

Digital unterstützte universitäre Anfangslehre in der Informatik unter Berücksichtigung heterogener Gruppen

Dissertation

zur Erlangung des Doktorgrades (Dr. rer. nat.) des Fachbereichs
Mathematik/Informatik der Universität Osnabrück

vorgelegt von

Elisaweta Ossovski

März 2023

Datum der Disputation: 23.06.2023

Erstgutachter: Prof. Dr. Michael Brinkmeier, Universität Osnabrück

Zweitgutachter: Prof. Dr. Karsten Morisse, Hochschule Osnabrück

Danksagung

Ich möchte mich bei allen Personen, die mich bei meiner Promotion unterstützt haben, herzlich bedanken. Insbesondere gilt mein Dank:

- Meinem Doktorvater Prof. Dr. Michael Brinkmeier für die umfassende Betreuung in den letzten Jahren, die Idee und Bereitschaft das Modul in einen Flipped Classroom umzugestalten, viele wertvolle Gespräche und angenehmes Arbeiten.
- Prof. Dr. Karsten Morisse für den angenehmen Austausch über die Lehre und die Bereitschaft zur Übernahme des Zweitgutachtens.
- Dem Ernst-Ludwig-Ehrlich-Studienwerk für die ideelle und finanzielle Förderung während meiner Promotion. Danke an Dr. Maria Ulatowski, Dr. David Kowalski und Dr. Michal Or sowie meinen Mitstipendiaten.
- Meinen Kollegen Laura, Agnes, Daniel, Ann-Katrin, Sven und Andreas für eine immer gute Zusammenarbeit, einen angenehmen Austausch und Unterstützung.
- Den Mitarbeitern der Universität: Friedhelm Hofmeyer, den Teams der Hochschuldidaktik, der KoPro, des Virtuos und der AG Software Engineering für die Unterstützung, Zusammenarbeit, Workshops und den Austausch.
- Den Organisatoren und Teilnehmenden der Konferenzen und Doktorandenkolloquien, die ich während der Promotion besucht habe und bei denen ich viel lernen konnte. Danke an Simone, Franziska, Jennifer, Nathalie und viele weitere!
- Den Personen und Einrichtungen, die an weiteren Projekten beteiligt waren, die es nicht in die Dissertation geschafft, aber mir wichtige Erfahrungen in der Forschung ermöglichen haben. Danke an das GSG und die Informatikdidaktiker der HU Berlin!
- Den Tutoren und den Studierenden des Moduls AuD für die Bereitschaft zur Teilnahme an den Befragungen und die Akzeptanz des Konzepts trotz anfänglichen Schwierigkeiten. Danke an Lara für die Hilfe beim Übersetzen des Fragebogens!
- Agnes und Fiona für das Korrekturlesen dieser Arbeit.
- Meiner Familie und Freunden für das Zuhören, Mut machen und die Unterstützung während der gesamten Zeit meines Studiums und der Promotion. Besonders danke ich meinen Eltern Inna und Igor, meinem Bruder David, meinen Großeltern, Niklas, Eva, Laura, Kim, Ann-Kristin, Fiona und Ilona!

Da in dieser Arbeit im Rahmen von Gamification Easteregg eine besondere Rolle spielen, ist während der Verschriftlichung der Dissertation eine Kollateralproduktivität¹ [Neu18] entstanden, die als Danksagung an die Musiker zu verstehen ist, die mich (eher unwissentlich) durch musikalischen Ausgleich in der Freizeit unterstützt haben.

¹für die Druckversion: <https://abbozza.informatik.uos.de/lisa/easteregg.php>

Zusammenfassung

Programmierenlernen stellt eine hohe kognitive Belastung dar. Aufgrund unterschiedlicher Vorkenntnisse und Motivation, abhängig vom Studienziel, liegt insbesondere in der studiengangübergreifenden Anfangslehre der Informatik eine hohe Heterogenität bei den Studierenden vor. Der vorliegende Mantelbeitrag zur kumulativen Dissertation bettet Forschungsbeiträge ein, die den Gesamtkontext und ausgewählte Aspekte der Herausforderungen der Lehre in diesem Bereich untersuchen. Dabei werden die Möglichkeiten und Vorteile digitaler Werkzeuge in der Lehre berücksichtigt.

Im Zentrum der Arbeit steht dabei ein iterativ über mehrere Jahre gestaltetes und evaluiertes Modulkonzept für ein studiengangübergreifendes Einführungsmodul in der Informatik. Als Basis diente dabei das Konzept des Flipped Classrooms. Bei diesem wird die Inhaltsvermittlung mit geeignetem Material auf Selbstlernphasen ausgelagert, während Aktivitäten zur Anwendung und Übung in synchronen Veranstaltungen stattfinden, um die Vorteile gemeinsamer Arbeit und direkter Hilfestellung zu nutzen. Unterstützt durch Aspekte der Projektarbeit, Gamification und wachsenden Abgabezeiträumen für verpflichtende Aufgaben entstand dabei ein Konzept, das sich für Lernende mit unterschiedlichen Graden an Vorwissen und diversen Studienfächern zu eignen scheint, wie die Evaluationsergebnisse auf Ebene der Studierenden und Tutoren zeigen. Eine Studie zu fünf verschiedenen Arten digitaler Lernmaterialien, die sich in Perfektionismusanspruch sowie in der Art als Live Coding oder Worked Example unterscheiden, ergänzt die praxisbezogene Forschung. Dabei konnten keine signifikanten Unterschiede bezüglich der Lerntransferleistung der Teilnehmenden mit verschiedenen Lernmaterialien festgestellt werden. Aus der Befragung geht hervor, dass Studierende Kombinationen verschiedener Arten bevorzugen und Fehler auch als Lerngelegenheiten auffassen können.

Zur genaueren Erforschung der Heterogenität unter den Studierenden des Moduls wurden zwei Studien durchgeführt. Eine qualitative Untersuchung beschäftigt sich mit den Einstellungen und Erwartungen zum Modul zu dessen Beginn. Durch Interviews mit ausgewählten Studierenden konnte dabei festgestellt werden, dass der Vorkurs und die ersten Vorlesungen, aber auch der Studiengang und Vorkenntnisse einen Einfluss auf die Vorstellungen der Teilnehmenden haben. Studierende thematisch entfernterer Studiengänge ohne Vorkenntnisse erwarteten teilweise Inhalte, die nicht zur Informatik gehören. Eine weitere Studie beschäftigt sich quantitativ mit der Bedeutung des Begriffs der Frustrationstoleranz für das Studium der Informatik in der Anfangsphase. Dabei wurden die Daten für einen psychologischen Fragebogen zur Frustrationstoleranz sowie eine Selbsteinschätzung am Anfang und am Ende des Kurses erfasst und analysiert. Dabei zeigte sich, dass die erfasste Frustrationstoleranz sich durch den Informatikkurs nicht signifikant änderte. Die Verwendung des Begriffs der Frustrationstoleranz als Voraussetzung oder Ziel eines Informatikstudiums sollte daher überdacht werden.

Insgesamt beschreibt die Arbeit Untersuchungen auf unterschiedlichen Ebenen eines Einführungsmoduls der Informatik mit heterogenen Teilnehmenden hinsichtlich der Vorkenntnisse und Motivation. Es werden Ansätze dargelegt mit den erfassten unterschiedlichen Bedürfnissen umzugehen, Materialien lernförderlich zu gestalten und zeitgemäße Lehrkonzepte, die Möglichkeiten der Digitalisierung gewinnbringend nutzen, zu entwickeln.

Abstract

Learning to program involves a high cognitive load. Due to different prior knowledge and motivation depending on the study programme there is a high heterogeneity among students especially in the cross-curricular initial teaching of Computer Science. This cumulative dissertation synopsis embeds research contributions that investigate the overall context and selected aspects of the challenges of university teaching in this area. The possibilities and advantages of digital tools in teaching are considered.

The core of this thesis is a module concept for a cross-curricular introductory module in computer science that was designed and evaluated iteratively over multiple years. The concept of the Flipped Classroom served as the basis. In this concept, the transmission of content with suitable material is outsourced to self-learning phases, while activities for application and practice take place in synchronous meetings in order to benefit from the advantages of collaborative work and direct assistance. A concept involving aspects of project work, gamification and increasing submission periods for compulsory tasks was developed that seems to be suitable for learners with different degrees of prior knowledge and diverse subjects, as the evaluation results at student and tutor levels show. A study conducted with five different types of digital learning materials, which vary in their perfectionism and in the way they are presented as live coding or worked examples, supplements the practice-oriented research. No significant differences were found in the learning transfer performance of the participants with different learning materials. The survey indicates that students prefer combinations of several types of material and can also perceive mistakes as learning opportunities.

Two studies were conducted in order to explore the heterogeneity among the students of the module in more detail. A qualitative study focuses on the attitudes and expectations towards the module at the beginning of it. Through interviews with selected students, it could be observed that the preliminary course and the first lecture dates, but also the major and previous knowledge, have an influence on the participants' conceptions. Students from thematically rather unrelated majors with no prior knowledge partially expected content that does not belong to Computer Science. Another study deals with the significance of the concept of Frustration Tolerance for the study of Computer Science in the initial phase in quantitative terms. This involved collecting data for a psychological questionnaire on Frustration Tolerance and the participants' self-assessment at the beginning and at the end of the course for an analysis. It was found that the questionnaire-assessed Frustration Tolerance did not change significantly after taking a Computer Science course. The use of the term Frustration Tolerance as a prerequisite or goal of studying Computer Science should therefore be reconsidered.

Overall, the paper describes investigations at different levels of an introductory Computer Science module with heterogeneous participants in terms of prior knowledge and motivation. Approaches are presented to handle the different needs recorded, to design materials that promote learning and to develop state-of-the-art teaching concepts that use the possibilities of digitalisation beneficially.

Inhaltsverzeichnis

I	Einleitung und theoretische Basis	1
1	Einleitung	3
1.1	Motivation	3
1.2	Aufbau der Arbeit	5
2	Literatur	7
2.1	Forschung zu Einführungsmodulen in der Informatik	7
2.1.1	Forschung zum Flipped Classroom	11
2.1.2	Forschung zu Projektarbeit	16
2.1.3	Forschung zu Gamification	19
2.2	Forschung zu digitalen Lernmaterialien	25
3	Methodik	31
3.1	Kontext der durchgeführten Studien	31
3.2	Übersicht über Veröffentlichungen	34
II	Teil A: Voraussetzungen – heterogene Gruppen an Hochschulen	37
4	Erwartungen an ein Einführungsmodul der Informatik	39
4.1	Einleitung	39
4.2	Theoretischer Hintergrund	40
4.2.1	Should- und Will-Expectations nach Boulding et al. [BKSZ93]	40
4.2.2	Bisherige Studienlage	41
4.3	Methodik	42
4.3.1	Datenerhebung und Fragen	42
4.3.2	Auswahl der Interviewten	43
4.4	Ergebnisse	45
4.4.1	Kategorien	45
4.4.2	Festgestellte Typen	47
4.4.3	Einflüsse auf Einstellungen und Erwartungen	48
4.5	Diskussion	48
4.6	Folgerungen	49

III Teil B: Praktische Gestaltung von Einführungsmodulen in der Informatik unter Berücksichtigung von Heterogenität und Digitalisierung	51
5 Modulkonzept	53
6 Weitere Ergebnisse	59
6.1 Modulkonzept im Wintersemester 2021/22	59
6.2 Quantitative Ergebnisse	61
6.3 Qualitative Ergebnisse	67
6.4 Diskussion	68
7 Lernmaterialien	73
IV Diskussion und Ausblick	77
8 Diskussion	79
8.1 Limitierungen	79
8.2 Diskussion der Ergebnisse	81
9 Folgerungen	85
9.1 Folgerungen für Modul	85
9.2 Generalisierung	87
10 Zusammenfassung	93
Literaturverzeichnis	99
Abbildungsverzeichnis	119
A Veröffentlichungen	121
A.1 Beitrag 1	121
A.2 Beitrag 2	145
A.3 Beitrag 3	157
A.4 Beitrag 4	165
A.5 Beitrag 5	177

Teil I

Einleitung und theoretische Basis

1 Einleitung

In diesem Kapitel werden die Motivation für die vorliegende Arbeit und deren Bestandteile sowie der Aufbau dieser Arbeit als Mantel der kumulativen Dissertation dargelegt.

