zum wissenschaftlichen Arbeiten, Gliederung und weiteren Informationen zu Kernpunkten der Bachelorarbeit?
- Spielerische Einbindung der Bachelorarbeit durch sammeln von Achievements?
- Planung und Übersicht von Meilensteinen, Aufgaben

Art der Antwort	Häufigkeit der Antwort
Positiv	14
Negativ	0
Neutral	3

A.1.9. Ergebnisse der Online-Umfrage für die Studierenden

Allgemeine Fragen

In welcher Phase des Studiums/der Bachelorarbeit befindest Du dich? Bitte wähle die am ehesten zutreffende Antwort.

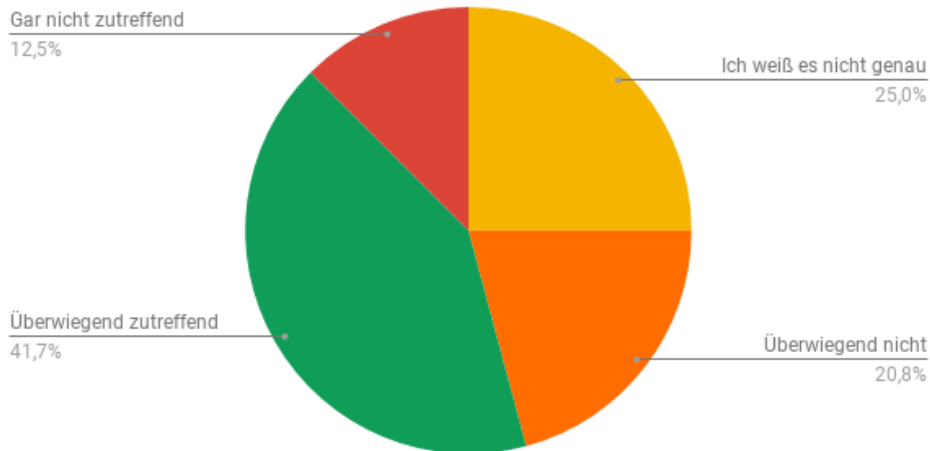


Ca. 55% der Befragten befinden sich zum Zeitpunkt der Umfrage vor dem Beginn der Bachelorarbeit (d.h. auch in den vorherigen Semestern), ca. 33% befinden sich bei der Bearbeitung und etwa 13% haben bereits eine Bachelorarbeit geschrieben.

Die nachfolgenden Fragen wurden dem jeweiligen Status der Befragten angepasst.

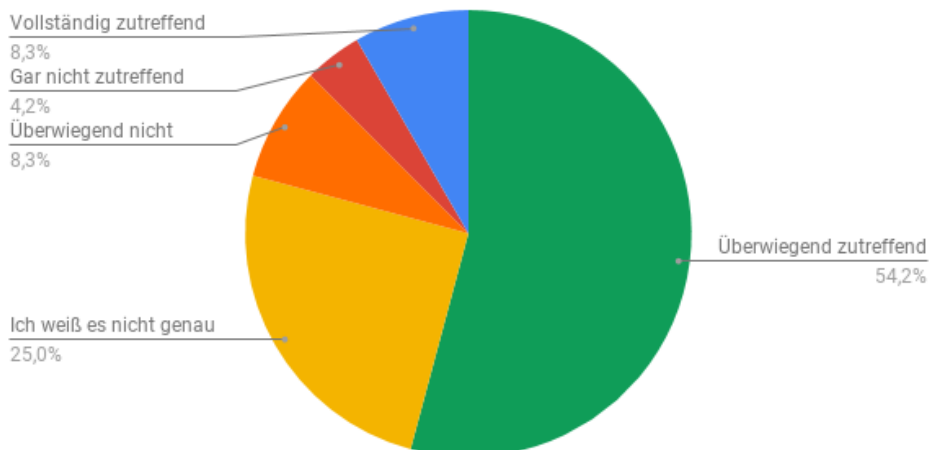
Vor dem Beginn der Bachelorarbeit

Ich fühle mich durch das bisherige Studium sehr gut auf das Bearbeiten der Bachelorarbeit vorbereitet



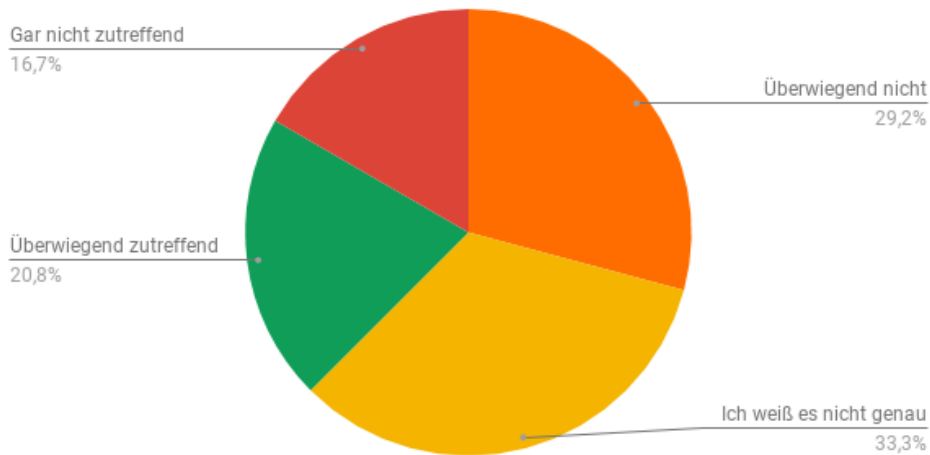
Etwa 40% der Studenten fühlen sich durch das Studium überwiegend auf das Bearbeiten der Bachelorarbeit vorbereitet. Der Rest fühlt sich unsicher oder überwiegend nicht bis gar nicht durch das Studium auf die Bachelorarbeit vorbereitet.

Das selbständige Vorgehen wird kein Problem für mich darstellen



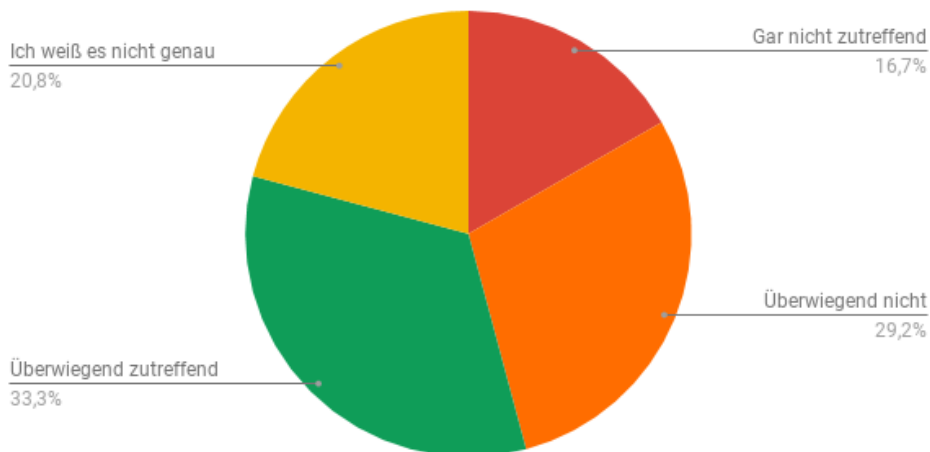
Knapp über die Hälfte der Studenten sehen überwiegend keine Probleme beim selbstständigen Vorgehen, 8% sehen sich auch vollständig dazu in der Lage. Ein Viertel ist sich unsicher und der Rest fühlt sich überwiegend nicht bis gar nicht dazu fähig, selbstständig vorzugehen.

Ich habe eine genaue Vorstellung davon, wie die Bearbeitung einer Abschlussarbeit abläuft



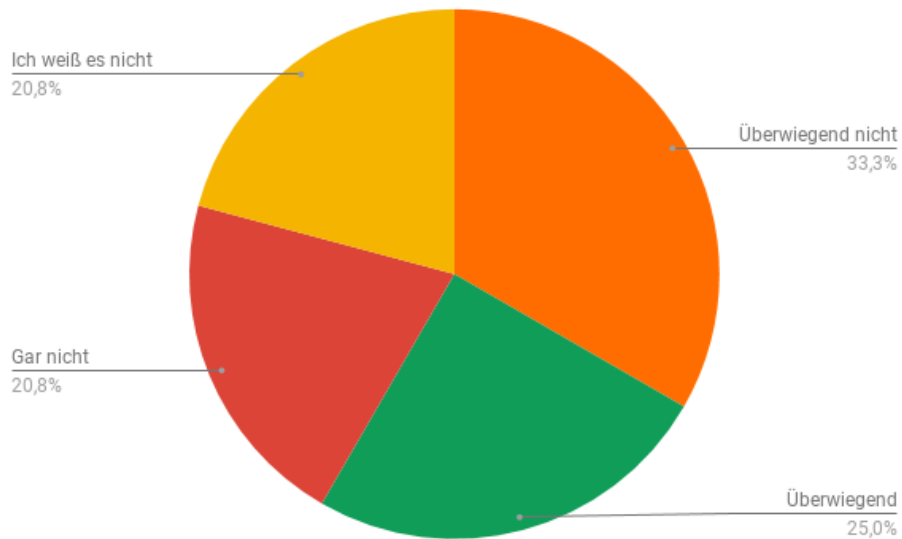
Knapp die Hälfte der Studenten haben überwiegend keine oder gar keine genaue Vorstellung davon, wie die Bearbeitung einer Abschlussarbeit abläuft. Ein Drittel ist sich unsicher und die restlichen 20% haben überwiegend eine Vorstellung davon.

Das Einschätzen von Zeitaufwand (Meilensteine, Teilziele) fällt mir leicht



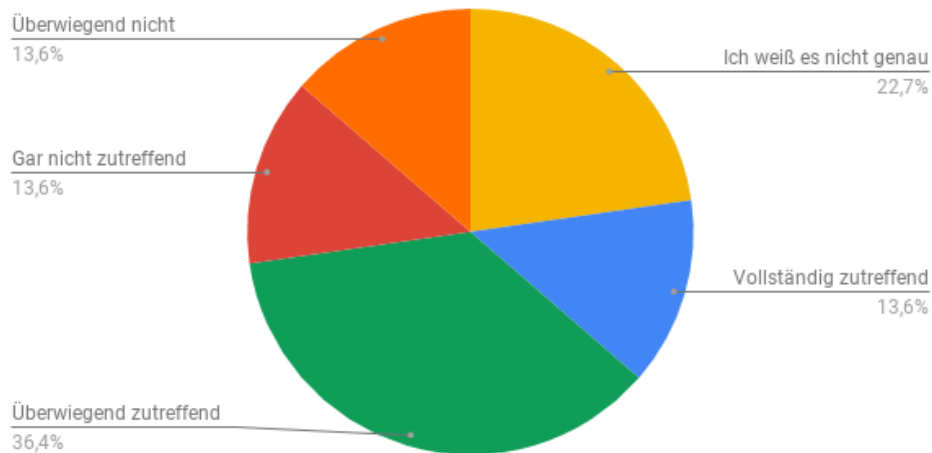
Ein Drittel der Studenten fühlt sich überwiegend sicher mit dem Einschätzen von Zeitaufwand, 20% sind sich unsicher und die restlichen ca. 45% fühlen sich überwiegend nicht oder gar nicht sicher dabei.

Ich weiß ganz genau, was bei der Bearbeitung meiner Abschlussarbeit von mir verlangt wird und bei welchen Punkten die Prioritäten liegen



Ein Viertel der Befragten weiß überwiegend, was von ihnen verlangt wird. 20% sind sich unsicher, ein Drittel ist sich überwiegend unsicher und 20% wissen gar nicht was von ihnen erwartet wird.

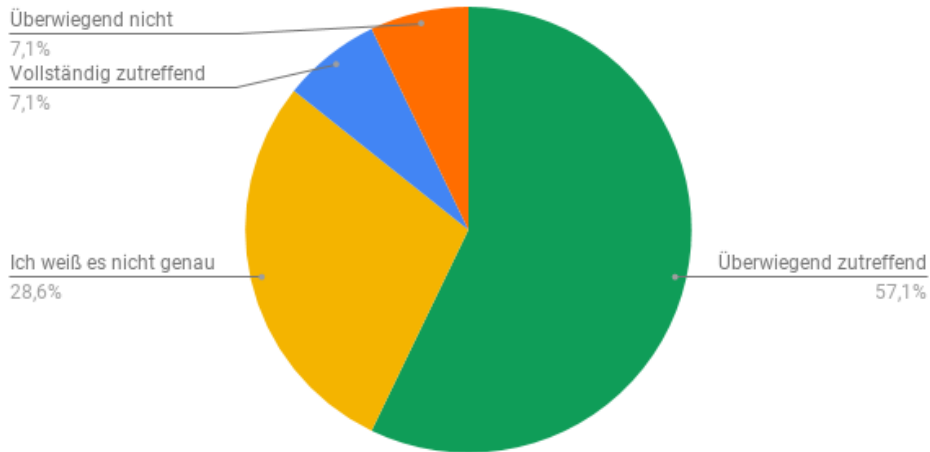
Ich weiß, wie man korrekt wissenschaftlich arbeitet und vorgeht (Literaturarbeit, Quellenangaben)



Die Hälfte der Studenten weiß mindestens überwiegend, wie man korrekt wissenschaftlich arbeitet, 13% davon sind sich vollständig sicher. Fast ein Viertel weiß es nicht genau und der Rest weiß es überwiegend nicht oder gar nicht.

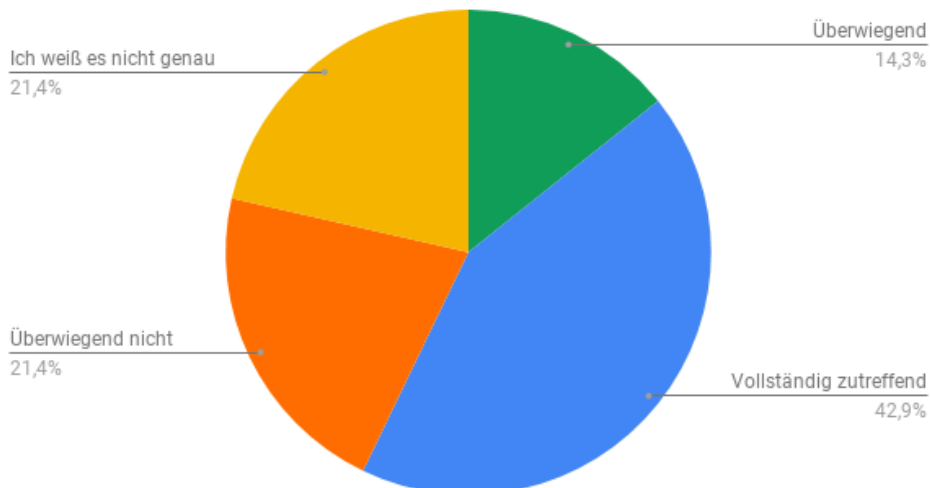
Während der Bearbeitung der Bachelorarbeit

Ich fühle mich durch das Studium sehr gut auf das Bearbeiten der Bachelorarbeit vorbereitet



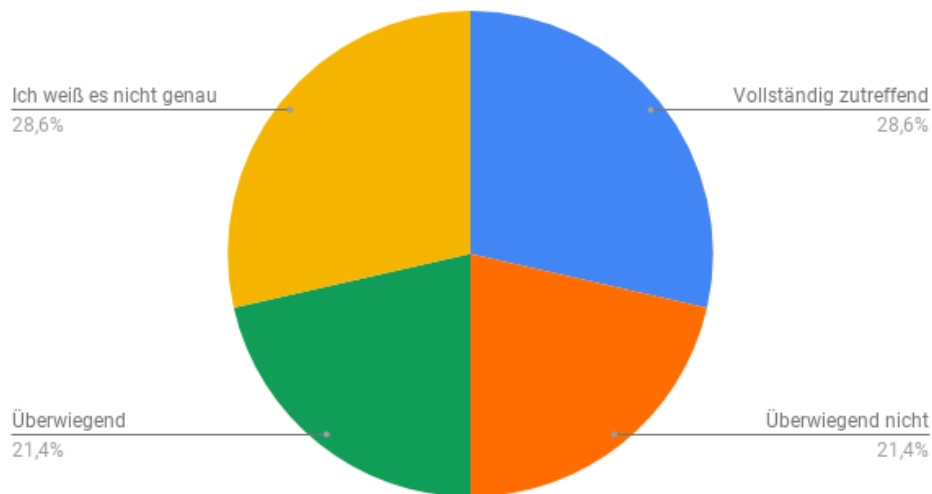
Knapp 65% der Studenten sehen sich mindestens überwiegend auf das Bearbeiten einer Bachelorarbeit vorbereitet, 7% davon auch vollständig. Der Rest ist sich mindestens unsicher, 7% davon fühlen sich überwiegend unvorbereitet.

Es fiel mir leicht, das Thema meiner Bachelorarbeit zu finden



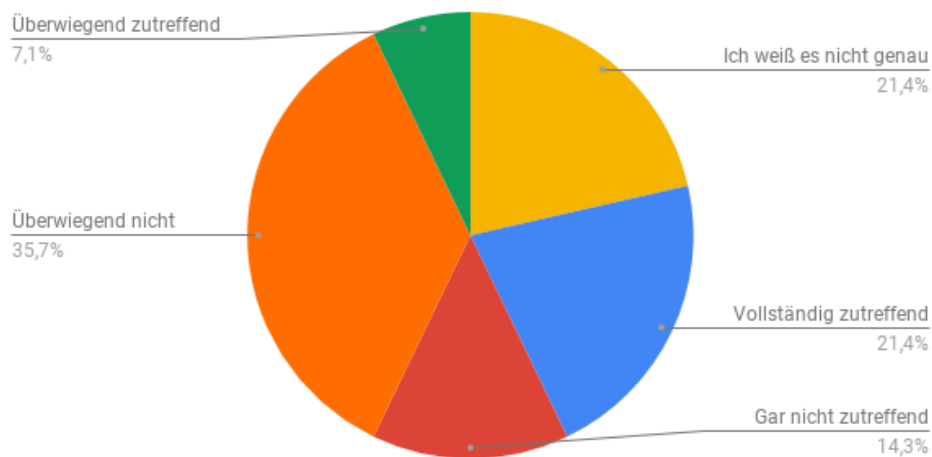
Einem großen Teil fiel es vollständig leicht, ein Thema für die Arbeit zu finden, 14% fielen es überwiegend leicht. Dagegen war sich der Rest unsicher oder überwiegend nicht in der Lage, leicht ein Thema zu finden.

Das selbständige Vorgehen stellt kein Problem für mich dar



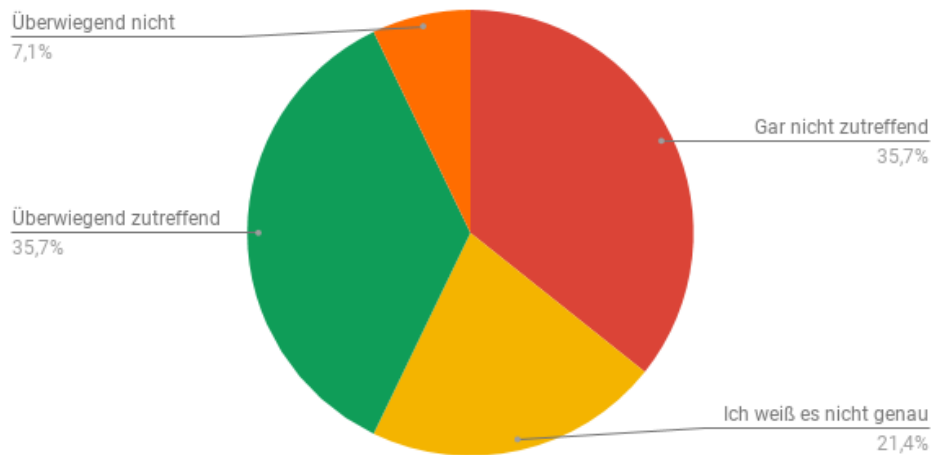
Für die Hälfte der Studenten stellt selbständiges Arbeiten mindestens überwiegend kein Problem dar, 29% sehen sich vollständig dazu in der Lage. Die andere Hälfte ist sich mindestens unsicher, 21% sehen sich überwiegend nicht zu selbstständigem Arbeiten in der Lage.

Das Einschätzen von Zeitaufwand (Meilensteine, Teilziele) fällt mir leicht



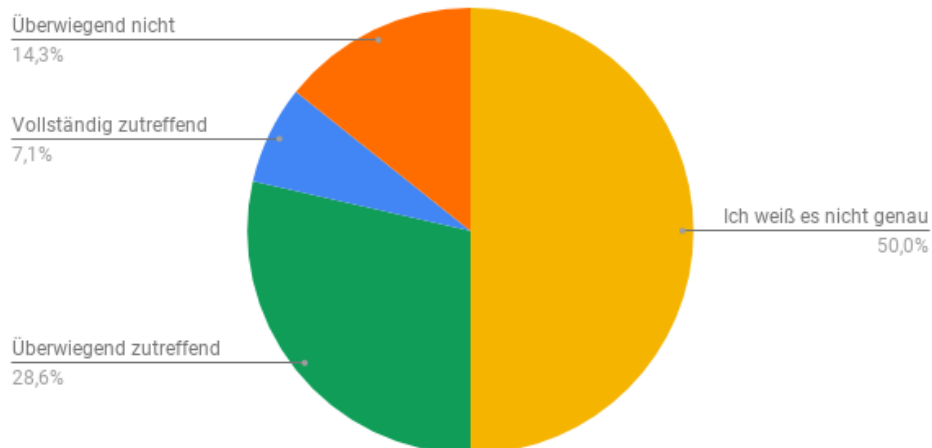
50% der Befragten fällt es mindestens überwiegend schwer, 14% auch komplett schwer, Zeitaufwand einzuschätzen. Etwa 20% fühlen sich unsicher, vom Rest fühlen sich etwa 7% überwiegend sicher dabei, ca. 21% auch vollständig sicher.