1.1 Motivation

Programmierenlernen wird als schwierige Tätigkeit beschrieben [Jen02, RRR03]. Zudem sind Universitäten mit einer zunehmenden Heterogenität unter den Studierenden konfrontiert [Sei14]. Aufgrund der Situation von Informatik als Wahl- bzw. Wahlpflichtfach an Schulen [SHF21] besteht zusätzlich auf dieser Grundlage eine erhebliche Heterogenität an Vorkenntnissen für den Einstieg in die Informatik an der Universität. Durch Programmieren als Hobby sowie Ausbildungen im Bereich der Informatik wird diese Situation weiter verstärkt.

In studiengangübergreifenden Einstiegsmodulen in der Informatik kommt noch eine zweite Ebene der Heterogenität unter den Teilnehmenden hinzu. Während Studierende¹ stark informatiknaher Studiengänge diese als zentralen Aspekt ihres Studiums betrachten, können einige Studierende in nur leicht informatiknahen Studiengängen oft nicht nachvollziehen warum sie ein Modul in Informatik belegen müssen.

Trotz bereits vorhandener Entwicklungen in der insgesamt relativ neuen fachdidaktischen Lehr-Lern-Forschung (siehe Kapitel 2) bestanden und bestehen noch viele offene Fragen. Diese kommen vor allem dadurch zustande, dass das Erlernen komplexer Themengebiete von diversen Faktoren abhängt. Mit der vorliegenden Arbeit sollen einige Aspekte dieser aufgegriffen und anhand der Situation an der Universität Osnabrück erläutert werden. Dazu wurden Forschung und Lehrpraxis auf unterschiedlichen Ebenen eines Einführungsmoduls in der Informatik durchgeführt.

Insgesamt verfolgt die Arbeit die Ziele mehr über die Voraussetzungen der Studierenden in studiengangübergreifenden Einführungsmodulen der Informatik zu erfahren sowie die Lehre in solchen auf Ebene der Konzeptgestaltung, aber auch Lernmaterialien zu verbessern. Für gute Lehre müssen Lehrende auch die Lernenden genauer kennenlernen [Ram03, S. 99]. Durch die Kenntnis der Erwartungen Studierender ist es möglich diese mit der geplanten Lehre abzugleichen und sowohl Anpassungen bei Bedarfen der Teilnehmenden sowie bei den Erwartungen dieser vorzunehmen. Ziel ist dabei eine möglichst gute Passung zu erreichen. Doch auch die Erwartungen Lehrender in Informatikmodulen können von den gegebenen Bedingungen abweichen. In diesem Zusammenhang entstand

¹In dieser Arbeit wird aus Gründen der Lesbarkeit grundsätzlich eine geschlechtsneutrale Form, falls vorhanden, oder im anderen Falle ein generisches Maskulinum verwendet. Auch bei letzterem sind Personen jegliches Geschlechts explizit eingeschlossen.

auch die Frage, welche Rolle der Begriff der Frustrationstoleranz in der Anfangslehre der Informatik spielt und ob das Erlernen informatischer Inhalte Auswirkungen auf die Frustrationstoleranz als psychologisches Konzept hat.

Kern des Dissertationsprojekts war die Erstellung eines neuen Kurskonzepts auf Basis des Flipped Classroom. Ein zentrales Ziel war dabei die Studierenden stärker zu motivieren aktiv miteinander programmieren zu lernen. Damit sollte sowohl das Lernen selbst verbessert als auch der Charakter der zukünftigen Arbeit realistischer abbildet werden können. Insbesondere die zuvor eher passiv gestaltete Übung sollte einen echten Mehrwert für Studierende mit diversen Vorkenntnisstufen bringen.

Durch Untersuchungen zur aufgewendeten Studienzeit auch außerhalb der Anwesenheit bei Lehrveranstaltungen [SM11] wurde festgestellt, dass viele Studierende die Selbstlernzeiten im Vergleich zur durch die Studienordnungen erwarteten Zeit häufig verkürzen. Die Studierenden im Einführungsmodul der Informatik besitzen bei dessen Belegung im ersten Semester noch kaum Erfahrungen mit eigenständigem Lernen an einer Universität. Eine Hypothese für die Ursache eines als ähnlich zum beschriebenen vermuteten Verhaltens bestand daher darin, dass diese zu wenige Materialien und Anregungen haben, um die vorgesehene Selbstlernzeit optimal füllen zu können. Folglich war ein weiteres Ziel der Umstrukturierung die Erstellung von Materialien, die das eigenständige Lernen der Studierenden unterstützen.

Auf Detailebene stellt sich ebenso die Frage nach der Gestaltung digitaler Lernmaterialien. Obwohl es hierzu bereits einige Grundlagenforschung gibt, wurden Materialien mit verschiedenen Eigenschaften bisher kaum explizit für das Programmierenlernen erforscht. Dabei existieren neben grundsätzlich verschiedenen Möglichkeiten für digitale Materialien wie Videos oder interaktivem Lernmaterial für Programmierbeispiele zwei konkrete Gestaltungsmöglichkeiten. Beim *Live Coding* [BMNR14] wird der Programmierprozess, meist interaktiv, konkret durchgeführt. Dabei kann die Vorbereitung darauf entweder möglichst gering ausfallen, um den Prozess authentisch zu zeigen, wohingegen stärkere Vorbereitung Programmierfehlern vorbeugen kann. Im Gegensatz dazu werden bei *Worked Examples* [SLR14] vollendete Programme zur Verfügung gestellt, bei denen die Lernenden das Ergebnis als Lösung für eine Programmieraufgabe nachvollziehen sollen. Im Rahmen der Arbeit wurde in einer Fallstudie untersucht, welche dieser Arten von digitalen Lernmaterialien Studierende in diesem Kontext bevorzugen und ob die verschiedenen Gestaltungsmöglichkeiten Auswirkungen auf den Lerntransfererfolg haben.

Die Arbeit widmet sich daher mit den zugehörigen Publikationen vor allem folgenden Forschungsfragen:

- Welche Erwartungen haben Studierende an Einführungsmodule der Informatik?
- Welcher Zusammenhang besteht zwischen dem psychologischen Konzept der Frustrationstoleranz und dem Informatikstudium?
- Wie kann ein Einführungsmodul in der Informatik unter Einbezug digitaler Möglichkeiten gestaltet werden, um der Heterogenität der Studierenden in Vorwissen und Zielen angemessen zu begegnen?
 - Welche Anforderungen für Abgaben von Studierenden führen zu regelmäßiger und angemessener Mitarbeit, Zufriedenheit und Lernerfolg?

- Wie können Lehrveranstaltungen aktiv und gewinnbringend für das Programmierenlernen gestaltet werden?
- Wie können Studierende zur aktiven Mitarbeit und kooperativen Verhalten motiviert werden?
- Welchen Einfluss haben Interaktivität und Perfektionismus bei der Gestaltung von digitalen Lernmaterialien zur Programmierung?

1.2 Aufbau der Arbeit

Die vorliegende Arbeit dient der Einbettung der veröffentlichten Beiträge im Rahmen einer kumulativen Dissertation.

Aufgrund des bei Publikationen begrenzten Umfangs werden insbesondere dort nicht berücksichtigte Aspekte erläutert. Zudem wird den Beiträgen ein Rahmen gegeben, der sowohl theoretische Grundüberlegungen als auch Implikationen und die Verbindung der Beiträge umfasst.

Zunächst wird im Kapitel 2 die relevante Literatur erläutert, die die Basis für die folgenden Kapitel bildet. Kapitel 3 erläutert die Methodik im Studienkontext und gibt eine Übersicht über die zugrundeliegenden Veröffentlichungen. Teil A (II) der Arbeit widmet sich dann den Voraussetzungen heterogener Gruppen an Hochschulen, wie diese insbesondere in der Informatik häufig vorliegen. Dabei werden die Erwartungen und Einstellungen zum ersten Informatikmodul von ausgewählten Studierenden betrachtet. Zudem beschäftigt sich eine eingereichte Publikation zur Frustrationstoleranz von Studierenden in ihrem ersten Informatikmodul mit einem weiteren Aspekt der Voraussetzungen in heterogenen Gruppen. Dabei stand im Fokus die Zusammenhänge zwischen Informatiklernen und Frustrationstoleranz sowie zwischen dem anekdotischen Narrativ und dem psychologischen Konzept zu erforschen und tatsächliche mögliche Einflussfaktoren herauszustellen.

Teil B (III) bildet den stärker praktischen Teil der Arbeit ab. Zentral ist hierbei das erstellte Modulkonzept, das die in Kapitel 2 beschriebenen Konzepte des Flipped Classroom, der Projektarbeit sowie weiterer relevanter Forschung umsetzt. Neben einer Publikation zum grundsätzlichen Modulkonzept und dem ebenfalls publizierten Gamificationkonzept werden hier weitere Evaluationsergebnisse aus einem folgenden Veranstaltungsdurchgang vorgestellt. Den Abschluss bildet die Einbettung einer Studie zu Lernmaterialien, bei der verschiedene Arten von Videos und interaktiven Materialien im Hinblick auf die nötige Perfektion und den Unterschied zwischen Worked Examples und Live Coding verglichen wurden.

In Teil IV werden die vorliegenden Ergebnisse insgesamt noch einmal diskutiert und Folgerungen für die konkrete Situation sowie auch generell für die universitäre und schulische Anfangslehre in der Informatik gezogen. Kapitel 10 bildet mit einer Zusammenfassung über die Inhalte den Abschluss dieser Arbeit.

2 Literatur

In diesem Kapitel wird ein Überblick über die für die Dissertation relevante Forschung gegeben. Dabei werden zunächst Metastudien im Bereich der universitären Einführungslehre in der Informatik erläutert und im Folgenden die Forschung zu den konkreten Themenschwerpunkten der Arbeit dargelegt.

2.1 Forschung zu Einführungsmodulen in der Informatik

Die Studieneingangsphase, in der sich viele Studierende der Einführungsmodule der Informatik befinden, nimmt eine besondere Rolle innerhalb des Studiums ein. Die Studierenden müssen nicht nur fachliche Kompetenzen erwerben, sondern sich auch in der Universität einfinden, den Rollenwechsel zwischen Schule und Studium begreifen, sich mit ihrer Fachwahl auseinandersetzen sowie Arbeitstechniken einüben [Web10, BT14]. Förderung sei aufgrund der Heterogenität in dieser Phase wesentlich wichtiger als Bestrafung beim Verfehlen von Zielen [Web10]. In der Mathematik, die als besonders schwierig für Studienanfänger gilt, existieren auch im deutschen Sprachraum bereits Ansätze, um explizit den Übergang zwischen Schule und Studium zu erleichtern [BHF11]. Dennoch weisen insgesamt mathematikbezogene Fächer wie auch die Informatik vor allem aufgrund der Leistungsanforderungen überdurchschnittliche Studienabbruchquoten auf [HHS⁺10].

Seit etwa 50 Jahren ist auch die Einführungslehre in der Tertiärbildung der Informatik ein wiederkehrendes Thema in der internationalen informatikdidaktischen Forschung [BQ19]. Becker und Quille [BQ19] kategorisierten in einer 2019 veröffentlichten Metastudie 481 Veröffentlichungen der SIGSCE, welche als größte Informatikdidaktik-Konferenz betrachtet werden kann, zur universitären Einführungslehre der Informatik. Als acht Hauptkategorien erwiesen sich die Themen *Lehren* (105 Paper), *Lernen und Bewertung* (81 Paper), *Studierende* (78 Paper), *CS1 Design, Struktur und [Lehr-]Ansatz* (60 Paper), *Erste Sprachen und Paradigmen* (45 Paper), *Tools, CS1 Inhalte* (jeweils 38 Paper) sowie *Kollaborative Ansätze* (36 Paper) [BQ19, , übersetzt]. Bezüglich der Entwicklung über die Zeit ist vor allem eine Zunahme an Forschung zu Studierenden erkennbar. Während diese in den 1970er Jahren bei 16 % der insgesamt für diesen Zeitraum betrachteten Paper lag und in den 1990er Jahren keine Paper dazu bei der SIGSCE publiziert wurden, stellten Veröffentlichungen mit diesem Thema im zweiten Jahrzehnt des 21. Jahrhunderts mit 25 % einen bedeutenden Anteil. Zu dieser Oberkategorie gehören auch Aspekte zu *Vorkenntnissen* (11 Paper) und *Studierende anderer Hauptfächer als Informatik* (13 Paper). Die Autoren stellen dabei fest, dass es zwar auch ältere Veröffentlichungen z. B. zu Studierenden anderer Hauptfächer gibt, ein größerer Fokus darauf allerdings erst in den letzten Jahren erkennbar ist.