Ich fühle mich durch das Bearbeiten der Bachelorarbeit überfordert



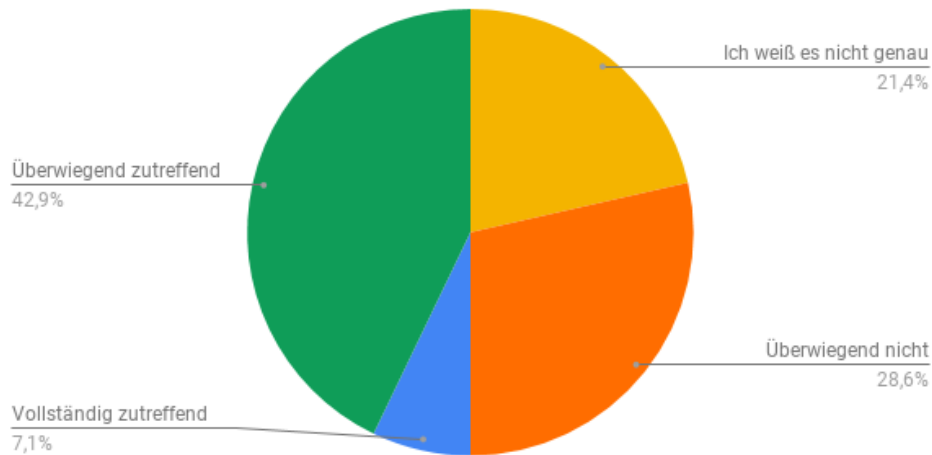
Etwa 35% der Befragten fühlen sich überwiegend durch das Bearbeiten der Bachelorarbeit überfordert, genauso viele fühlen sich gar nicht überfordert. Ca. 21% sind sich unsicher und der Rest fühlt sich überwiegend nicht überfordert.

Ich weiß ganz genau, was bei der Bearbeitung meiner Abschlussarbeit von mir verlangt wird und bei welchen



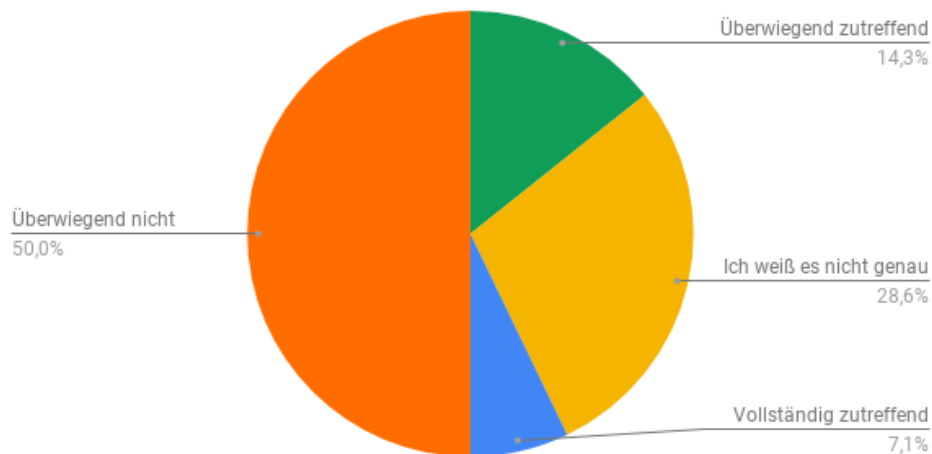
Die Hälfte der Studierenden ist sich unsicher, was von ihnen verlangt wird. Ca. 14% wissen es überwiegend nicht. Der Rest weiß es mindestens überwiegend, etwa 7% weiß es vollständig.

Ich kenne mich gut mit den Formalien zur Bachelorarbeit aus
(Anmeldung, Ablauf, Zulassungsanforderungen)



Die Hälfte der Befragten kennt sich mindestens nicht genau aus, ca. 28% davon kennt sich überwiegend nicht mit den Formalien zur Bachelorarbeit aus. Die andere Hälfte kennt sich mindestens überwiegend aus, 7,1% auf vollständig.

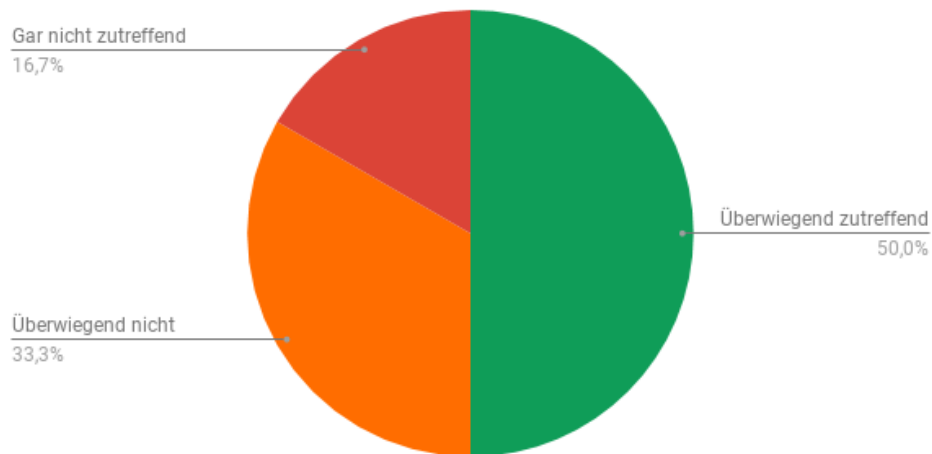
Ich weiß, wie man korrekt wissenschaftlich arbeitet und
vorgeht (Literaturarbeit, Quellenangaben)



Die Hälfte der Studierenden kennt sich überwiegend nicht mit wissenschaftlichem Arbeiten aus. Etwa 29% sind sich dabei unsicher, der Rest weiß mindestens überwiegend, 7,1% auch vollständig, wie man wissenschaftlich arbeitet.

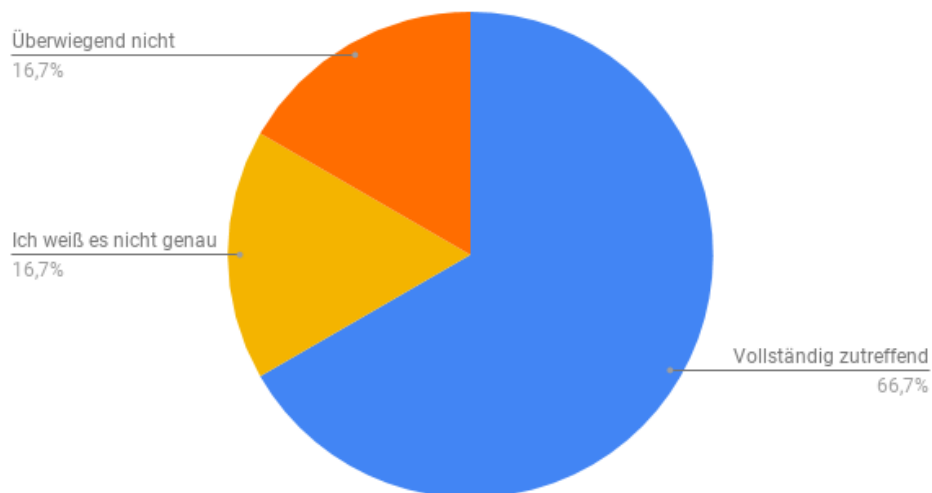
Nach Abschluss der Bachelorarbeit

Das Studium hat mich auf das Bearbeiten der Bachelorarbeit sehr gut vorbereitet



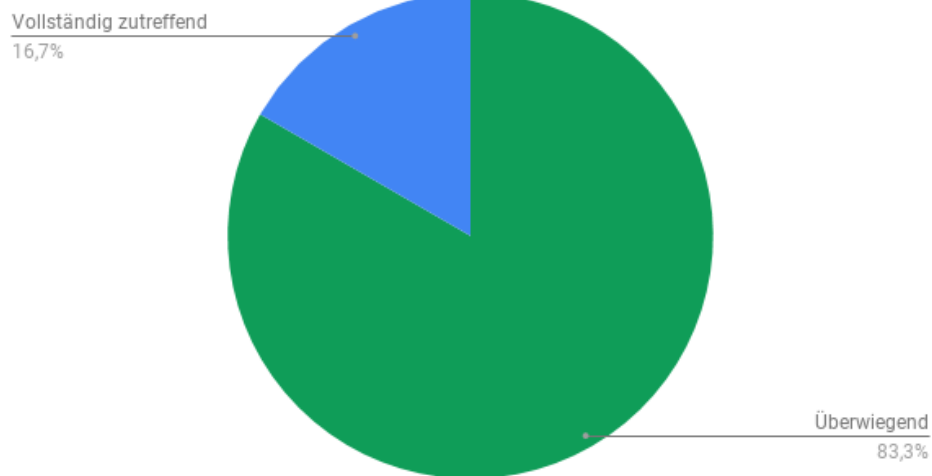
50% der Absolventen fühlten sich durch das Studium überwiegend auf die Bachelorarbeit vorbereitet. Ein Drittel fühlte sich überwiegend nicht, und der Rest fühlte sich gar nicht vorbereitet.

Es fiel mir leicht, das Thema meiner Bachelorarbeit zu finden



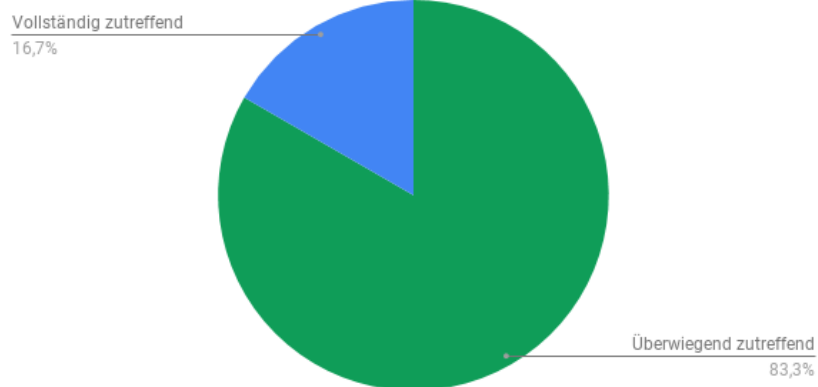
Zwei Drittel der Befragten fiel es leicht, ein Thema für die Bachelorarbeit zu finden. Der Rest war sich unsicher, oder ihm fiel es überwiegend nicht leicht.

Das selbständige Vorgehen stellte kein Problem für mich dar



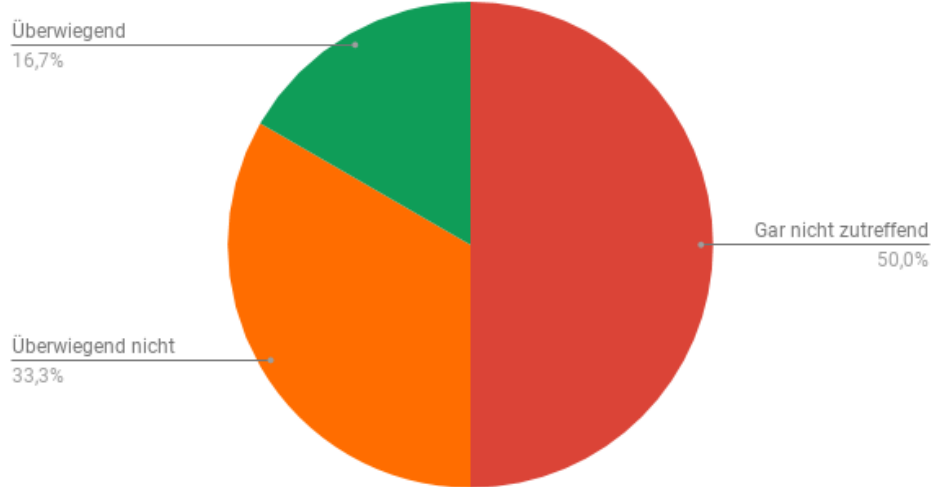
Alle Befragten fühlten sich mindestens überwiegend sicher beim selbstständigen Vorgehen, ca. 17% fühlten sich auch vollständig sicher.