Eine ähnliche Entwicklung ist auch bei der Kategorie *Lernen und Bewertung* erkennbar. In der Kategorie *Lehren* wurde ab 1980 eine leichte Abnahme des Anteils an den ausgewerteten Veröffentlichungen festgestellt, jedoch sind einzelne Unterkategorien wie z. B. *Flipped approaches* [BQ19, S. 342] erst in den letzten Jahren überhaupt erforscht worden. Die Forschung zu kollaborativen Ansätzen begann erst in den 1990er Jahren und wuchs bis auf etwa 10 % der betrachteten Paper ab 2010.

Eine ältere Metastudie von Robins et al. aus dem Jahr 2003 [RRR03] klassifiziert die damals vorliegende informatikdidaktische Forschung vor allem danach, ob sich die Studien mit Programmieranfängern und damit der Lernpsychologie beim Programmieren oder mit fortgeschritteneren Programmierenden und deren konkreten technischen Strategien wie z. B. Testen oder Debugging, beschäftigt. Nach Davies [Dav93] ist entsprechend der Klassifizierung in Lerntaxonomien [BEF⁺73] auch zwischen dem Wissen über bestimmte Konzepte und der sinnvollen Anwendung dieser zu unterscheiden. Programmieranfänger würden dabei vor allem über Kompetenzen auf der als niedriger anzusehenden Wissens Ebene verfügen oder zwar die kognitiven Fähigkeiten für die Anwendung besitzen, diese aber aufgrund mangelnder Programmiererfahrung nicht einsetzen können [Dav93]. Insbesondere Lehrbücher zu Programmier-Einführungen würden nach Robins et al. [RRR03] vor allem Wissen ohne ausreichende Anwendung vermitteln. Dies wurde in früheren Studien auch als *fragiles Wissen* [PM86] bezeichnet. Konkrete Übungsaufgaben würden dagegen - unter Anwendung gelernter Strategien - vor allem in Kombination mit Kollaboration und grafischer Darstellung zu Motivation und größerem Erfolg führen [RRR03]. Nach Winslow [Win96] würde es jedoch etwa zehn Jahre dauern, bis ein Novize zum Experten im Programmieren wird. Somit erscheint es nicht zielführend einen umfassenden Wissensaufbau in diversen Programmierparadigmen in einem Einführungskurs anzustreben.

Auch in der allgemeinen Hochschuldidaktik liegt der Fokus bereits seit einiger Zeit auf der Studierendenzentrierung, bei der das Lernen im Gegensatz zum Lehren als wichtigster Prozess betrachtet wird [Ram03]. Hochschullehre müssen Studierende dazu befähigen ihr Lernen zu kontrollieren. Dies werde allerdings nicht durch einen passiven und kompetativen Lehranspruch, sondern durch Kooperation und tiefgehende Aktivitäten erreicht [Ram03, S. 112, 147].

Eine Arbeitsgruppe um Luxton-Reilly et al. [LRSA⁺18] hat in einem umfassenden Literature Review die Entwicklungen zur Einführungslehre in der Informatik zwischen 2003 und 2017 untersucht. Dabei wurden 2189 Paper diverser Datenbanken betrachtet, von denen 1666 tatsächlich inhaltlich analysiert und klassifiziert wurden. Als vier Hauptkategorien wurden dabei *Studierende* (489 Paper), *Lehren* (905 Paper), *Curriculum* (258 Paper) sowie *Bewertung* (192 Paper) [LRSA⁺18, , übersetzt] festgestellt, wobei auch die Einordnung in mehrere Kategorien erlaubt war. Beim zeitlichen Aspekt konnten die Autoren feststellen, dass eine grundsätzliche Zunahme an Veröffentlichungen zum Thema der anfänglichen Programmierlehre vorliegt. Durch weitere Analyse der Paper als Anteil aller bei den ACM SIGSCE Konferenzen veröffentlichten Paper folgerten sie allerdings, dass der Anstieg vor allem außerhalb dieser stattgefunden hat. Dies sei entweder durch die grundsätzlich beschränkte Menge der akzeptierten Beiträge oder durch stärkere Öffnung der Einreichenden auch für andere Publikationsstellen zu begründen. Auch gäbe

es Bereiche, in denen die Forschung relativ konstant bleibe, z. B. Theorien oder das Curriculum [LRSA⁺18].

In der Kategorie der *Studierenden* wurden in der Unterkategorie *Inhalt* Beiträge zu theoretischen Modellen des Verständnisses, zu Code-bezogenen Kompetenzen, zum Studierendenverhalten und zu Studierendenfähigkeiten gefunden. In der Unterkategorie *Gefühl* wurden Beiträge zu Studierendeneinstellungen, Studierendenengagement sowie Erfahrungen in der Programmierung ausgewertet und als *Subgruppen* wurden Studierende mit erhöhtem Risiko des Nichtbestehens sowie unterrepräsentierte Gruppen wie z. B. Frauen oder Minderheiten aufgeführt. Die theoretische Basis sei dabei vor allem qualitativer Natur und bisher kaum empirisch belastbar erforscht. Bezüglich der Code-bezogenen Kompetenzen beschäftigen sich die meisten Studien mit Zusammenhängen zwischen diesen. Code angemessen lesen zu können sei dabei eine Voraussetzung für erfolgreiches Problemlösen durch Programmierung. Beim Schreiben von Code seien außerdem Aufgabenschwierigkeiten von Bedeutung, sodass ein Teil der Forschung auch konkrete Aufgaben thematisiert. Im Kontext des Verbesserns von Fehlern werden häufig auch Werkzeuge zum Erkennen dieser, darunter auch Entwicklungsumgebungen, erforscht. Diese seien auch bei der Erforschung von Studierendenverhalten als Interaktion und mit Debugging als Tätigkeit relevant. Beim Studierendenverhalten wird zunehmend mit Learning Analytics gearbeitet und dadurch eine große Menge an Daten analysiert. Auch automatisierte Korrektur führt zu einer schnelleren Generierung von Daten, die zum besseren Verständnis eines Lernprozesses hilfreich sein können. Auch Interaktionen mit Lern-Management-Systemen und Lernmaterialien diverser Arten wie z. B. Video-Vorlesungen werden in die Kategorie des *Studierendenverhaltens* eingeordnet. Bei den Einstellungen der Studierenden gegenüber Einführungskursen der Informatik beschäftigen sich mit 105 Papern viele Forschende mit der Herausforderung diese positiv zu beeinflussen. Ein weiteres bedeutendes Thema in dieser Kategorie ist die Selbsteffizienz, diese hat auch erheblichen Einfluss auf das Studium unterrepräsentierter Gruppen. Unter den Maßnahmen zur besseren Motivation der Studierenden finden sich diverse Ansätze in Kurskonzepten, u. a. auch Flipped Classroom und Gamification. Aus Studien zu Studierenden mit einem hohen Risiko des Nichtbestehens kann vor allem gefolgert werden, dass der Erfolg zu Beginn eines Einführungskurses von großer Bedeutung für den weiteren Verlauf ist. [LRSA⁺18]

In der Oberkategorie *Lehren* wurden Paper zu den Themen Theorien, Kursstruktur, Lehrtechniken, Werkzeuge und Infrastruktur betrachtet. Nur wenige Veröffentlichungen beschäftigen sich konkret mit Lerntheorien, die meisten davon betrachten dabei die Erkenntnisse aus allgemeinen bildungswissenschaftlichen Theorien. Unter den vor allem in Fallstudien erforschten Lehransätzen finden sich u. a. problembasiertes Lernen, Flipped Classroom sowie kognitive Lehre, wobei diese auch miteinander kombiniert werden können. Unter den 44 Papern, die sich mit der Kontext der Lehre im Sinne des Anwendungsbezugs beschäftigen, besteht eine Häufung im Bereich der Spiele sowie der Bildverarbeitung. Diese wirken überwiegend motivierend und positiv auf die Studierenden, bilden an sich jedoch noch keine gute Lehre. Kollaborative Ansätze wie z. B. Pair Programming, bei dem in der Regel zwei Personen bei einer Programmierung zusammenarbeiten und sich in ihren Rollen ergänzen, führen dagegen vor allem bei schwächeren Studierenden zu belastbaren Verbesserungen. In der aktuellen Diskussion seien auch Fragen zur Reihenfolge der vorgesehenen Themen, zum systematischen Testen

von Code sowie zu Aufgaben und dem Vorgehen beim Lösen dieser. Einige Studien widmen sich außerdem dem Einsatz von Medien und explizit auch Videos, den Effekten von Analogien auf Basis des Vorwissens der Studierenden und objekt-orientierter Programmierung als übergeordnetes Konzept. Unter den Beiträgen zu Tools für den Einsatz in der Lehre finden sich Anwendungen verschiedenster Ebenen. Diese reichen von Programmierumgebungen bis zu Werkzeugen für konkrete kleine Anwendungen zum Programmierenlernen, für konkrete programmierbezogene oder allgemeine Tätigkeiten wie z. B. Problemlösen, Visualisierung oder Kollaboration sowie auch Werkzeuge für die Metaebene der Erforschung von Lernprozessen. [LRSA⁺18]

Im Bereich des *Curriculums* stehen neben grundsätzlichen Fragen zu Inhalten von Einführungsmodulen die Programmiersprachen und -paradigmen im Vordergrund. Dabei existieren zwar einige Forschungsarbeiten zum Vergleich von zwei oder mehr Programmiersprachen, allerdings besteht keine Einigkeit über die beste Programmiersprache für den Beginn der Lehre. Lediglich bei den Programmierparadigmen bevorzugen die meisten Lehrenden objektorientierte Programmierung. Ein weiterer bedeutender Teil der Forschung beschäftigt sich mit der Prüfung und Bewertung von erworbenen Kompetenzen. Dabei stehen neben Prüfungskonzepten, dem Bezug zu Lerntaxonomien und der Gestaltung von Prüfungsaufgaben auch Feedback und hierfür hilfreiche Tools, aber auch Strategien zur Vermeidung von Täuschungen durch Studierende im Fokus. [LRSA⁺18]

Zwischen 2005 und 2009 nahmen Nikula et al. [NGK11] die Umgestaltung eines Einführungsmoduls der Informatik mit etwa 150 bis 250 Studierenden pro Semester vor. Dabei stellten sie fest, dass neben der grundsätzlichen fachlichen Schwierigkeit auch die Struktur des Kurses und Motivationsprobleme zum Abbruch vieler Studierender beitragen würden. Im ersten Jahr wurden dabei neben inhaltlich-gestalterischen Änderungen in kleinem Umfang ein Projekt eingeführt und ermöglicht einen Teil der Kursnote bereits über die Aufgaben zu verbessern. Im zweiten Jahr erfolgte ein Programmiersprachenwechsel von C zu Python und die Aufgaben wurden verpflichtend sowie entsprechend strukturierter. Erst zu diesem Zeitpunkt wurde ein zentraler Abgabetermin für diese eingeführt. Außerdem wurden Gelegenheiten zur aktiven Übung eingebaut. Die bisher im Umfang der Abgaben getrennten Gruppen Hauptfach- und Nebenfachstudierender wurden im dritten Jahr vollständig zusammengelegt. Darüberhinaus ergänzten Beispiellösungen für die Aufgaben sowie die Möglichkeit für Fehlerberichte weitere Punkte zu sammeln den Kurs, der nun auch eine Turtle als Zeichenwerkzeug für ein neues Projekt nutzte. Im vierten Jahr wurden die bisherigen Konzepte um eine Aufzeichnung der Vorlesungen und kleine Änderungen an den durch die Aufgaben erreichten Punkten, die nun auch einsehbar waren, ergänzt. Auch im fünften Jahr folgten weitere Anpassungen der Punkte. Außerdem wurden die verpflichtenden Aufgaben um Tracing erweitert. Während dieser Umstellung erfolgten zudem in einigen Jahren Versuche mit Laborübungen und der Umstellung auf eine neuere Python-Version, mit deren Ergebnissen die Autoren jedoch nicht zufrieden waren. Die Autoren sprechen sich auch grundsätzlich für eine Analyse der genauen Bedingungen und Studierenden aus, wobei auch andere Autoren die Wichtigkeit dessen betonen [Ram03, S. 99]. Trotz einiger nicht erfolgreicher Versuche, führte das veränderte Kurskonzept [NGK11] insgesamt sowohl zu höheren Bestehensquoten als auch zu höherer Zufriedenheit der Studierenden in den Evaluationen. Die Autoren empfehlen die Motivation dadurch zu verbessern, dass zunächst demotivierende Faktoren

entfernt werden, im zweiten Schritt die intrinsische Motivation und im dritten Schritt die extrinsische Motivation gefördert werden.

Zur Erforschung des Erfolgs eines Lernkonzepts bestehen grundsätzlich verschiedene Möglichkeiten mit unterschiedlich hohem Aufwand. Einige Studien berücksichtigen die Modulabschlussnoten als Indikator für Lernerfolg. Andere Studien ziehen die zentralen Ergebnisse aus Befragungen der Studierenden. Eine weitere, jedoch sehr aufwendige, Methode wären Kompetenzmessungen. Hier besteht allerdings noch Forschungsbedarf dazu wie diese konkret durchgeführt werden sollten und welche Kompetenzen dabei relevant sind [LRSA⁺18]. Aus der Forschung ist jedoch bekannt, dass die möglichen Indikatoren häufig zusammenhängen. So korreliert beispielsweise die Selbsteinschätzung zu guter Lehre mit 0,47 positiv zum akademischen Erfolg des Studierenden [WLR97]. Dadurch können Fragebögen oft einen umfassenderen Eindruck geben und genauer auf konkrete Aspekte eingehen als teilweise an die Lerngruppe angepasste Bewertungen. Insgesamt sollten nach Möglichkeit jedoch möglichst umfassende Daten ausgewertet werden.

Die informatikdidaktische Forschung beschäftigt sich damit seit einiger Zeit bereits auch explizit mit Einführungsmodulen, es liegen aber nur wenige Ergebnisse zu Gesamtkonzepten und Umgestaltungen vor. In der Entwicklung ist eine Fokussierung auf die Studierenden und deren Lernprozess erkennbar. Ein konkreteres Konzept, das aktives Lernen erfordert, wird mit dem Flipped Classroom im Folgenden betrachtet.

2.1.1 Forschung zum Flipped Classroom

Eine Herausforderung für die Hochschullehre stellt die Heterogenität der Studierenden dar [Sei14, KSWB13, S. 38]. Dabei kann Heterogenität zwar auch rein deskriptiv oder normativ - als Beschreibung oder Maßstab - verstanden werden, bedeutet jedoch auch eine Herausforderung an die organisatorische und didaktische Gestaltung der Lehre [Wal14, S. 25f.]. Für heterogene Lerngruppen geeignete Konzepte basieren daher auf Flexibilität, unterschiedlichen Aufgabenmengen und -schwierigkeiten sowie Materialien, die selbstständiges Lernen fördern [Wal14, S. 46].

Wie in 2.1 beschrieben, findet sich unter den Lehransätzen, die auch in der hochschuldidaktischen Forschung zur Informatik seit kurzer Zeit zunehmend Relevanz finden und selbstständiges Lernen ermöglichen, das Konzept des *Flipped* oder *Inverted Classrooms*. Dieses wurde in der Hochschullehre erstmals im Jahr 2000 von Lage, Platt und Treglia [LPT00] umgesetzt. Diese beschreiben das Konzept dadurch, dass Aktivitäten, die üblicherweise in der Präsenzzeit stattfinden, außerhalb der Präsenz stattfinden und umgekehrt. Für ihren Wirtschaftskurs nutzten sie daher Vorlesungsfolien mit einer Tonspur sowie weitere Materialien wie z. B. Texte für die Selbstlernphase der Studierenden. In der Präsenzphase klärten die Lehrenden zunächst offene Fragen, bevor interaktiv konkrete wirtschaftliche Experimente demonstriert und von den Studierenden auch untereinander diskutiert wurden.

Einige Jahre später wandten Bergmann und Sams [BS12] dieses Konzept auf ihren Schulunterricht an und entwickelten es erheblich weiter. Dabei sammelten sie ihre Erfahrungen und beschrieben wie ein Flipped Classroom durchgeführt werden sollte. Sie

hatten zuvor festgestellt, dass ihre Lernenden, die z. B. aus gesundheitlichen Gründen den Unterricht verpasst hatten, von Selbstlernvideos profitiert haben.

Bishop und Verleger [BV13] definieren den Flipped Classroom auf Basis einer systematischen Literaturrecherche angelehnt an die Definition nach Lage et al. [LPT00], jedoch mit der Erweiterung, dass die konkrete Umsetzung über den reinen Tausch der Aktivitäten in Präsenz- und Selbstlernphasen hinausgehen muss. Andernfalls sind diese nicht ausreichend an die Phasen angepasst, da eine in Präsenz gehaltene Vorlesung nicht ohne weiteres zum Selbstlernen genutzt werden kann. Sie unterscheiden daher auch vollständig und partiell umgesetzte Flipped Classrooms. Beim vollständigen Flipped Classroom sind keine Vorlesungsanteile mehr in der Präsenzphase vorhanden. In diesem Fall werden in der Präsenz Elemente aktiven Lernens genutzt, zu denen problembasiertes Lernen, Kollaboration und Kooperation sowie Peer-Learning gehören. Für die Selbstlernzeit müssen die Lerninhalte entsprechend aufgearbeitet und in möglichst motivierender Form präsentiert werden. Kurze Videos würden sich hierbei besser eignen als Texte [BV13].

Insgesamt bietet dieser Lehransatz Möglichkeiten für Differenzierung. Ein besonderer Vorteil liegt hierbei darin, dass er erlaubt zeitlich individueller zu lernen. Somit unterstützt dieser Ansatz Lernende, die Schwierigkeiten mit den Lerninhalten haben, da die Materialien im Selbststudium in eigenem Tempo und beliebig oft bearbeitet werden können [BS12, S. 22f., S. 28]. Das Konzept eignet sich aufgrund des selbstständig regulierbaren Lerntempos in der Inhaltsvermittlung besonders für Lernende mit unterschiedlichen Vorkenntnissen [AD14, Mor19]. Bei regelmäßiger Teilnahme an den Kontaktterminen der Präsenzphasen steigen sowohl die Interaktionen zwischen Lehrenden und Lernenden als auch zwischen den einzelnen Lernenden im Vergleich zu klassischer Lehre [BS12, S. 27]. Zu den Vorteilen interaktiver Lehre im Vergleich zu klassischer Lehre existieren bereits seit längerer Zeit insbesondere für Inhalte, in denen das Lösen von Problemen zentral ist, Forschungsergebnisse zu deren Erfolg [Hak98].

Für das Gelingen des Lernprozesses im Flipped Classroom-Modell müssten sich jedoch auch Lernende darauf einlassen und ihre Lerngewohnheiten anpassen [BS12, S. 18]. Die ständige Verfügbarkeit von Lernmaterialien erhöhe dabei die Gefahr des Aufschiebens. Allerdings werden durch erfolgreiche Anpassungen auch überfachliche Kompetenzen wie die Fähigkeit zum Selbstlernen, Kommunizieren und Kooperieren gestärkt [Mor19]. Eine klare Kommunikation der Herausforderungen, aber auch Erwartungen an die Studierenden, kann den Prozess der Anpassung unterstützen [Mor19]. Um das selbstregulierte Lernen zu stärken, sollten außerdem klare Ziele kommuniziert, die aktive Übung gefördert und Feedback gegeben werden [FR16]. Grundsätzlich würden sich Selbstlernkompetenzen allerdings vor allem dann entwickeln, wenn diese auch gefordert werden [KSWB13, S. 39]. Eine weitere Belastung durch das Konzept liegt im Aufwand der Materialproduktion durch die Lehrenden, der teilweise zusätzlich die Einarbeitung in entsprechend geeignete Technologien erfordert [SCZ15]. Abhängig vom tatsächlichen Vorbereitungsaufwand und der zur Verfügung stehenden Zeit können allerdings auch extern produzierte Materialien als Grundlage verwendet werden [Mor19].

Zur konkreten Umsetzung des Konzepts existiert eine Vielzahl von Empfehlungen. Köppe et al. [KP14, KNH⁺15a, KNBH16] erläutern, dass es für die Präsenzphasen des Flipped Classrooms einer guten Planung bedarf. Dabei sollten die Lösungen der Studierenden im Vordergrund stehen und durch qualitatives Feedback auch auf individueller Basis

bestätigt werden. Die Lernenden sollten außerdem Gelegenheit erhalten weiterführende Aufgaben bearbeiten zu können und gleichzeitig bemerken, dass die Lehrenden ihren Fortschritt erkennen [KNBH16]. Für den Einstieg in eine Umgestaltung empfehlen einige Lehrende auch aufgrund von Rückmeldungen der Studierenden nicht direkt einen vollständigen Flipped Classroom anzustreben. Durch die Ergänzung des bisherigen Konzepts um digitale Anteile und zusätzliche Materialien kann dabei ein Blended Learning als Kombination von klassischer Lehre und digitalen Angeboten zum Selbstlernen eine erste Stufe auf dem Weg zu einem Flipped Classroom-Konzept darstellen [Mor19, ZLM⁺09]. Allerdings sollte dennoch eine didaktisch überlegte Einbettung der Materialien erfolgen und nicht erwartet werden, dass Studierende ausschließlich mit den zur Verfügung stehenden Materialien auskommen [EMV14].

Die Gestaltung der Materialien für die Selbstlernphasen ist wiederum ein zentraler Bestandteil des Flipped Classroom-Konzepts. Wie bereits erwähnt, werden multimediale Materialien grundsätzlich als bessere Basis verstanden als reine Texte [BV13]. Allerdings sind auch hier verschiedene Umsetzungen möglich. Erste Studien zum Vergleich von rein erklärenden Videos und Aufgaben, die die Studierenden dazu anregen sollten sich zunächst selbst mit dem Problem auseinanderzusetzen, deuten an, dass interaktive Materialien tendenziell ein tieferes Verständnis hervorrufen können, das höhere Kompetenzebenen bedient [WS14]. Innerhalb von grundsätzlich aktiven Lehrkonzepten scheinen auf inhaltsvermittelnde Materialien folgende Quizzes jedoch keine Auswirkungen auf die Prüfungsergebnisse zu haben [LL15].

Weiter sollten Lehrende nicht davon ausgehen, dass Studierende zur Verfügung gestellte Materialien vollständig und wie zeitlich vorgesehen bearbeiten [DSFD16]. Eine in einem Einführungsmodul der Informatik durchgeführte Studie gibt allerdings Hinweise darauf, dass kurze anschließende Aufgaben und die klaren zeitlichen Anweisungen zur Bearbeitung des Materials die Quote der Studierenden, die sich die Materialien zumindest anschauen, erhöhen könnten [DSFD16].

Um das aktive Lernen zu fördern, kann auch eine besondere Raumeinrichtung hilfreich sein. Hakimzadeh et al. [HAB11] haben von den Lernenden positives Feedback für ihre Umsetzung von *SCALE-UP* [BS03] erhalten. Dabei war der Lernraum mit Gruppentischen entlang der Wände sowie für alle Gruppenteilnehmenden sichtbaren Bildschirmen ausgestattet. Einige Computerarbeitsplätze sowie zwei Großbildschirme und ein Arbeitsplatz für die Lehrperson gehörten ebenfalls zur Einrichtung und sollen insgesamt die aktive Zusammenarbeit auch für digitale Aufgaben fördern. Kohls et al. [KHGP20] bemerken, dass sich insbesondere durch eigene Endgeräte (*Bring your own device*) und Flipped Classroom auch die Lernräume verändern und angepasst werden müssen. Durch die förderliche Gestaltung von Lernräumen kann auch kollaboratives Arbeiten begünstigt werden. Benyon und Mival [BM15] sprechen dabei von *Blended Spaces*, bei denen digitale Räume in physische integriert werden und so eine neue Erfahrung ermöglichen. Auch in diesem Beitrag wird betont, dass die Einrichtung eines Lernraums für die Zusammenarbeit zwischen Lernenden von Bedeutung ist. Auch im deutschsprachigen Raum haben bereits einige Hochschulen spezielle Lernräume mit digitalen Geräten eingerichtet, die kollaborative Arbeit ermöglichen [KHGP20].