Das Einschätzen von Zeitaufwand (Meilensteine, Teilziele) fiel mir leicht



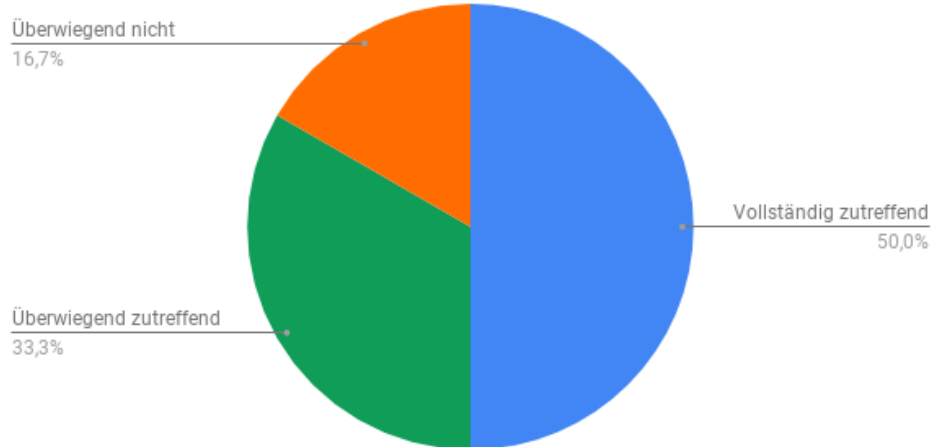
Alle Befragten fühlten sich mindestens überwiegend sicher beim Einschätzen von Zeitaufwand, ca. 17% fühlten sich auch vollständig sicher.

Das Fertigstellen der Bachelorarbeit hat mich überfordert



Die Hälfte der Absolventen hat sich gar nicht mit der Bachelorarbeit überfordert gefühlt. Ein Drittel fühlte sich überwiegend nicht überfordert und der Rest fühlte sich überwiegend überfordert.

Die Benotung meiner Abschlussarbeit entsprach genau meinen Erwartungen



Die Benotung der Arbeit entspricht bei 50% der Absolventen genau der Erwartung, bei einem Drittel überwiegend und beim Rest überwiegend nicht.

A.2. Übersicht der Herausforderungen und Achievements

Herausforderungen & Titel der Challenge-Achievements

Anfangszeitraum

1. **Challenge:** Beantrage deine Zulassung zur Bachelorarbeit
Achievement: „Dein erster Schritt ist getan!“
2. **Challenge:** Finde das Thema deiner Bachelorarbeit
Achievement: „Dieses Thema will ich haben! Kein anderes!“
3. **Challenge:** Suche dir deinen Betreuer/deine Firma für die Bachelorarbeit
Achievement: „In guten Händen...“
4. **Challenge:** Bereite deinen Arbeitsplatz für die Arbeit vor
Achievement: „Genug Platz für Snacks, Arbeit und Snacks!“
5. **Challenge:** Schreibe deine Aufgabenstellung
Achievement: „Gratulation! Die Probleme sind klar. Jetzt kommst du!“
6. **Challenge:** Erstelle die Gliederung für deine Arbeit
Achievement: „Alles in Reih’ und Glied“
7. **Challenge:** Erstelle deinen Zeitplan
Achievement: „Pünktlich wie die Deutsche Bahn“
8. **Challenge:** Arbeite dich in deine Schreibumgebung ein (Beispiel „LaTeX“)
Achievement: „Exzellente Vorbereitung!“
9. **Challenge:** Erarbeite deine Recherchegrundlagen
Achievement: „So viel Text! ... und so wenig Inhalt?“
10. **Challenge:** Erstelle die Ordnerstruktur für die Organisation deines Projektes
Achievement: „Alles da, wo es hingehört!“
11. **Challenge:** Kümmere dich um eine geeignete Versionsverwaltung (Beispiel „GitHub“)
Achievement: „Sicher ist sicher!“
12. **Challenge:** Suche dir eine geeignete Vorlage für deine gewählte Schreibumgebung
Achievement: „Das Rad nicht neu erfinden...“
13. **Challenge:** Lasse deine Prüfung Anmelden
Achievement: „Schnall dich an! Jetzt geht es los!“
14. **Challenge:** Kümmere dich bereits frühzeitig um mögliche Korrekturleser
Achievement: „Das ist wohl eher keine Einschlaflektüre...“
15. **Challenge:** Besorge dir Nervennahrung für die Bearbeitungszeit
Achievement: „Mühsam ernährt sich der Student“

Bearbeitungszeitraum

1. **Challenge:** Beschreibe die für den Leser relevanten Rechercheinformationen in deinem Grundlagenkapitel
Achievement: „Das weiß doch jedes Kind...“
2. **Challenge:** Stelle den getätigten Arbeitsinhalt in deinem Hauptteil dar
Achievement: „Der schwerste Teil ist geschafft!“
3. **Challenge:** Fasse deine Ergebnisse in dem Abschlusskapitel zusammen
Achievement: „Alle Fäden laufen zusammen“
4. **Challenge:** Überprüfe alle Kapitel deiner Bachelorarbeit auf Konsistenz
Achievement: „Passt, wackelt und hat Luft!“
5. **Challenge:** Suche dir deinen Zweitprüfer
Achievement: „Vier Augen sehen besser als zwei“

Abschlusszeitraum

1. **Challenge:** Lasse deine Arbeit Korrekturlesen
Achievement: „Mein Freund, die Rechtschreibung“
2. **Challenge:** Finalisiere deine Bachelorarbeit durch eine abschließende Überarbeitung
Achievement: „Letzter Feinschliff“
3. **Challenge:** Lasse deine Arbeit in einem Druckgeschäft geeignet binden (ca. 1-2 Tage Dauer)
Achievement: „Unter Druck zum Diamanten“
4. **Challenge:** Führe die Abgabe deiner Bachelorarbeit durch
Achievement: „FunFact: Einsteins Doktorarbeit umfasste 17 Seiten“
5. **Challenge:** Bereite deine Präsentation für das Kolloquium vor
Achievement: „Fast geschafft!“
6. **Challenge:** Führe deine Präsentation im Kolloquium
Achievement: „Feier schön! Du hast es dir verdient“

A.3. Oberflächenprototyp

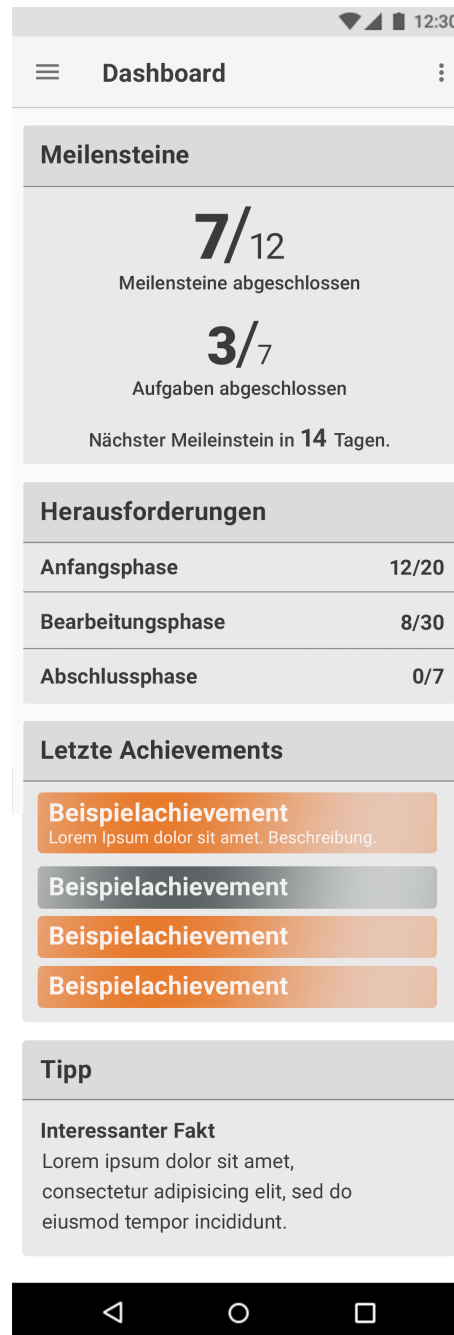
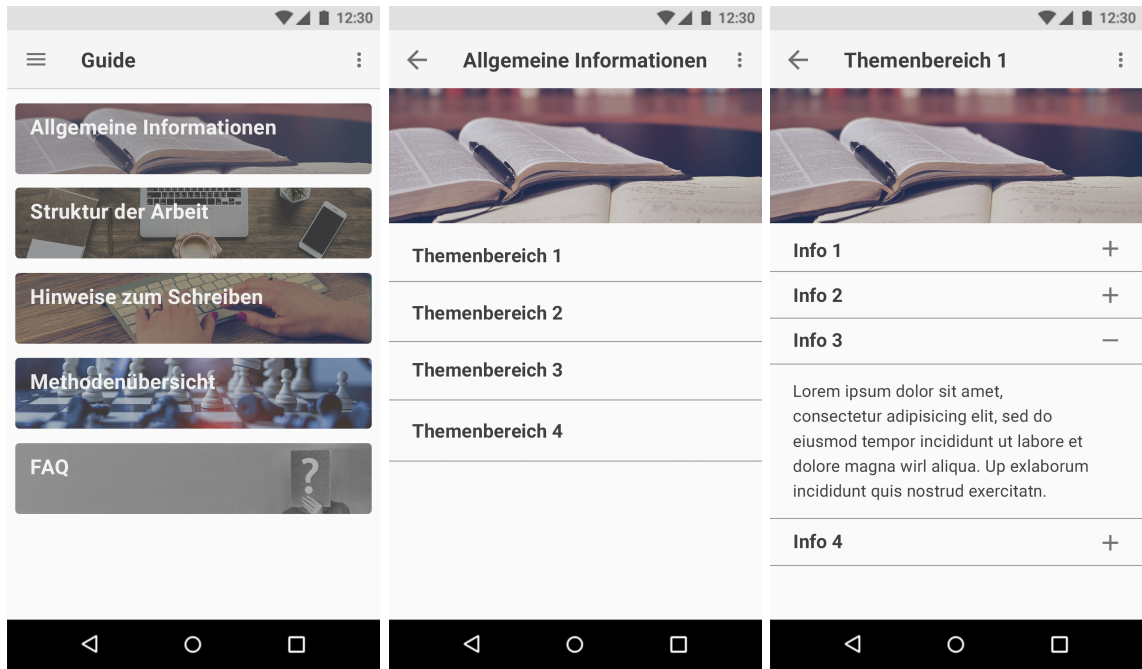