Auch in Informatikeinführungsmodulen wurden bereits Erfahrungen mit unterschiedlichen Umsetzungen des Flipped Classroom Konzepts gesammelt. Insbesondere im eng-

lischsprachigem Raum werden solche Module häufig als *Computer Science 1*, abgekürzt *CS1*, bezeichnet.

Horton, Campbell et al. [HCC⁺14, CHCG14] nutzten Videos für die Selbstlernphase in ihrem CS1-Kurs mit der Programmiersprache Python und einem Objects-First-Ansatz. Im Vergleich zur bisherigen Lehre konnten sie dabei eine Verbesserung der Noten der Studierenden im Flipped Classroom-Konzept feststellen. Auch bei der Befragung der Studierenden bevorzugten diese die Lehre im Flipped Classroom. Lediglich geringere Anwesenheitsquoten in den freiwilligen Präsenzveranstaltungen konnten festgestellt werden. Diese werden jedoch dadurch erklärt, dass bei gutem Verständnis der Materialien keine zwingende Notwendigkeit von weiterer Übung bestehe. Ähnliche Ergebnisse erzielten auch Lockwood und Esselstein [LE13] sowie Sarawagi [Sar14].

Lasserre und Szostak [LS11] stellten Erfolge mit einem Flipped Classroom und teambasierter Aufgabenbearbeitung in ihrem CS1-Kurs fest. Die Studierenden erreichten dabei in der Abschlussklausur bessere Leistungen und insgesamt bestanden mehr Studierenden den Kurs. Allerdings stellten sie fest, dass der Wechsel im folgenden Kurs mit eher klassischer Lehre zu geringerer Aktivität führt und der Anteil Studierender, die diesen Kurs erfolgreich abschließen, sinkt. Sie führen dies zum Teil darauf zurück, dass Studierende im CS1-Kurs motiviert wurden einen weiteren, für ihren Studiengang möglicherweise nicht verpflichtenden, Informatikkurs zu belegen, diesen in Folge der abweichenden Lehrmethodik jedoch abbrechen. Eine Verbesserung der Klausurergebnisse in einem ebenfalls teambasierten Flipped Classroom konnten aber auch Elnagar und Ali [EA12] in mehreren Durchführungen erreichen.

Ebenfalls gute Erfolge in einem Flipped Classroom in einem Einführungsmodul der Informatik stellten Elmaleh und Shankararaman [ES17] fest. Als Material zum Selbstlernen verwendeten sie dabei Videos mit einer Länge von zehn bis 15 Minuten, die von Selbsttests begleitet wurden. In der Präsenzphase beschäftigten sich die sieben Gruppen mit jeweils etwa 40 Studierenden mit der Auswertung der Selbsttests sowie, begleitet von mehreren Lehrpersonen, zunehmend schwieriger werdenden Aufgaben. Zum Abschluss erklärte eine Lehrperson häufig beobachtete Probleme und mögliche Lösungsansätze für diese. Auf dieser Basis mussten die Studierenden weitere Aufgaben bearbeiten und abgeben, wobei diese persönlich von Tutoren korrigiert wurden. Obwohl die Studierenden insgesamt mehr Zeit aufwenden mussten, benötigten sie für die Bearbeitung der Abgabenaufgaben weniger Zeit und erzielten bessere Ergebnisse in der Abschlussprüfung.

Auch für nur teilweise geflippte Lehrveranstaltungen in Einführungsmodulen der Informatik sind Forschungsergebnisse aus Fallstudien verfügbar. Insgesamt liegen auch hier im Bezug auf die Zufriedenheit der Studierenden positive Ergebnisse vor [RR19, Hod19, Moh20]. Generell werden aktive Lernformen positiv bewertet und Studierende bevorzugen computerbasierte Übungen im Gegensatz zu klassischen, bei denen kein Code selbst geschrieben und ausprobiert werden kann [TDH16, Lar13].

In einer Weiterentwicklung des Konzepts aus [HCC⁺14] und [CHCG14] wurden auch Präsenz- und Onlineangebote miteinander verglichen [CHC16]. In der Präsenzlehre erfolgte signifikant mehr Interaktion mit den Lehrenden und Mitstudierenden. Außerdem wendeten die Studierenden in Präsenz mehr Zeit für die Veranstaltung auf. Auch aufgrund

höherer Abbruchquoten im Onlineangebot wurde die Präsenzlehre von den Autoren insgesamt als erfolgreicher bewertet.

Neben Einführungsmodulen entwickeln auch einige Lehrende Konzepte für weiter fortgeschrittenere Informatikmodule, die auf aktiven Lernsettings aufbauen. Pirker et al. [PRSG14] haben für einen Kurs zu Suchalgorithmen verschiedene Konzepte aktiven Lernens analysiert und einen Kurs aufgebaut, der neben einem Vorlesungsanteil von etwa 65 % auch interaktive Übungen, Quizzes und andere Aktivitäten enthält. Diese sollten je nach Art alleine oder in Gruppen von zwei oder vier Studierenden gelöst werden. Neben direktem Feedback für Quizfragen wurden Aufgaben ausführlicher diskutiert. Als Gamificationansatz (siehe auch 2.1.3) wurden Auszeichnungen für Erfolge vergeben. Eine positive Fehlerkultur und die Anpassung an das Lerntempo der Teilnehmenden ergänzten das als *Motivational Active Learning* bezeichnete Konzept. Die Studierenden empfanden die Interaktivität grundsätzlich als vorteilhaft, solange nicht zu viele Aufgaben gelöst werden mussten. Die Arbeit an den Lösungen bevorzugten sie in Paaren, sodass auch bei Aufgaben, die zu viert erledigt werden sollten, häufig zunächst Paare zusammen gearbeitet und sich danach ausgetauscht haben.

Morisse [Mor16, Mor19] hat zwei Module der Informatik, Audio-/Videotechnik sowie theoretische Informatik, schrittweise in einen Flipped Classroom überführt. Dabei wurden zunächst Vorlesungsaufzeichnungen angeboten, die im weiteren Verlauf in kürzere Einheiten unterteilt und um zusätzliches Material wie ein Skript und freiwillig zu bearbeitenden Aufgaben ergänzt wurden. Die zuvor klassische Vorlesung wurde mit der Zeit in ein Live Coaching umgewandelt. Diese bestanden aus einer Reflektion zu Beginn, in der ein Quiz oder eine andere Aufgabe bearbeitet werden sollte. Darauf folgte Zeit für Fragen. Den Hauptteil stellte eine aktive Phase dar, in der die Studierenden in kleinen Gruppen an den Aufgaben arbeiteten. Zum Schluss gab in den ersten Durchführungen eine Diskussion einiger Ergebnisse [Mor16], im weiteren Konzept gab die Lehrperson einen Ausblick auf das folgende Thema [Mor19].

Im Kontext von Ingenieurwissenschaften veröffentlichten Karabulut et al. [KJCJ17] eine Metastudie, die 62 Studien zum Flipped Classroom in diesem Fachgebiet betrachtet, die zwischen 2000 und 2015 publiziert wurden, wobei erst ab 2012 eine relevante Anzahl an Veröffentlichungen gefunden wurde. In der Hälfte der 30 analysierten Publikationen mit Ergebnissen eines konkreten Vergleichs waren die Flipped Classroom-Konzepte dabei trotz unterschiedlichen Evaluationsmethodiken klar erfolgreicher als bisherige klassische Lehransätze. Lediglich drei Publikationen hatten auch teilweise negative Ergebnisse. Die Autoren betonen, dass die Ansätze zur Gestaltung der Bestandteile innerhalb der Flipped Classrooms häufig noch wenig theoriebasiert sind. Zudem sei neben weiterer quantitativer Forschung zum Erfolg auch qualitative Forschung zur Gestaltung notwendig. Auch für das Fach Mathematik existiert eine ähnliche Metaanalyse. Lo et al. [LHC17] betrachteten 61 Veröffentlichungen zwischen 2012 und 2016, die eine vollständig Flipped Classroom-Umsetzung beschrieben und dabei statistische Daten enthielten. Aus 21 der Studien konnten die statistischen Daten für eine übergreifende Berechnung des Effekts auf die Lernergebnisse verwendet werden. Ein Großteil der Studien wurde dabei in den USA und im Kontext der ersten Studienjahre durchgeführt. Die statistische Analyse der 21 Veröffentlichungen ergab dabei, dass die Flipped Classroom-Konzepte signifikant bessere Ergebnisse hatten als traditionelle Lehrkonzepte.

Obwohl für den Erfolg des Lehrkonzept insgesamt erste evidenzbasierte Ergebnisse vorliegen, basieren die Umsetzungen häufig auf unterschiedlichen Bestandteilen, bei denen andere Lehrkonzepte ebenfalls involviert werden. Eine zentrale Rolle nehmen die Gestaltung der Materialien für die Selbstlernphase sowie die Durchführung der Präsenzphasen ein. Bei geeigneter Umsetzung wirkt das Flipped Classroom-Konzept erfolgreich auf das Lernen und die Zufriedenheit der Studierenden. Das nächste Teilkapitel beschäftigt sich mit der Forschung zu den ebenfalls im Kontext des aktiven Lernens relevanten Aufgaben, für die während des Aufbaus des Konzepts ein Projektbezug als Rahmen gewählt wurde.

2.1.2 Forschung zu Projektarbeit

Projektorientiertes Lernen kann auf unterschiedliche Weisen definiert und umgesetzt werden und wird bereits seit längerem erforscht. Der Projektunterricht nach Dewey [DKS35] entstand dabei vor allem als Gegenbewegung zum Frontalunterricht. Mit einer sozialen Dimension durch zwischenmenschliche Interaktion und einem praktischen Ansatz führt dies zum problemorientierten Lernen, bei dem die theoretischen Inhalte auf konkrete Fragestellungen angewendet werden [Kol96, Kno97].

Nach Gudjons [Gud14] sind im Rahmen einer Handlungsorientierung individuelle Handlungsmöglichkeiten für konkrete Fragestellungen in Bezug auf Zeit und Ziele wesentliches Merkmal von Projektarbeit. Nach Kolmos [Kol96] existieren verschiedene Typen von Projektarbeit, die vor allem in den Vorgaben der Lehrperson variieren. Für den Studienbeginn seien dabei insbesondere *aufgabenbasierte* Projekte geeignet, bei denen die betreuende Person Thema und Methoden vorgeben können. Für offenere Projekte muss ein bestimmter Wissenstand bei den Lernenden vorausgesetzt werden können.

In der Einführungslehre in Informatik werden Projektarbeiten häufig in Zusammenhang mit grafischen Szenarien gebracht, da diese einen visuell einfach greifbaren und realitätsnahen Kontext darstellen. Einige Beispiele dafür zeigen Wicentowski und Newhall [WN05], Razak [Raz13], Matzko und Davis [MD06] sowie Bruce et al. [BDM01]. Dabei bearbeiten Studierende die projektartigen Aufgaben häufig in kleinen Gruppen [WN05] und visuelles Feedback konnte als Ursache für eine Verringerung von Programmierfehlern identifiziert werden [BDM01]. Aus Erfahrungen in mehreren CS1 und weiterführenden Kursen empfiehlt Kussmaul [Kus08] zu Beginn eines Kurses engere Anforderungen an die Lösung von Aufgaben zu stellen und diese zum Ende zu öffnen. Mitbestimmung durch Studierenden bei einigen Aspekten durch die Kombination von Projekten und unabhängigen Aufgaben führe so zu höherer Zufriedenheit. Kenn und Mammen [KM15] konnten durch ein Semesterprojekt mit zunehmend offener definierten Meilensteinen feststellen, dass sich die Dekompositionsfähigkeiten der Lernenden verbesserten. Dadurch gelang es den Studierenden besser Modellierungen für Programme vorzunehmen, indem komplexe Anforderungen in kleinere Bestandteile aufgeteilt wurden.

Basierend auf den Prinzipien von Clancy und Knuth [CK77] haben Sheth et al. [SMRS16] an mehreren Universitäten Kurse für Problemlösen durch Programmieren für Studierende ab dem dritten Studienjahr entwickelt und durchgeführt. Die grundlegenden Bestandteile waren dabei, dass offene, abwechslungsreiche und zielgerichtete Aufgaben

in kleinen Gruppen von Studierenden bearbeitet werden und die Lehrenden in einer Moderatoren-Rolle eine aktive Besprechung leiten. Ziel war es sowohl Kollaboration als auch einen angemessenen Wettbewerb anzuregen. In den konkreten Umsetzungen fanden in Anlehnung an die Konzepte des problembasierten Lernens und des Flipped Classrooms (siehe 2.1.1) aktive Veranstaltungen mit etwa 25 bis 30 Studierenden statt, bei denen in Kleingruppen mit drei bis vier Studierenden konkrete Probleme durch Programmierung kreativ gelöst werden sollten. Auf klassische Vorlesungen wurde dabei verzichtet. Um auch fachübergreifende Kompetenzen wie z. B. die Präsentationsfähigkeiten zu fördern, lag ein besonderer Fokus darauf, dass die Teilnehmenden sich gegenseitig Lösungsansätze vorstellen und Feedback dazu geben. Unter Nennung der Urheber war dabei auch erlaubt Code von anderen Gruppen in die eigene Lösung einzubauen. Den Autoren zufolge würden ähnliche Kurse häufig lediglich niedrige Abstraktionslevel von Lerntaxonomien erreichen und selten echte Anwendungen zeigen. In diesem Fall würden durch Reflexionen allerdings auch höhere Taxonomiestufen erreicht werden.

Das Feedback der Studierenden war überwiegend positiv. Obwohl der hohe Aufwand angemerkt wurde, empfanden die Teilnehmenden im Verhältnis dennoch einen hohen Lernerfolg. Insbesondere wurden die Offenheit der Aufgaben, die Größe der Veranstaltung und die Erlaubnis Codeteile anderer Gruppen zu übernehmen als positiv hervorgehoben. Einige Studierende sorgten sich um negativ wirkende Gruppenzusammensetzungen, die durch regelmäßige Veränderungen jedoch ausgeglichen werden könnten. Aus ihren Erfahrungen empfehlen die Autoren eine gezielte Aktivierung der Studierenden. Diese sollten sowohl dazu motiviert werden freiwillig zum Thema beizutragen, aber auch direkt angesprochen werden. Lehrende sollten auf die Ideen der Studierenden eingehen und dabei nicht zu früh eigene Beispiellösungen präsentieren. Bei der Bearbeitung von möglichst abwechslungsreichen Aufgaben sollte die gegenseitige Hilfe forciert werden, indem Fragen in die Runde zurück gegeben werden. Allerdings plädieren die Autoren auch für eine Anwesenheitspflicht und Vorgaben für die Aufmerksamkeit, indem Nebentätigkeiten unterbunden werden. Für Aufgaben empfehlen sie möglichst reale Probleme als Grundlage zu verwenden und betonen die Vorteile von Visualisierungen [SMRS16].

Um ein realistischeres Berufsbild schon in der Studieneingangsphase zu vermitteln, haben Vosseberg et al. [VCEV15] Projekte mit unternehmerischer Einbettung in Teams von sechs bis acht Studierenden, die von studentischen Tutoren betreut wurden, im ersten Semester von Informatik- und Wirtschaftsinformatikstudiengängen durchgeführt. Obwohl die Studierenden die offenen Aufgaben zunächst nicht gewohnt waren, schätzten sie insbesondere die frühzeitige Zusammenarbeit mit anderen Studierenden und empfanden den Studieneinstieg insgesamt als zufriedenstellend. Im Rahmen der Fortführung und Weiterentwicklung [RVEL19] wurden die Projekte jährlich angepasst, wobei mehrere Teilgebiete der Informatik involviert waren.

Stevenson und Wagner [SW06] sprechen sich dafür aus, dass Aufgaben in der Informatik insbesondere in Einführungsmodulen einen Realitätsbezug aufweisen sollten, dabei Kreativität fördern sowie herausfordernd und interessant sein sollten. Außerdem sollten Lernende Bibliotheken nutzen dürfen, um realitätsnah und über Basisfunktionen hinausgehend arbeiten zu können und auf verschiedenen Schwierigkeitsstufen und thematischen Inhalten eine Wahl haben. Um die Komplexität eines realistischen Problems abzuschätzen, sollten Lehrende mögliche Aufgaben zunächst selbst lösen. Sollte die Komplexität

die gewünschten Anforderungen dabei übersteigen, sei es durch Vorgeben eines Teils des Codes möglich diese zu verringern. Als Testdaten sollten jedoch zunächst konstruierte Daten genutzt werden, um Studierende nicht zu überfordern und Erfolgserlebnisse zu ermöglichen. Später könnten auch einfache reale Datensätze verwendet werden, damit die Lernenden mögliche entstehende Probleme und die Komplexität bei der Arbeit mit realen Daten erkennen können [SW06].

Ein weiterer Projektansatz in einem Einführungskurses zur Programmierung mit Python wird von Avouris et al. [AKK⁺10] beschrieben. Dabei wurde ein fünfwöchiges Projekt in Teams mit verschiedenen Themen zum Ende des Kurses realisiert, wobei dies mit einer Präsentation der Gruppen mit anschließender Preisverleihung für die beste Leistung abgeschlossen wurde. Eine Evaluation ergab eine höhere Studierendenzufriedenheit als im bisherigen Lehransatz. Auch explizit offene Aufgaben, die die Kreativität anregen sollten und dabei häufig einen Grafikbezug hatten, wurden von Studierenden in einem Einführungsmodul der Informatik geschätzt [Van15].

Grundsätzlich werden Aufgaben mit verpflichtender Abgabe in der Forschung positiv bewertet, da kontinuierliche Mitarbeit zum Lernerfolg beiträgt und Rückstände verhindern kann [KL21]. Nach der Einführung der Abgabepflicht in einem Mathematikkurs stellten Krapf und Liebendörfer [KL21] eine erhöhte Klausurteilnahme und bessere Bestehensquoten fest, während die Bewertungen grundsätzlich auf dem gleichen Level wie bisher blieben. Dabei war auch ein Ziel die soziale Interaktion in Gruppen von Studierenden zu stärken. Aus der Evaluation geht hervor, dass Studierende die Abgabepflicht überwiegend befürworten, da diese eine Beschäftigung mit den Inhalten erzwingen würde und einen positiven Druck ausübe. Als Nachteile wurden negativer Druck und mögliches Kopieren von Lösungen anderer genannt, wobei diese insgesamt von den Studierenden als nachrangig zu den Vorteilen angesehen wurden. Auch in einer weiteren Studie haben sich Bachelorstudierende der Mathematik retrospektiv mehr Druck durch die Dozenten gewünscht, wenn keine Pflichtabgaben vorhanden waren [BMRW05]. Dies betraf insbesondere Studierende, die Schwierigkeiten mit den Lerninhalten gehabt hatten.

Nach Rach und Heinze [RH13] sind dabei Studierende, die Lösungen von anderen lediglich kopiert haben, weniger erfolgreich in ihrem Studium. Bereits das Verständnis und die damit eng verbundene Fähigkeit die Lösung erklären zu können, hängen jedoch mit einem besseren Erfolg zusammen. Die erfolgreichsten Studierenden lösen die Aufgaben vollständig selbst. Aus der schulmathematischen Forschung kommt die Forderung, dass Aufgaben für optimalen Erfolg angemessen fordernd und interessant sein sollten [DTL⁺11]. Grundsätzlich erleichtern Aufgaben dann die Vorbereitung auf die Klausur und bringen Struktur in den Lernprozess [Göl19, S. 383]). Außerdem seien sie hilfreich für die Selbstwirksamkeitserwartungen von Studierenden, da diese durch erfolgreiches Lösen ein besseres Verhältnis zum Lerngegenstand entwickeln würden [Göl19, S. 45].

Um das Kopieren von Lösungen anderer Studierender zu verhindern, existieren verschiedene Strategien unter Lehrenden der Informatik [SSB⁺17]. Durch ausführliche Informationen über den Zweck der Aufgaben, aber auch Vertrauen und angemessene Unterstützung können die Studierenden motiviert werden selbst an Lösungen zu arbeiten [SSB⁺17]. Durch stärkere Kontrollen bis hin zur Individualisierung von Aufgaben kann das Kopieren von Lösungen zudem erschwert werden, sodass der Aufwand von den Studierenden möglichst zugunsten eigener Bearbeitung abgewogen wird [SSB⁺17].

2.1.3 Forschung zu Gamification

Nach Deterding et al. [DDKN11] bezeichnet der Begriff *Gamification* den Einsatz von Designelementen aus Spielen in Kontexten außerhalb von Spielen. Als Designelemente können dabei Interaktionskomponenten wie Level, Abzeichen und Fortschrittsanzeigen, Spielmechaniken wie begrenzte Ressourcen oder ausführbare Spielzüge, grundsätzliche Prinzipien wie Spielziele, Modellierungselemente wie Themen und Welten sowie die Methode des Spielens selbst eingesetzt werden. Einen ähnlichen Ansatz für Elemente eines Games beschreiben auch Brathwaite und Schneider [BS08]. Als *atomare Elemente* eines Spiels werden hier Spielfiguren und deren Avatare sowie andere elementare Spielelemente, die Mechanik, die Dynamik, die Ziele sowie die Inhalte im Sinne eines Kontexts für das Spiel beschrieben. Dabei greifen die Elemente ineinander, sodass z. B. durch eine festgelegte Mechanik in Form von Regeln, möglichen Spielzügen und Ressourcen durch Spielfiguren und weiteren Elementen eine bestimmte Grunddynamik umgesetzt wird. Als Dynamik wird dabei das Muster des Spiels bezeichnet, beispielsweise ein Rennspiel oder ein Kartenspiel mit dem Ziel möglichst viele oder wenige Karten auf der Hand zu haben. Ziele haben dabei vor allem die Funktion die Spielenden zu bestimmten Handlungen zu motivieren [BS08, S. 31].

Gamification wird dabei, auch unter dem Begriff des *Gameful Design* auf zwei Dimensionen von den ähnlichen Begriffen *Playful Design* und *(Serious) Games* abgegrenzt. Ein Serious Game bezeichnet dabei die vollständige Umsetzung eines Spiels, während das Gameful design lediglich Elemente der Spiele enthält. Die Abgrenzung zum Playful Design betont den Unterschied zwischen den englischen Begriffen *Gaming* und *Playing*. Während letzteres das freie Spielen ohne Rahmen und Regeln inkludiert, fokussiert sich Gaming auf konkrete, vorab festgelegte Spielmechanismen [DDKN11, LA10]. Dennoch ist häufig auch die Intention der Erstellenden von Bedeutung, wenn es darum geht konkrete Beispiele einzuordnen und abzugrenzen [DDKN11].

Gamification wird häufig eingesetzt, um mehr Freude beim Erledigen einer bestimmten Aufgabe zu erzeugen und dadurch die Motivation und Ausdauer zu stärken [FGN⁺11, DDKN11]. Nach Malone [Mal81] existieren aus motivationspsychologischer Sicht drei Gründe, wieso Spiele insbesondere auch die intrinsische Motivation fördern. Diese beziehen sich auf die Herausforderung, die Fantasie sowie die Neugier der Lernenden. Eine Herausforderung zeichnet sich dabei dadurch aus, dass ein bestimmtes Ziel erreicht werden soll, dessen Ausgang ungewiss ist. Motivierend wirken dabei auch persönlich interessante Ziele und die Möglichkeit unterschiedliche Schwierigkeitsstufen über verschiedene Level zu bearbeiten. Das Erreichen eines Ziels wird dabei von der spielenden Person als positiv für die Selbstwahrnehmung aufgefasst und bereits die Aussicht darauf verstärkt die Motivation. Nach Groening und Binnewies [GB19] tragen Gamificationelemente ähnlich zur Motivation bei wie klassische Zielsetzungen.