Abbildung A.1.: Oberflächenprototyp: Dashboard

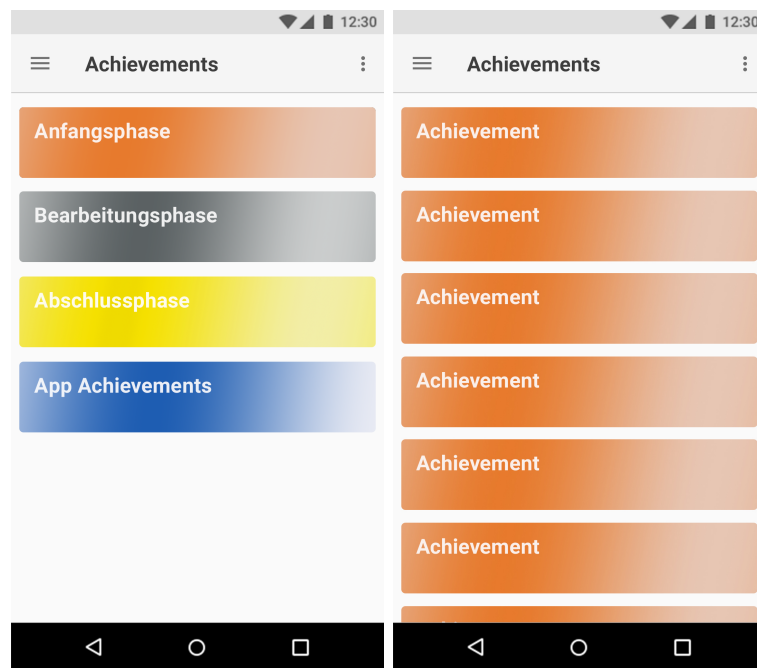


(a) Guide

(b) Guide - Themenebene

(c) Guide - Informationsebene

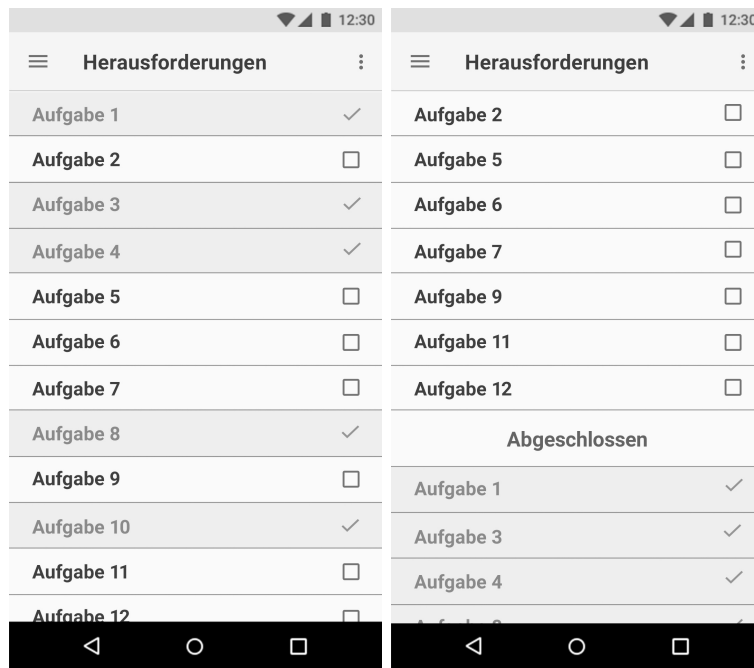
Abbildung A.2.: Oberflächenprototyp: Guide



(a) Achievements - Übersicht

(b) Achievements - Detailebene

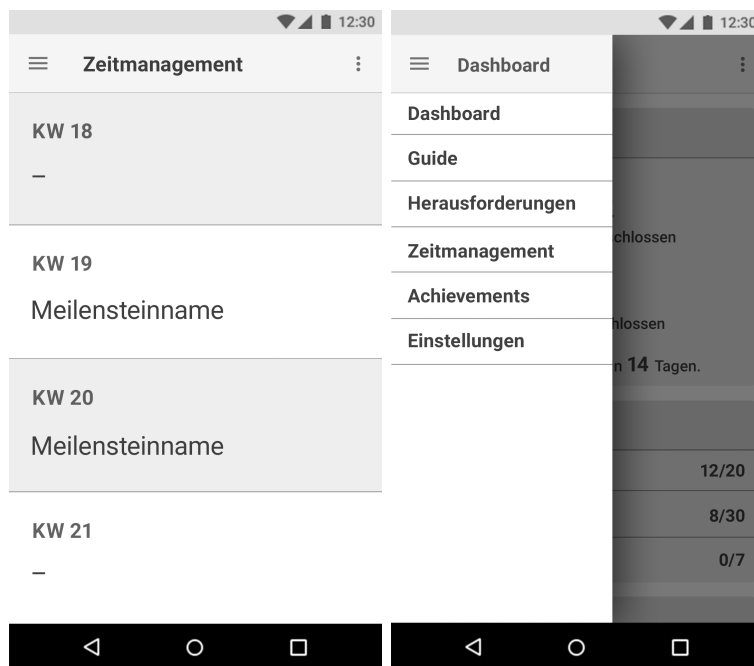
Abbildung A.3.: Oberflächenprototyp: Achievements



(a) Aufgaben - Übersicht

(b) Aufgaben - Sortiert

Abbildung A.4.: Oberflächenprototyp: Aufgaben



(a) Zeitmanagement - Übersicht

(b) Menu - Navigation

Abbildung A.5.: Oberflächenprototyp: Sonstiges

A.4. Detaillierte Beschreibung der Model Komponenten

Die nun folgenden Unterkapitel beschreiben die Klassen, die Teil der Model Komponente sind und teilweise wichtige Datenfelder und Operationen zur Verwaltung des Redux-Stores enthalten.

Hinweis: Alle verwendeten Klassen deren Objekt-Instanzen in dem Redux-Store liegen, sind mit dem Schlüsselwort `@immutable` versehen und können zur Laufzeit nicht verändert werden. Dies ist hinsichtlich der Redux-Architektur vorgesehen, da der Redux-Store und dessen Inhalt nicht verändert, sondern überschrieben wird. Weiterhin liegen die Attribute innerhalb der Objekt-Klassen im Rahmen des flach ausgerichteten Key-Value-Stores des AppStates im Allgemeinen nicht als Objektreferenzen, sondern als IDs in textueller Form vor.

A.4.1. AppState-Klasse (`@immutable`)

Die Klasse AppState stellt den aktuellen Status der Applikation dar und trägt somit eine Vielzahl an Informationen in sich. Diese Informationen werden im Rahmen der an Redux orientierten Architektur durch das Einsetzen von Key-Value Stores strukturiert und verwaltet.

Jede für den Zustand der Applikation relevante Information ist auf diese Weise an einer zentralen Stelle im System gespeichert (siehe Listing A.1). Durch einen lesenden Zugriff können die Oberflächenkomponenten alle benötigten aktuellen Informationen ermitteln und abbilden. Weiterhin kann der Status der Applikation durch das Ausführen von Aktionen des Benutzers, was das Auslösen der sogenannten Actions mit sich trägt, verändert werden.

```
1 class AppState{
2
3     final Map<String, Milestone> currentMilestones;
4     final Map<String, Map<String, Achievement>> achievedAchievements;
5     final Map<String, Property> properties;
6     final Map<String, Challenge> challenges;
7     final Map<String, Content> informationToolContent;
8
9     final Milestone selectedMilestone;
10    final DateTime begin;
11    final DateTime end;
12
13    ...
14 }
```

Listing A.1: Datenfelder der AppState-Klasse

Aktueller Zustand der Meilensteine

Die aktuellen Meilensteine werden durch das Datenfeld `currentMilestones` abgebildet. Diese Datenstruktur enthält alle vom Benutzer angelegten Meilensteine und deren jeweilige Unteraufgaben und wird als `SplayTreeMap` initialisiert. Die selbst-sortierende Eigenschaft dieser Datenstruktur bietet einige Vorteile im Rahmen der Handhabung von Meilensteinen in der Implementierung, da deren einzigartige IDs auf dem angegebenen Meilensteindatum basieren.

Aktueller Zustand der Achievements

Die von dem Benutzer der Applikation zu erreichenden Achievements werden im Datenfeld `achievements` gespeichert. Dieses Datenfeld weist einige Besonderheiten vor, auf die ich im Folgenden hinweisen möchte:

Hinweis: Es handelt sich bei dieser Datenstruktur im Detail um eine `HashMap`, welche weitere `HashMaps` beinhaltet.

Diese `HashMap` wird mit genau vier verschiedenen konstanten Einträgen initialisiert.

- **ALL_ACHIEVEMENTS**

Enthält alle vorhandenen Achievements.

- **ACHIEVED**

Enthält alle Achievements, die von dem Benutzer gesammelt wurden.

- **RECOGNIZED**

Enthält alle gesammelten Achievements, die von dem Benutzer betrachtet wurden.

- **NOT_RECOGNIZED**

Enthält alle gesammelten Achievements, die von dem Benutzer noch nicht betrachtet wurden.

Diese Unterscheidung ist durch die Idee begründet, die unterschiedlichen Zustände der Achievements bei der Handhabung der Benutzerinteraktion zu unterschiedlicher Zeit zu berücksichtigen. Beispielsweise sollen neu erworbene Achievements, welche vom Benutzer noch nicht eingesehen worden sind, bis zu dem Moment, in dem sie betrachtet wurden, gesondert behandelt werden.

Aktueller Zustand der Properties

Die zu verwaltenden Eigenschaften zur Erfüllung der Achievements werden in dem Datenfeld `properties` gespeichert. Dies bietet den Vorteil, dass die Properties, welche von unterschiedlichen Achievements geteilt werden können, an einer zentralen Stelle im State verwaltet werden können.

Aktueller Zustand der Herausforderungen

Die für den Benutzer zu erledigenden Herausforderungen werden in dem Datenfeld `challenges` verwaltet. Über die bereits genannten Verwaltungsstrukturen hinaus beinhaltet die Klasse weiterhin folgende einzelne Informationen:

Aktueller Zustand des Bachelorarbeit-Guide Inhaltes

Der aus einem Git Repository ausgelesene und durch Content-Objekte strukturierte Inhalt des Bachelorarbeit-Guides wird in dem Datenfeld `informationToolContent` verwaltet.

Weitere Datenfelder

Der aktuell, im Rahmen der Bedienung der Applikation von dem Nutzer ausgewählte, Meilenstein wird im Datenfeld `selectedMilestone` gespeichert.

Die beiden Felder `begin` und `end`, bezeichnen den durch den Benutzer angegebenen Bearbeitungszeitraum der Bachelorarbeit.

A.4.2. Milestone-Klasse (@immutable)

Ein Meilenstein besitzt folgende Attribute und Methoden:

String id

Jeder Meilenstein besitzt eine ID, die immer auf dem jeweils festgelegten Datum des Meilensteins basiert.

String title

Der eigentliche Titel beschreibt den, von dem Benutzer gegebenen, Namen des Meilensteins und dient als Identifikations- und Anzeigemittel für den Benutzer.

String description

In dem Datenfeld description befindet sich die von dem Benutzer eingetragene Beschreibung des Meilensteins.

DateTime date

Jeder Meilenstein muss ein Ablaufdatum enthalten, welches den Tag darstellt, an dem der Meilenstein eingetragen ist.

Map<String, Task> tasks

Meilensteine besitzen, von dem Benutzer hinzugefügte, Unteraufgaben, die in dem Datenfeld tasks liegen.

Map<String, Task> getCompletedTasks(), getNotCompletedTasks()

Durch die Methoden getCompletedTasks() und getNotCompletedTasks() ist die Möglichkeit gegeben, abgeschlossene und noch offene Unteraufgaben der Meilensteine herauszufiltern.

MilestoneState getMilestoneState()

Durch die Methode getMilestoneState() bietet sich die Möglichkeit, den aktuellen Status des Meilensteins zu bestimmen, welcher auf abgeschlossene Unteraufgaben, noch offene Unteraufgaben sowie der Relation zwischen dem aktuellen Zeitpunkt und dem definierten Ablaufzeitpunkt basiert. Für diesen Zweck wird weiterhin die Enumeration MilestoneState verwendet, die die vier Zustände Empty, AllTasksCompleted, SomeTasksNotCompleted und outOfDateAndSomeTasksNotCompleted darstellt.

A.4.3. Task-Klasse (@immutable)

Ein Task besitzt folgende Attribute:

String id

Jeder Task besitzt eine ID als Identifikationsmöglichkeit.

String title

Der Titel eines Tasks stellt die, von dem Nutzer erstellte, inhaltlich zu lösende Aufgabe dar.

TaskState taskState

Ein Task kann die Zustände **Completed** und **NotCompleted** annehmen, welche durch eine Enumeration dargestellt werden. Eine Enumeration wurde einem booleschen Wert vorgezogen, da weitere Zustände, wie zum Beispiel das Favorisieren von Unteraufgaben, bei Bedarf hinzugefügt werden könnten.

A.4.4. Challenge-Klasse (@immutable)

Die von dem Benutzer zu absolvierenden Herausforderungen enthalten folgende Attribute:

String title

Jede Herausforderung besitzt einen Titel, welche den Inhalt der zu absolvierenden Aufgabe darstellt. Dies ist also der Inhalt, welcher der Benutzer in der Applikation angezeigt bekommt.

ChallengeType type

Der Typ einer Challenge lässt sich durch eine definierte Enumeration ausdrücken. Vorgesehen ist dies, um den jeweiligen Herausforderungen einen Kontext zu verleihen. Als Beispiel kann hier eine Aufteilung in verschiedene Phasen der Bachelorarbeit erfolgen.

bool completed

Ob eine Herausforderung abgeschlossen wurde oder noch offen ist, wird über das Feld **completed** angegeben.

A.4.5. Achievement-Klasse (@immutable)

Ein Achievement besitzt folgende Attribute:

String id

Stellt einen einmaligen Identifier der Achievements dar.

String title

Der Titel des Achievements ist der Teil, der dem Nutzer als Identifikationsmittel des jeweiligen Achievements angezeigt wird.

bool completed

Die Information, ob ein Achievement erreicht wurde, wird durch den Wert des Attributs completed dargestellt.

DateTime completionDate

Stellt den Zeitpunkt des Erreichens des Achievements dar.

List<String> properties

Weiterhin beinhaltet ein Achievement bestimmte Eigenschaften, die erfüllt werden müssen, damit das Achievement als erreicht bezeichnet werden kann. Ein Achievement wird also nur erreicht, wenn alle seine Properties als erfüllt gelten.

AchievementType type

Der Typ, welcher durch eine Enumeration verdeutlicht wird, stellt die Art des Achievements dar.

Hinweis: Bei der nicht ganz einfachen Implementierung von Achievements wurde sich nach ausgiebiger Recherche an einem im Internet geteilten Beitrag (vgl. [19]) orientiert und die dort beschriebene Umsetzung für die eigene Redux-basierte Variante adaptiert.

A.4.6. Property-Klasse (@immutable)

Die Klasse Property enthält folgende Attribute und Funktionen:

String name

Der Name einer Property-Instanz stellt die Bezeichnung dar, welche dem Nutzer als Identifikationsmöglichkeit dient und im Rahmen der Applikation als Instruktion angezeigt werden kann, weshalb diese Bezeichnung wohlüberlegt gewählt werden sollte.

int initialValue

Der Initialwert beschreibt, wie der Name bereits impliziert, den Startwert, der bei Initialisieren des Objektes angenommen wird.

int currentValue

Ein aktueller Wert stellt den aktuellen Fortschritt der jeweiligen Eigenschaft, welche es zu erfüllen gilt, dar.

int activationValue

Das Datenfeld activationValue stellt den Zielwert dar, den es zu erfüllen gilt.

String activationRule

Die genannten Felder initialValue, currentValue und activationValue werden durch eine definierte Regel in Beziehung zueinander gesetzt. Diese Regel kann verschieden ausgeprägt sein, um dem Ersteller von Achievements sowie deren Kriterien zur Erfüllung eine gewisse Freiheit zu ermöglichen. Im Rahmen des Projektes wurden drei verschiedene Regeln in Form von konstanten Variablen des Datentyps String aufgestellt, welche die Regeln "<", ">" und "==" enthalten.

bool isActive()

Diese Regeln werden als Werkzeug benutzt, um die Semantik der zu erfüllenden Eigenschaften auf die logische Ebene zu übertragen. Beispielsweise kann somit in der Methode `isActive()` überprüft werden, ob der Benutzer bereits drei Meilensteine erstellt hat, also die jeweilige Eigenschaft erfüllt wurde.

A.4.7. Content-Klasse (@immutable)

Eine Objektinstanz der Content-Klasse stellt den Inhalt der “Guide“-Funktion dar und enthält folgende Attribute:

String id

Die ID eines Content-Objekts dient als Zuweisungsmittel von weiteren Content-Objekten als Unterabschnitten. Die verwendeten IDs werden in der ausgelesenen JSON-Datei, welche in dem Git-Repository liegt, zugewiesen

String title, description

Der Titel des Content-Objekts ist gleichzusetzen mit einer Überschrift für den jeweiligen Inhalt, der durch die Beschreibung ausgedrückt wird.

List<String> subsections

Das Datenfeld subsections enthält Referenzen durch in textueller Form zugewiesenen IDs weiterer Content-Objekte, welche in Beziehung mit dem jeweils übergeordneten Content-Objekts als Unterabschnitte zu bezeichnen sind.

String type

Der Typ enthält Informationen darüber, welche Art von Darstellung innerhalb des “Guide“-Tools für die Benutzeroberfläche verwendet werden soll. Das bietet den Vorteil, dass bei dem Erstellen der JSON-Datei für den Inhalt dieser Funktion keine Kenntnisse über die softwareinterne Lösung vorhanden sein muss, sondern der Ersteller nur vordefinierte Begriffe eintragen muss, die dann softwareintern den jeweiligen Bildschirminhalt generieren.

Hinweis: Im Rahmen der Prototypentwicklung sind zwei verschiedene Arten (type) der Darstellung von Content-Objekten vorgesehen:

- 1. Die erste Möglichkeit ist das Erzeugen eines Buttons (“ButtonListWidget“), welcher den Titel des Content-Objektes anzeigt und bei Antippen auf eine nächste Seite führt und dort die jeweiligen subsections anzeigt.*
- 2. Die zweite Möglichkeit ist das Generieren eines ausklappbaren Elements (“ExtensionPanelWidget“), welches den Titel des Content-Objektes anzeigt und dessen Beschreibung bei Ausklappen sichtbar wird.*

A.4.8. **AppContentLoader-Klasse**

Die Klasse **AppContentLoader** stellt die Schnittstelle zum gewählten Git-Repository dar und trägt in diesem Umfang die Aufgabe, die in dem Repository-Ordner angelegten JSON-Dateien durch HTTP-Requests einzulesen und dem System zur weiteren Verarbeitung zur Verfügung zu stellen. Dies wird umgesetzt, indem diese eingelesenen Dateien in lokaler Form auf dem Endgerät gespeichert werden.

Die Klasse besitzt lediglich zwei Datenfelder

- **String httpRequest**
Beinhaltet die auszuführende HTTP-Anfrage.
- **List<JSONAppContentFile> _jsonFiles**
Beinhaltet die eingelesenen JSON-Dateien

Weiterhin dient die Klasse als Werkzeug zur Generierung des Inhaltes der Software-Sektion “Bachelorarbeit Guide“ der Applikation. Die Content-Objekte werden durch die jeweilige JSON-Datei, welche zuvor aus dem Git-Repository ausgelesen und gespeichert wurde generiert. Jedes Content-Objekt kann mehrere Content-Objekte in Form der jeweiligen IDs referenzieren, welche in diesem Rahmen als subsections bezeichnet werden.

A.4.9. **JSONAppContentFile-Klasse**

Eine Instanz dieser Klasse, die durch eine Objektinstanz der Klasse **AppContentLoader** erzeugt wird, stellt eine aus dem Repository eingelesene JSON-Datei dar und enthält den Namen der im Repository vorliegenden JSON-Datei, welche auch weiterhin als Dateiname in dem lokalen Speicher des Endgeräts verwendet.

Die durch den **AppContentBuilder** ermittelte jeweilige Download-Adresse der verschiedenen im Repository liegenden Dateien, dient zum Herunterladen des jeweiligen Inhaltes der jeweiligen JSON-Datei, was wiederum durch die Methode **Future<String> getJsonContent()**, im Detail durch ein HTTP-Request umgesetzt wird.

A.4.10. **Achievement-Notification Funktionen**

Für die Implementierung der Notification wurde ein externes Plugin genutzt (siehe [13]), da die grundsätzliche Möglichkeit der Handhabung von Notifications durch das Framework Flutter nicht gegeben ist. Durch die Verwendung des Plugins sind jedoch einige zielplattformabhängige Konfigurationen nötig, welche sich unter [13, Android Integration] und [13, iOS Integration] nachvollziehen lassen.

Bei der Umsetzung der Notifications wurde sich an den dort gegebenen Beispielen orientiert, wobei deren Funktionsweisen für die eigene Umsetzung adaptiert wurden.

Die Verwendung der Achievement-Notifications wird im Rahmen der prototypischen Implementierung durch spezielle Top-Level-Funktionen gelöst, welche bei Aufrufen die jeweiligen Achievement-Actions auslösen. Die dadurch hervorgerufene Aktualisierung der zu den Achievements zugehörigen Bedingungen findet in der **updateProperties Reducer-Funktion** statt. Hier werden die Nutzeraktionen an zentraler Stelle im System abgefragt und die Änderungen an den aktuellen, im Zustand gespeicherten, Bedingungen durchgeführt. Somit ist dies die zentral zu erweiternde Stelle im Code, wenn neue Achievements Einzug in das System erhalten.

A.5. Dokumente der Evaluation

A.5.1. Aufgabenkatalog

Fragebogen Benutzerevaluation

Ablauf

1. Folge den Anweisungen auf dem Bildschirm. (Bearbeitungszeit konfigurieren)
2. Was siehst du nun auf dem Bildschirm? (Dashboard)
3. Informiere dich über die Anmeldung der Bachelorarbeit. (Bachelorarbeit Guide)
4. Erstelle zwei Meilensteine (Meilensteinverwaltung)
5. Was ist passiert? Begib dich zu den Achievements. (Achievements)
6. Füge den Meilensteinen Unteraufgaben hinzu (Meilensteinverwaltung)
7. Bearbeite den Meilenstein (Datum ändern)
8. Begib dich zum Dashboard (Dashboard)
9. Hake eine Unteraufgabe ab (Dashboard – Meilenstein Widget)
10. Was denkst du über den Zeitstrahl? (Dashboard – Zeitstrahl Widget)
11. Ändere dein Datum deiner Bearbeitungszeit. (Bearbeitungszeit neu konfigurieren)

Fragebogen für Studierende

Was ist dir bei der Benutzung der Software negativ aufgefallen? Welche Probleme sind aufgetreten und wo gibt es Verbesserungspotential?
Glaubst du eine solche Software würde dir bei der Bearbeitung deiner Bachelorarbeit eine sinnvolle Hilfe bieten können?
Welche Funktionen der Software würdest du nutzen wollen? Warum würdest du sie nutzen?
Welche Funktionen der Software würdest du nicht nutzen wollen? Warum würdest du sie nicht nutzen?
Welchen Effekt hatte das Anzeigen von Erfolgen auf dich?
Hast du weitere Wünsche/Anregungen für hilfreiche Funktionen der Software? Was würde dich bei der Bearbeitung der Bachelorarbeit unterstützen?