Unter dem Aspekt der Fantasie wird das Thema verstanden. Dieses kann durch kognitive oder emotionale Verbindungen von Interesse sein. Eine thematische Einordnung kann dabei extrinsisch lediglich die Folgen einer Handlung bestimmen oder intrinsisch eine Wechselwirkung aus Handlungen, Folgen und dem daraus resultierenden Fortschritt darstellen. Eine extrinsische Fantasie liegt vor, wenn eine Handlung beispielsweise zum direkten Gewinn oder Verlust des Spiels führt. Eine intrinsische Fantasie spiegelt

dagegen konkret wieder warum ein Ergebnis auf die Handlung folgt. Dies kann z. B. durch die Sichtung der Differenz zwischen dem getroffenen Ziel bei einem Spiel und dem gewünschten Ziel durch die daraus resultierende Zielkorrektur zu einer Verbesserung der Fähigkeiten führen. [Mal81]

Um die Neugier einer spielenden Person zu erreichen, ist ein angemessenes Komplexitätslevel notwendig [Mal81, GB19]. Beim Lernen sollte dabei an bestehendes Wissen angeknüpft werden, das in nicht überfordernder Art ergänzt wird. Neue und überraschende Inhalte können dabei genauso wie sensorische Effekte Neugier erzeugen. Effekte auditiver oder visueller Art können mit verschiedenen konkreten Zielen eingesetzt werden. Neben rein dekorativer Verwendung oder der Verwendung von Effekten für die Darstellung des Inhalts können sie auch eine Belohnung darstellen oder beim Feedback helfen, indem empfohlene weitere Strategien z. B. graphisch dargestellt werden. Auf kognitiver Ebene kann Neugier durch Malone [Mal81] dadurch erreicht werden, dass Lernende Defizite in ihrem aktuellen Wissensstand bezüglich der Vollständigkeit, der Konsistenz oder der Effizienz dieser feststellen. Nach Moore und Anderson [MA68] ist angemessenes Feedback von besonderer Bedeutung für den Lernerfolg in einer anspruchsvoll entwickelten Lernumgebung. Für Malone [Mal81] lässt sich dieses im Hinblick auf die Neugier am besten umsetzen, wenn die Folge einer Handlung für den Lernenden überraschend erscheint. Um jedoch auch lehrreich zu sein, sollte es gleichzeitig konstruktiv vermittelt werden. Für ein Spiel könnte das bedeuten, dass eine Strategie, die in einem einfacheren Level erfolgreich war, in einem höheren Level nicht mehr funktioniert. Wenn dem Lernenden dies verbunden mit einem Hinweis auf mögliche alternative Strategien aufgezeigt wird, ist eine für das Lernen förderliche Neugier zu erwarten. Diese sollte dann zu weiteren Lösungsversuchen unter Berücksichtigung des Hinweises führen. Durch das unerwartete Ergebnis entsteht eine Dissonanz bei der lernenden Person, die zu einer Auflösung dieser motiviert [Fes62, S. 3]. Bei Spielen, an denen mehr als eine Person teilnimmt, sollten zusätzlich Wettbewerb und Kooperation als möglicherweise auf die Motivation wirkende Effekte betrachtet werden [Mal81]. Außerdem kann bereits die Illusion eine Wahl zu haben, wie dies bei Spielen durch verschiedene Züge und Handlungen ermöglicht wird, eine ähnlich starke Auswirkung auf die Motivation haben wie extrinsische Anreize (z. B. Bestrafungen) [Zim66].

Auch in der grundsätzlichen Lernmotivationsforschung finden sich Vorgehen zur Motivation Lernender, die zentralen Elementen von Games entsprechen. Für Pintrich [Pin03] sind dabei vor allem angemessene Ziele und interessante Aufgaben zentral. Eine zumindest kurzzeitige Motivation im Rahmen einer *situativen Motivation* kann auch durch die Gestaltung der Aufgabe selbst, unabhängig von der Motivation durch den Inhalt, ausgelöst werden [Pin03].

In einer Literaturrecherche zum motivationalen Erfolg von Gamification im Allgemeinen analysierten Hamari et al. [HKS14] 24 Studien, die eine empirische Betrachtung eines Gamificationeinsatzes enthalten. Bei den eingesetzten Gamificationelementen waren die häufigsten dabei vor allem Ranglisten mit zehn Publikationen sowie Punktesysteme und Auszeichnungen mit jeweils neun Veröffentlichungen. Sieben Studien enthielten Wettbewerbe, jeweils sechs enthielten Level, Sachkontexte oder Feedback. Jeweils vier Studien inkludierten klare Ziele, Belohnungen oder Feedback. Nahezu alle Studien (21) betrachteten Auswirkungen der Gamification im Verhalten beispielweise durch

die Messung von Nutzungsdaten. Zwölf Studien setzten auch weitere Messinstrumente ein, um psychologische Wirkungen zu erfassen. Dabei wurden z. B. Motivation oder Freude erfragt, jedoch nur in einem Fall mit einer validierten Skala. 17 Studien erfassten qualitative Daten, zwei quantitative und fünf Daten beider Arten als Mixed-Methods-Studien. Dabei weisen zwei Studien ausschließlich positive und zwölf weitere Studien teilweise positive Ergebnisse auf. Sieben Studien haben lediglich deskriptive Ergebnisse und keine Veröffentlichung berichtet von negativen oder ausschließlich nicht signifikanten Ergebnissen. Bei den Kontexten des Gamificationeinsatzes stellt die Bildung mit neun Studien den größten zusammenhängenden Bereich. Allerdings wird Gamification auch z. B. in beruflichen oder organisatorischen Systemen (jeweils vier Studien) sowie vereinzelt in weiteren Bereichen eingesetzt. Insgesamt beurteilen die Autoren die Studienlage noch als gering, zumal viele Studien nur geringe Teilnehmerzahlen von etwa 20 Personen haben und Neuigkeitseffekte ebenfalls Einfluss auf die Ergebnisse haben könnten.

Um den Einsatz von Gamification in Lehrkontexten zu erfassen, führten Dicheva et al. [DDAA15] eine Literaturrecherche durch. Diese zielte darauf ab die Lehrkontexte und die eingesetzten Gamification-Elemente zu klassifizieren. Dabei wurden nach Exklusion nicht relevanter Beiträge 34 Veröffentlichungen, die empirisch ein konkret beschriebenes Gamificationkonzept in einem Lehrkontext untersuchten, analysiert und eingeordnet. Eine Kategorie zur Einordnung der Konzepte sind die Designprinzipien, wobei lediglich sechs der 34 Konzepte nur ein Designprinzip verwenden. Am häufigsten sind neben der grundsätzlichen visuellen Sichtbarkeit von Erfolgen soziale Interaktionen zu finden, die sowohl den Wettbewerb zwischen den Studierenden als auch Kooperationen beinhalten. Einige der gefundenen Konzepte geben den Lernenden Freiräume entweder bei der Wahl von Aufgaben oder ermöglichen das Scheitern ohne Nachteile. Die fehlenden Konsequenzen seien allerdings umstritten und bislang kaum empirisch untersucht [DDAA15]. Unter den konkreten Elementen, die im Rahmen der Gamification eingesetzt werden, wurden vor allem Punkte, Auszeichnungen und Bestenlisten sowie Level identifiziert.

Die Einsatzszenarien involvieren überwiegend digitale Angebote. Bei 18 der 34 Veröffentlichungen wurde Gamification in einem Blended Learning Kurs eingesetzt und lediglich in sieben Fällen handelt es sich um Kurse, die vollständig ohne digitale Anteile konzipiert sind. Da zwei Kurse im Kontext der Schulbildung stattgefunden haben, ist es wahrscheinlich, dass lediglich fünf der Kurse an Hochschulen keine digitalen Angebote inkludieren. Bis auf sechs fachneutrale Veröffentlichungen, die Plattformen beschreiben, liegt die fachliche Einbettung der Konzepte in der Informatik, Spieleprogrammierung, Mathematik oder Natur- oder Ingenieurwissenschaften. Die Autorinnen und Autoren vermuten, dass die Fähigkeiten zum Aufbau einer entsprechenden Plattform vor allem bei den Lehrenden dieser Fächer vorhanden sind. Durch die priorisierte Suche in entsprechenden Fachdatenbanken kann allerdings auch ein Bias dadurch nicht ausgeschlossen werden. Unter den Ergebnissen finden sich vor allem positive Evaluationen. Lediglich in einem Fall wurde das Gesamtergebnis als negativ klassifiziert, während 18 von 34 Veröffentlichungen von rein positiven Ergebnissen berichten. Bereits Gestaltungselemente wie Abzeichen könnten dabei positiv auf die Lernenden wirken ohne direkte Auswirkungen auf die Endnoten zu haben. Einige Lehrende, die den Erfolg noch als unklar beurteilen, erkennen teilweise eine mögliche Überforderung der Lernenden durch die Wahlmöglichkeiten, die viele so nicht gewohnt seien. [DDAA15]

Einige frühe Ansätze von Gamification in Software Engineering Kursen zwischen 2011 und 2014 haben Pedreira et al. [PGBP15] im Rahmen einer systematischen Einordnung untersucht. Neben der Einordnung in den konkreten Teil des Software Engineering Prozesses wurde bei 29 Veröffentlichungen analysiert, welche Gamificationelemente eingesetzt wurden, wie dies methodisch beforscht wurde und welche Art der Publikation vorliegt. Die höchste Zahl an Veröffentlichungen für einen Software Engineering Prozess wurde mit elf Beiträgen der Implementierung zugeordnet. Bei den Gamificationelementen verwenden 14 Beiträge Punktesysteme, sieben Abzeichen und jeweils vier Ranglisten sowie soziales Ansehen. Nur in einzelnen Publikationen wurden auch Level, visuelle Metaphern und weitere Elemente verwendet. Nahezu die Hälfte (48 %) der Beiträge enthielt keine Beschreibung einer konkreten Umsetzung oder Evaluation, sondern war lediglich konzeptionell-beschreibend (38 %) oder philosophisch (10 %) angelegt. Die Quellen der Veröffentlichungen waren zu 47 % Konferenzen, zu 39 % Workshops und zu 7 % Journals. Die restlichen Publikationen wurden in sonstigen Quellen gefunden. Insgesamt betonen die Autoren, dass bisher wenig systematische Evaluationen zum Einsatz und Nutzen von Gamification im Bereich des Software Engineerings vorliegt und sich die meisten Konzepte lediglich auf Punkte- oder Auszeichnungssysteme mit zugehörigen Vergleichen zwischen den Teilnehmenden beziehen.

Azmi et al. [AIA15] haben 2015 eine Literaturanalyse zu Gamification in online durchgeführten kollaborativen Programmierkursen durchgeführt, deren Methodik jedoch nur teilweise beschrieben ist. Sie kategorisieren die Elemente in die Kategorien Mechanik, Dynamik und Ästhetik. Die meisten untersuchten Veröffentlichungen haben dabei Aspekte von Spielmechanik umgesetzt, die wenigsten ästhetische Elemente. Im Rahmen der mechanischen Elemente wurden auch hier vor allem Auszeichnungen, Ranglisten und Punktesysteme eingesetzt. Diese und andere Ansätze werden in drei verschiedenen Situationen eingesetzt. Zum einen kann das Lernen selbst mit Gamificationelementen angereichert werden. Zum anderen kann aber auch explizit die soziale Interaktion durch Gamification gefördert werden. Als dritte Möglichkeit ergibt sich Gamification bei der Prüfung der erlernten Kompetenzen.

Eine konkrete Anwendung von Gamification zur Bewertung für Gruppenleistungen beschreiben Moccozet et al. [MTOL13]. In einem Kurs zu Informationssystemen für Studierende im ersten Studienjahr hatten diese die Möglichkeit sich auf verschiedene Weisen über eine digitale Plattform inhaltlich und organisatorisch am Lernen zu beteiligen. Für das Lesen oder Kommentieren von Beiträgen anderer Studierender, aber auch für das Moderieren von Lösungsprozessen erhielten die Studierenden jeweils Punkte, die sie jedoch nicht einsehen konnten. Diese Punkte verbesserten am Ende des Kurses die Note aus der Klausur. Bei der Punkteverteilung stellen die Autoren fest, dass sich die Aktivität der einzelnen Studierenden vor allem innerhalb der Gruppen unterschieden hat. Die Aktivitäten fanden passend zu den Veranstaltungs- und Abgabeterminen der einzureichenden Ergebnisse statt, wobei tendenziell mit weniger aufwendigen Aktivitäten begonnen wurde und zum Ende des Kurszeitraums insgesamt mehr Punkte verteilt wurden.