A.5.2. Ergebnisse der Professoren

Softwareevaluation Testdurchführung Prof 1

Startbildschirm

Es wurden verschiedene Anmerkungen zu dem Bildschirm geäußert, auf dem die Datumseingabe für die Bearbeitungszeit der Bachelorarbeit erfolgt. Zum einen wurde kritisiert, dass es sich bei dem Datum um das amerikanische Datumsformat MM/DD/YYYY handelt und dies verwirrend ist.

Es wird sich weiterhin gewünscht, dass Informationskästchen zu der Bachelorarbeit Bearbeitungszeit auf dem Bildschirm platziert werden.

Dashboard

Das Dashboard wurde korrekt beschrieben und es wurde selbstständig erkannt, dass erst Meilensteine angelegt werden müssen, damit der Zeitstrahl seinen Zweck erfüllen kann.

Nach befüllen des Zeitstrahls wurde angemerkt, dass sich das Anwählen die jeweiligen platzierten Meilensteine erleichtern würde, wenn über eine andere Visualisierung oder größere Markierungen nachgedacht würde.

Meilenstein-Tool

Grundsätzlich wurden alle Meilenstein Tool spezifischen Aufgaben erfolgreich gelöst, jedoch kam es bei dem Bestätigen von Unteraufgaben aufgrund der nicht erkennbaren Bestätigungsmöglichkeit zu Schwierigkeiten.

Herausforderungs-Tool

Zu den Herausforderungen in der Applikation wurden verschiedene Anmerkungen geäußert:

- Selbständiges Erinnern an Herausforderungen wäre nett
- Verbindung zu anderen Features der Applikation wie Informations-Tool, nicht zu isolierte Features, da diese sich sehr gut ergänzen könnten
- SEHR WICHTIG ist die Formulierung der Herausforderungen. Es wäre nicht sinnvoll, wenn man nur Achievements bekommt, wenn man etwas angefangen hat.
- Mechanismus um Selbstbetrug zu verhindern wäre wünschenswert

Allgemein

Es wurde die Idee geäußert, vielleicht eine vordefinierte Beispielschablone einzufügen, welche den Anwendern einen Eindruck über die Features der Applikation und deren Verwendung ermöglicht.

Softwareevaluation Testdurchführung Prof2

Startbildschirm

Aufgaben ausgeführt – keine Anmerkungen

Dashboard

Alle Dashboard-Elemente wurden korrekt beschrieben. Des Weiteren wurden nach anlegen von Meilensteinen Anmerkungen gemacht, die den Zeitstrahl und dessen Nützlichkeit betreffen.

Hierbei wurden die übermittelten Informationen als sehr sinnvoll bezeichnet da sie die folgenden Aspekte fördern oder abdecken:

- Diszipliniert Studenten sich überhaupt ein Plan zu machen
 - o Welche Meilensteine habe ich?
 - o Wie lange brauche ich?
- Die einfache Möglichkeit der Übersicht
 - o Wie weit bin ich denn gerade?
 - o Bin ich früh oder zu spät dran?

Weiterhin wurden das Dashboard und die Visualisierung des Arbeitsfortschritts als sinnvoll bezeichnet, da der Anwender bei Öffnen der Applikation direkt mit dem aktuellen Stand konfrontiert wird.

Meilenstein-Tool

Die Aufgaben zu dem Meilenstein-Tool wurden korrekt ausgeführt, jedoch wurde fälschlicherweise auf den Zeitstrahl auf dem Dashboard geklickt um zu dem Meilensteinbildschirm zu gelangen.

Achievements-Tool

Achievements und Herausforderung werden als wenig trennscharf betitelt. Es wird gewünscht, dass klassische Gamification-Badges Einzug in die Applikation erhalten.

Herausforderungs-Tool

Die Herausforderungen wurden als sinnvoll bezeichnet, da sie die folgenden Aspekte erfüllen:

- Dient als Checkliste
 - o Man kann überprüfen, ob man alle relevanten Inhalte einer Bachelorarbeit bedacht hat
 - o Man wird auf Dinge hingewiesen, die man sonst nicht rechtzeitig bedenkt
 - o Kann einen auffordernden Charakter haben
 - o Gute Ergänzung zu individuellen Meilensteintool durch stärkere Vorgliederung
 - o Orientierungsmöglichkeit

A.5.3. Ergebnisse der Studierende

Usability Test Protokoll: Stud 1

Aufgabe: Folge den Anweisungen auf dem Bildschirm.

- Aufgabe gelöst
- Keine Anmerkungen

Aufgabe: Was siehst du nun auf dem Bildschirm?

- Aufgabe gelöst
- Keine Anmerkungen

Aufgabe: Informiere dich über die Anmeldung der Bachelorarbeit.

- Aufgabe gelöst
- Keine Anmerkungen

Aufgabe: Erstelle zwei Meilensteine

- Aufgabe gelöst
- Keine Anmerkungen

Aufgabe: Was ist passiert? Gehe zu den Achievements

- Aufgabe gelöst
- Keine Anmerkung

Aufgabe: Füge den Meilensteinen Unteraufgaben hinzu.

- Aufgabe nach Hilfestellung gelöst
- Es kam zu Irritationen bei dem Bestätigen, da die Tastatur nicht als Bestätigungsmöglichkeit genutzt wird

Aufgabe: Bearbeite den Meilenstein (Datum ändern)

- Aufgabe gelöst
- Keine Anmerkung

Aufgabe: Begib dich zum Dashboard

- Aufgabe gelöst
- Keine Anmerkung

Aufgabe: Hake eine Unteraufgabe ab

- Aufgabe gelöst
- Keine Anmerkung

Aufgabe: Schließe eine Herausforderung ab

- Aufgabe gelöst
- Keine Anmerkung

Was denkst du über den Zeitstrahl?

- Zeitstrahl wird als nützlich und hilfreich empfunden, solange eine gute Übersicht vorhanden bleibt.

Was ist dir bei der Benutzung der Software negativ aufgefallen? Welche Probleme sind aufgetreten und wo gibt es Verbesserungspotential?

Es wäre gut, wenn eine Nachfrage bei Löschen einer Aufgabe vorhanden wäre, falls man ausversehen eine falsche Aufgabe löscht.

Glaubst du eine solche Software würde dir bei der Bearbeitung deiner Bachelorarbeit eine sinnvolle Hilfe bieten können?

Definitiv

Welche Funktionen der Software würdest du nutzen wollen? Warum würdest du sie nutzen?

Ich würde jede Funktion nutzen wollen, da ich von Natur aus sehr unorganisiert bin.

Welche Funktionen der Software würdest du nicht nutzen wollen? Warum würdest du sie nicht nutzen?

-

Welchen Effekt hatte das Anzeigen von Erfolgen/Achievements auf dich?

Einen sehr positiven Effekt, da es mich motiviert.

Hast du weitere Wünsche/Anregungen für hilfreiche Funktionen der Software? Was würde dich bei der Bearbeitung der Bachelorarbeit unterstützen?

-

Usability Test Protokoll: Stud 2

Aufgabe: Folge den Anweisungen auf dem Bildschirm.

- Aufgabe gelöst
- Keine Anmerkungen

Aufgabe: Was siehst du nun auf dem Bildschirm?

- Aufgabe gelöst
- Keine Anmerkungen

Aufgabe: Informiere dich über die Anmeldung der Bachelorarbeit.

- Aufgabe unter Hilfestellung gelöst
- Die als „Bachelorarbeit Guide“ betitelte Funktion der Applikation wurde im Rahmen der Aufgabenstellung nicht als Ziel identifiziert. Die Testperson wechselt hektisch zwischen den verschiedenen Funktionen hin und her, bis der Moderator auf die richtige Kategorie hingewiesen hat.

Aufgabe: Erstelle zwei Meilensteine

- Aufgabe gelöst
- Fälschlicherweise auf den Zeitstrahl des Dashboards gedrückt um Meilensteine anzulegen

Aufgabe: Was ist passiert? Gehe zu den Achievements

- Aufgabe gelöst
- Keine Anmerkung

Aufgabe: Füge den Meilensteinen Unteraufgaben hinzu.

- Aufgabe gelöst
- Keine Anmerkung

Aufgabe: Bearbeite den Meilenstein (Datum ändern)

- Aufgabe gelöst
- Keine Anmerkung

Aufgabe: Begib dich zum Dashboard

- Aufgabe gelöst
- Keine Anmerkung

Aufgabe: Hake eine Unteraufgabe ab

- Aufgabe gelöst
- Keine Anmerkung

Aufgabe: Schließe eine Herausforderung ab

- Aufgabe gelöst
- Keine Anmerkung

Was denkst du über den Zeitstrahl?

- Grundsätzlich wurden die durch den Zeitstrahl übermittelten Informationen als sinnvoll bewertet, jedoch wurde auch der Wunsch nach einer klassischen Kalenderansicht geäußert.

Was ist dir bei der Benutzung der Software negativ aufgefallen? Welche Probleme sind aufgetreten und wo gibt es Verbesserungspotential?

-

Glaubst du eine solche Software würde dir bei der Bearbeitung deiner Bachelorarbeit eine sinnvolle Hilfe bieten können?

Hilfreich ja, aber nutzen würde ich sie wahrscheinlich nicht, weil ich zu faul bin.

Welche Funktionen der Software würdest du nutzen wollen? Warum würdest du sie nutzen?

Die Herausforderungen sind eine gute Hilfeleistung als Leitfaden.

Welche Funktionen der Software würdest du nicht nutzen wollen? Warum würdest du sie nicht nutzen?

-

Welchen Effekt hatte das Anzeigen von Erfolgen/Achievements auf dich?

Kein Effekt

Hast du weitere Wünsche/Anregungen für hilfreiche Funktionen der Software? Was würde dich bei der Bearbeitung der Bachelorarbeit unterstützen?

-

Usability Test Protokoll: Stud 3

Aufgabe: Folge den Anweisungen auf dem Bildschirm.

- Aufgabe gelöst
- Keine Anmerkungen

Aufgabe: Was siehst du nun auf dem Bildschirm?

- Aufgabe gelöst
- Keine Anmerkungen

Aufgabe: Informiere dich über die Anmeldung der Bachelorarbeit.

- Aufgabe gelöst
- Keine Anmerkungen

Aufgabe: Erstelle zwei Meilensteine

- Fälschlicherweise auf den Zeitstrahl des Dashboards gedrückt um Meilensteine anzulegen
- Aufgabe gelöst

Aufgabe: Was ist passiert? Gehe zu den Achievements

- Achievements erst nicht durch Push Benachrichtigung als Achievements wahrgenommen
- Aufgabe gelöst

Aufgabe: Füge den Meilensteinen Unteraufgaben hinzu.

- Aufgabe nach Hilfestellung gelöst
- Ausklappbare Aufgaben wurden nicht gesehen, was zu Irritationen geführt hat.
- Feedback Meldung bei hinzufügen wird gewünscht (Beispiel: „Unteraufgabe wurde gespeichert“)

Aufgabe: Bearbeite den Meilenstein (Datum ändern)

- Aufgabe gelöst
- Keine Anmerkung

Aufgabe: Begib dich zum Dashboard

- Aufgabe gelöst
- Keine Anmerkung

Aufgabe: Hake eine Unteraufgabe ab

- Aufgabe gelöst
- Keine Anmerkung

Aufgabe: Schließe eine Herausforderung ab

- Aufgabe gelöst
- Keine Anmerkung

Was denkst du über den Zeitstrahl?

- Darstellung der Informationen wird als „sehr sinnvoll“ bezeichnet, nur die Darstellung an sich wird als zu unhandlich kritisiert

Was ist dir bei der Benutzung der Software negativ aufgefallen? Welche Probleme sind aufgetreten und wo gibt es Verbesserungspotential?

Mehr Feedback beim Erstellen von Aufgaben.

Glaubst du eine solche Software würde dir bei der Bearbeitung deiner Bachelorarbeit eine sinnvolle Hilfe bieten können?

Ja

Welche Funktionen der Software würdest du nutzen wollen? Warum würdest du sie nutzen?

Informationen und Herausforderungen bieten eine gute Anlaufstelle zur Vorbereitung

Welche Funktionen der Software würdest du nicht nutzen wollen? Warum würdest du sie nicht nutzen?

Aufgabenplanung. Ich habe auch keinen Kalender – Ich habe alles im Kopf

Welchen Effekt hatte das Anzeigen von Erfolgen/Achievements auf dich?

Keinen.

Hast du weitere Wünsche/Anregungen für hilfreiche Funktionen der Software? Was würde dich bei der Bearbeitung der Bachelorarbeit unterstützen?

-

Usability Test Protokoll: Stud 4

Aufgabe: Folge den Anweisungen auf dem Bildschirm.

- Aufgabe gelöst
- Keine Anmerkungen

Aufgabe: Was siehst du nun auf dem Bildschirm?

- Aufgabe gelöst
- Keine Anmerkungen

Aufgabe: Informiere dich über die Anmeldung der Bachelorarbeit.

- Aufgabe gelöst
- Keine Anmerkungen

Aufgabe: Erstelle zwei Meilensteine

- Aufgabe gelöst
- Keine Anmerkungen

Aufgabe: Was ist passiert? Gehe zu den Achievements

- Aufgabe gelöst
- Keine Anmerkung

Aufgabe: Füge den Meilensteinen Unteraufgaben hinzu.

- Aufgabe nach Hilfestellung gelöst
- Es kam zu Irritationen bei dem Bestätigen, da die Tastatur nicht als Bestätigungsmöglichkeit genutzt wird

Aufgabe: Bearbeite den Meilenstein (Datum ändern)

- Aufgabe gelöst
- Keine Anmerkung

Aufgabe: Begib dich zum Dashboard

- Aufgabe gelöst
- Keine Anmerkung

Aufgabe: Hake eine Unteraufgabe ab

- Aufgabe gelöst
- Keine Anmerkung

Aufgabe: Schließe eine Herausforderung ab

- Aufgabe gelöst
- Keine Anmerkung

Was denkst du über den Zeitstrahl?

- Zeitstrahl wird als „sinnvolle Sache“ für den Verlauf der Arbeit beschrieben, jedoch wird die unvorteilhaft und zu kleine Darstellung kritisiert.

Was ist dir bei der Benutzung der Software negativ aufgefallen? Welche Probleme sind aufgetreten und wo gibt es Verbesserungspotential?
-
Glaubst du eine solche Software würde dir bei der Bearbeitung deiner Bachelorarbeit eine sinnvolle Hilfe bieten können?
Ja
Welche Funktionen der Software würdest du nutzen wollen? Warum würdest du sie nutzen?
Alle außer Erfolge
Welche Funktionen der Software würdest du nicht nutzen wollen? Warum würdest du sie nicht nutzen?
Ich würde Erfolge wahrscheinlich nur nachgucken aber nicht abhaken.
Welchen Effekt hatte das Anzeigen von Erfolgen/Achievements auf dich?
Achievements sind eins netten Feature, was von ich aus Videospielen kenne. Ich finde es auf jeden Fall sehr lustig, dass ich von meiner App mit einem Feedback gelobt werde und irgendwie motiviert es mich auch.
Hast du weitere Wünsche/Anregungen für hilfreiche Funktionen der Software? Was würde dich bei der Bearbeitung der Bachelorarbeit unterstützen?
-

Literatur

- [1] *Actions*. (Zugriff Juni 8, 2018). URL: <https://redux.js.org/basics/actions>.
- [2] (Prof. Dr. rer. nat. Andreas Hanemann). “Anleitung zum Erstellen einer Abschlussarbeit”. Herausgegeben an Studenten der Fachhochschule Lübeck, die das Bachelorseminar besuchen.
- [3] *Announcing Flutter beta 1: Build beautiful native apps*. Februar 27, 2018 (Zugriff Juni 3, 2018). URL: <https://developers.googleblog.com/2018/02/announcing-flutter-beta-1.html>.
- [4] Helmut Balzert. *Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering* -. Berlin Heidelberg New York: Springer-Verlag, 2010. ISBN: 978-3-827-42247-7.
- [5] Richard Bartle. “Hearts, clubs, diamonds, spades: Players who suit MUDs”. In: *Journal of MUD research* 1.1 (1996), S. 19.
- [6] Ivo Blohm und Jan Marco Leimeister. “Gamification: Gestaltung IT-basierter Zusatzdienstleistungen zur Motivationsunterstützung und Verhaltensänderung”. In: *Wirtschaftsinformatik* 55.4 (2013), S. 275–278.
- [7] *Building Layouts in Flutter*. Februar 27, 2018 (Zugriff Juni 3, 2018). URL: <https://flutter.io/tutorials/layout/>.
- [8] *Completer<T> class*. (Zugriff Juni 16, 2018). URL: <https://api.dartlang.org/stable/2.0.0/dart-async/Completer-class.html>.
- [9] *Cupertino (iOS-style) Widgets*. (Zugriff Juni 3, 2018). URL: <https://flutter.io/widgets/cupertino/>.
- [10] *Dart Homepage*. (Zugriff Juni 10, 2018). URL: <https://www.dartlang.org/>.
- [11] Sebastian Deterding u. a. “Gamification: Toward a definition”. In: *CHI 2011 gamification workshop proceedings*. Bd. 12. Vancouver BC, Canada. 2011.
- [12] *Flutter Charting Libary*. (Zugriff August 20, 2018). URL: https://pub.dartlang.org/packages/charts_flutter.
- [13] *Flutter Local Notification Plugin*. (Zugriff Juni 16, 2018). URL: https://pub.dartlang.org/packages/flutter_local_notifications.
- [14] *Flutter Performance Profiler*. (Zugriff Juni 10, 2018). URL: <https://flutter.io/ui-performance/>.
- [15] *Flutter Redux*. (Zugriff Juni 8, 2018). URL: https://pub.dartlang.org/packages/flutter_redux.
- [16] *Flutter System Architecture*. April 23, 2017 (Zugriff Juni 3, 2018). URL: https://docs.google.com/presentation/d/1cw7A4HbvM_Abv320rVgPVGiUP2msVs7tfGbkgrTy0I/edit#slide=id.p.

- [17] *Foundation Library*. (Zugriff August 6, 2018). URL: <https://docs.flutter.io/flutter/foundation/foundation-library.html>.
- [18] *Get Startet*. (Zugriff Juni 10, 2018). URL: <https://flutter.io/get-started/>.
- [19] *How to Code Unlockable Achievements for Your Game (A Simple Approach)*. (Zugriff August 16, 2018). URL: <https://gamedevelopment.tutsplus.com/tutorials/how-to-code-unlockable-achievements-for-your-game-a-simple-approach--gamedev-6012>.
- [20] *iOS Design Themes*. (Zugriff Juni 3, 2018). URL: <https://developer.apple.com/design/human-interface-guidelines/ios/overview/themes/>.
- [21] *JSON and serialization*. (Zugriff August 20, 2018). URL: <https://flutter.io/json/#code-generation>.
- [22] *Json Serializable*. (Zugriff August 20, 2018). URL: https://pub.dartlang.org/packages/json_serializable.
- [23] Prof. Dr. Rainer Koschke. *Software-Projekt*. (Zugriff August 22, 2018). URL: <https://www.informatik.uni-bremen.de/st/Lehre/swp0809/skript/analyse-1x2.pdf>.
- [24] *Material Design*. (Zugriff Juni 3, 2018). URL: <https://material.io/design/>.
- [25] Philipp Mayring. *Qualitative Inhaltsanalyse - Grundlagen und Techniken*. Langensalza: Beltz, 2015. ISBN: 978-3-407-25730-7.
- [26] Elisa D Mekler u. a. "Disassembling gamification: the effects of points and meaning on user motivation and performance". In: *CHI'13 extended abstracts on human factors in computing systems*. ACM. 2013, S. 1137–1142.
- [27] Elisa D Mekler u. a. "Do points, levels and leaderboards harm intrinsic motivation?: an empirical analysis of common gamification elements". In: *Proceedings of the First International Conference on gameful design, research, and applications*. ACM. 2013, S. 66–73.
- [28] *Mobile World Congress*. (Zugriff August 6, 2018). URL: <https://www.mobileworldcongress.com/about/>.
- [29] *Path Provider*. (Zugriff August 20, 2018). URL: https://pub.dartlang.org/packages/path_provider.
- [30] Nicolas Pflanzl. *Gamification for business process modeling*. 2018.
- [31] *Pixabay*. (Zugriff August 18, 2018). URL: <https://pixabay.com/de/abzeichen-blau-button-aufkleber-150755/>.
- [32] *Read Me*. (Zugriff Juni 8, 2018). URL: <https://redux.js.org/>.
- [33] *Reducers*. (Zugriff Juni 8, 2018). URL: <https://redux.js.org/basics/reducers>.
- [34] *Redux*. (Zugriff Juni 8, 2018). URL: <https://reduxframework.com/>.
- [35] *Redux (JavaScript-Bibliothek)*. (Zugriff Juni 8, 2018). URL: [https://de.wikipedia.org/wiki/Redux_\(JavaScript-Bibliothek\)](https://de.wikipedia.org/wiki/Redux_(JavaScript-Bibliothek)).
- [36] *Redux Persist*. (Zugriff August 20, 2018). URL: https://pub.dartlang.org/packages/redux_persist.

- [37] Byron Reeves und J Leighton Read. *Total engagement: How games and virtual worlds are changing the way people work and businesses compete*. Harvard Business Press, 2009.
- [38] Michael Sailer. *Die Wirkung von Gamification auf Motivation und Leistung - Empirische Studien im Kontext manueller Arbeitsprozesse*. Berlin Heidelberg New York: Springer-Verlag, 2016. ISBN: 978-3-658-14309-1.
- [39] *Skia Graphics Library*. (Zugriff Juni 3, 2018). URL: <https://skia.org/>.
- [40] *Store*. (Zugriff Juni 8, 2018). URL: <https://redux.js.org/basics/store>.
- [41] *Technical Overview*. (Zugriff Juni 3, 2018). URL: <https://flutter.io/technical-overview/>.
- [42] *Technology*. (Zugriff Juni 3, 2018). URL: <https://flutter.io/faq/>.
- [43] *Welcome to the Engine Wiki*. (Zugriff Juni 3, 2018). URL: <https://github.com/flutter/engine/wiki>.
- [44] Kevin Werbach und Dan Hunter. *For the win: How game thinking can revolutionize your business*. Wharton Digital Press, 2012.
- [45] *What's Revolutionary about Flutter*. (Zugriff Juni 3, 2018). URL: <https://hackernoon.com/whats-revolutionary-about-flutter-946915b09514>.