Auch im Rahmen eines Flipped Classrooms wurden in Informatikmodulen bereits verschiedene Gamificationansätze in Fallstudien beschrieben. Dicheva et al. [DID19] verglichen ihren bisherigen kleinen Datenstrukturenkurs mit einem Kurs, in dem ein

Gamification-Konzept für Selbstlernaktivitäten eingesetzt wurde. Auf einer Lernplattform konnten dabei Punkte für das erfolgreiche Absolvieren von Lernaktivitäten außerhalb der Pflichtaufgaben gesammelt und für selbst gewählte Belohnungen für die Bewertung oder für Zeitvorteile genutzt werden. Dieses Konzept führte zu geringeren Fehlerquoten und guter Akzeptanz bei den Studierenden.

In einem größeren CS1-Kurs kombinierten Latulipe et al. [LLS15] die Zusammenarbeit in festen Gruppen im Klassenzimmer als Teil eines Flipped Classrooms mit Auszeichnungen für aktive Teilnahme. Dabei führten zehn Auszeichnungen zu einer Verbesserung der Endnote um 1 %. Neben der allgemeinen Zufriedenheit führte dies zu einer besseren Vernetzung der Studierenden untereinander. Aufbauend auf dieser Forschung entwickelten Sprint und Fox [SF20] einen CS1-Kurs, der in mit jeweils 21 bis 44 Teilnehmenden pro Studiengruppe und mehreren Kombinationen für den Einsatz von festen Gruppen sowie Gamification-Elementen erprobt wurde. Dabei wurde vor allem gutes Zeitverhalten der Studierenden belohnt, indem Punkte für die frühzeitige Abgabe von Programmierlösungen vergeben wurden. Aber auch Fragen und Antworten im Forum führten zur Vergabe von Punkten. Als Belohnung konnten die besten Teams für ausgewählte Fragen in der Abschlussprüfung eine höhere Punktzahl erreichen.

Fotaris et al. [FMLR16] setzten in einem Programmierkurs verschiedene Arten von Quizzes ein. Dabei wurden Formate wie *kahoot!* und eine Adaption von *Wer wird Millionär?* für kurze Fragen und eine Plattform mit Aufgaben und Codeeditor für größere Programmieraufgaben verwendet. Die Studierenden äußerten sich positiv zu diesem Konzept, das auf sie leicht motivierend wirke. Sie gaben an dadurch ein besseres Verständnis erlangt zu haben und die Interaktionen mit anderen Studierenden stiegen.

Im Rahmen eines Einführungskurses zum Programmieren wollten Knutas et al. [KINP14] die Kollaboration zwischen Studierenden durch Gamification explizit fördern. Dabei wurden auf einer Forumsplattform Punkte für die Aktivität von Studierenden vergeben, wenn diese Fragen beantwortet oder Bewertungen abgegeben haben. Hohe Aktivitäten in der Interaktion mit anderen wurden dabei mit vielen Punkten belohnt, wobei diese allen Teilnehmenden angezeigt wurden. Am Kursende waren 87 % der gestellten 88 Fragen durch durchschnittlich etwa 3 Studierende und zum Teil Übungsleitende beantwortet. Obwohl die durchschnittliche Antwortzeit bei 19,8 Stunden lag, wurden 50 % der Anliegen innerhalb von zwei Stunden gelöst. Dabei ist zu beachten, dass 76 % der Fragen außerhalb klassischer Arbeitszeiten vor allem am Wochenende oder abends gestellt wurden. Durch die Möglichkeit auf der Plattform eine Antwort zu bekommen, sanken die Mailanfragen an die Übungsleitenden um 88 %, wobei diese auch seltener die gleiche Fragen gestellt bekamen. Bei der Auswertung der Studierenden mit hoher Aktivität und hoher Punktzahl ergab sich eine Korrelation zum Studienerfolg. Die Autoren schließen daher darauf, dass Gamification besonders die sehr guten Studierenden dazu motiviert anderen im Kurs zu helfen.

Einen ebenfalls digitalbasierten Ansatz wählten Ohno et al. [OYT13], um ihren Ingenieursstudierenden Interaktionen unter höherer Anonymität zu ermöglichen als in den Präsenzveranstaltungen. Als Blended Learning ergänzte eine digitale Plattform dabei die gewünschten Diskussionen in der Präsenzveranstaltung. Die Studierenden gestalteten sich dabei einen Avater als Spielfigur und konnten damit von den Lehrenden gestellte Quizfragen beantworten sowie Erklärungen und Fragen diskutieren. Elf der 17

Studierenden im Kurs empfanden die Anonymität auf der Quizplattform als sinnvoll, während vier diese nicht benötigt hätten. Die übrigen zwei Studierenden enthielten sich. Zwölf Studierende gaben an aufgrund der Plattform mehr Fragen stellen zu würden als sonst, fünf verneinten dies. 15 Studierende empfanden die Plattform als grundsätzlich vorteilhaft und 16 würden sich eine solche Plattform für ihre Kurs grundsätzlich wünschen. Einzelne Studierende äußerten Befürchtungen, dass die Anonymität zu qualitativ schlechten Beiträgen führen könnte oder die Existenz der digitalen Plattform die Diskussionen in den Präsenzveranstaltungen einschränken könnte. Um die erste Befürchtung zu vermeiden, könnten die Lehrenden den Studierenden erklären, dass diese lediglich für die anderen Studierenden anonym wären und bei Missbrauch der Plattform der Urheber entsprechender Beiträge ermittelt werden könne.

Hurtienne et al. [HSS15] sprechen sich dafür aus, dass Gamificationelemente adaptierbar sein sollten, um durch Wettbewerbseffekte die intrinsische Motivation einiger Studierender nicht zu mindern. Sie testeten daher einen Ansatz, in dem die Studierenden in einem Programmierereinführungsmodul die Elemente eines Spiels zu- und abschalten konnten, sodass wahlweise keine Gamification, ein Single-Player-Spiel oder ein Spiel mit Konkurrenz durch andere Studierende entstand. Etwa die Hälfte der Studierenden wählte dabei den maximal möglichen Ansatz mit Konkurrenz, wenige Studierende verzichteten allerdings vollständig auf Gamification-Elemente.

In weiteren verwandten Fächern wurden Gamificationansätze ebenfalls bereits erfolgreich umgesetzt. In einem Statistik-Grundlagenkurs untersuchte Weyers [Wey18] ein Flipped-Classroom-Konzept, bei dem die Studierenden sowohl für sich selbst als auch für die gesamte Gruppe Punkte sammeln konnten. Diese führten zur direkten Verbesserung der Endnote, aber auch zu Vorteilen beim Prüfungsdesign. Unter den Studierenden konnten eine gute Beteiligung und ein hohes Interesse an den Präsenzveranstaltungen beobachtet werden. Allerdings wurde das kollektive Punktesammeln besser bewertet als der individuelle Wettbewerb.

Barata et al. [BGJG13] haben einen auf zwei Gruppen aufgeteilten Kurs zur Produktion multimedialer Inhalte mit etwa 40 Masterstudierenden pro Jahr mit einem Gamification-Konzept versehen und mit dem bisherigen Kurskonzept verglichen. Auf einer Plattform waren dabei in unterschiedlichen Kategorien wie z. B. Quizzes, Download von Lernmaterialien oder Forenbeiträge verschiedene Stufen erreichbar, für die Erfahrungspunkte gutgeschrieben wurden. Je 900 Erfahrungspunkte erfolgte ein Levelaufstieg, wobei die Level direkt in Endnoten übertragen wurden. In diesem Konzept wurden deutlich mehr Downloads der Lernmaterialien registriert. Außerdem verfassten die Studierenden besonders zu Kursbeginn deutlich mehr Forenbeiträge. Die Anwesenheit in den regulären Veranstaltungen war mit der Gamification signifikant höher, bei Gastvorträgen jedoch etwas niedriger. Die Noten der Studierenden haben sich im Vergleich zur regulären Durchführung nicht verändert.

Grundsätzlich haben soziale Faktoren und die Interaktion mit anderen in der gleichen Situation auch innerhalb von Gamificationmodellen einen hohen Stellenwert. So stellten Hamari und Koivisto [HK13] fest, dass innerhalb einer Gamification-App, die Auszeichnungen für das Erreichen sportlicher Ziele vergeben hat, die gegenseitige Verstärkung im Kommunikationskreis der Nutzenden einer der wichtigsten Faktoren für den Erfolg war. Diese Erkenntnis unterstützt vor allem kooperative Gamificationkonzepte. Auch in

der grundsätzlichen Lernforschung weisen kooperative Ansätze größere Erfolge auf als kompetitive [JJS86].

Insgesamt wird Gamification daher als vielversprechender Ansatz gesehen, unter dem jedoch diverse Konzepte möglich sind und zu dem bisher wenig empirische Forschungsergebnisse vorliegen. Der Einsatz spielerischer Elemente kann dabei bei Berücksichtigung einiger Designprinzipien motivierend wirken. Im Rahmen der aktuellen Forschungslage bleibt jedoch offen, ob kompetitive oder kooperative Konzepte in Einführungsmodulen der Informatik zielführender sind, da in diesem Kontext lediglich Fallstudien vorliegen.

2.2 Forschung zu digitalen Lernmaterialien

Schon seit 1983 debattierten Clark [Cla83] und Kozma [Koz94] darüber, welche Rolle Medien für das Lernen haben. Clark [Cla83] vertrat dabei die Ansicht, dass Medien lediglich Lerninhalte bereitstellen, aber für sich selbst genommen keinen Lernprozess verursachen. Dabei argumentierte dieser mit der Möglichkeit das Medium auszutauschen und unterschied dabei Medium und Methode. Kozma [Koz94] stellte Medien dagegen in den Kontext der Methode und erkannte Auswirkungen der Medien auf das Lernen. Zu Zeiten digitaler Lehre sind Medien jedoch sowohl als Werkzeug als auch als Methode wichtig für die Gestaltung der Lehre geworden. In einer Metaanalyse von Studien zu Videos in der Lehre überwiegend in MINT-Fächern zwischen 2007 und 2017 haben Poquet et al. [PLMD18] gezeigt, dass diverse Faktoren wie die Darstellung, das Einsatzszenario, Inhalte und Aufgaben bei der Gestaltung und im Umgang mit digitalen Lernmaterialien relevant für die Lernwirkung sein können. Auch Schulmeister diskutierte bereits 2001 den Einsatz von Medien in der Hochschullehre vor dem Hintergrund der Kommerzialisierung durch externe Anbieter [Sch01].

Eine weit verbreitete Form digitaler Lernmaterialien sind Lehrvideos. Nach Mayer [May21] sind Lehrvideos eine Art *multimedialer Lehrmitteilung*. Diese seien Präsentationen bestehend aus Worten und Visualisierungen und haben die Förderung von Lernen zum Zweck [May21]. Nach Persike [Per19] handelt es sich bei Videos in der Lehre um audiovisuelle Aufbereitungen, die Lerninhalte vermitteln. Persike unterscheidet dabei Erklärvideos von Demonstrationsvideos. Während Erklärvideos explizit zum Erklären eines fachlichen Inhalts erstellt wurden und diesen bewusst erläutern, handelt es sich bei Demonstrationsvideos beispielsweise um Aufzeichnungen von Praxissituationen. Diese können in der Lehre hilfreich sein, obwohl die dabei aufgezeichnete Handlung nicht primär für die Lehre erstellt wurde. Der Nutzen für die Lehre entstünde dabei erst durch die Auseinandersetzung und entsprechende Rahmung der gezeigten Situation. Erklärvideos werden nach Persike [Per19] meistens online wahlweise in synchronen oder asynchronen Lehrszenarien angeboten.

Dabei können verschiedene Typen von Lehrvideos unterschieden werden [Per19]. Als *Live Digitized Lecture* wird eine mitgeschnittene klassische Vorlesung mit Publikum bezeichnet, die in den meisten Fällen erst nach der Veranstaltung zur Verfügung gestellt wird. Eine Variante davon ist die *E-Lecture*, bei der die Aufnahme ohne Publikum erfolgt. Dabei können die Aufnahmebedingungen entsprechend angepasst werden, sodass beispielsweise das Format des *Talking Head* möglich ist, bei dem die sprechende Person

einen relevanten Teil des Bildes einnimmt. Mit niedrigem Aufwand ohne weitere Einblendungen produzierte E-Lectures können auch als *Videopodcasts* bezeichnet werden. Zwei weitere verbreitete Arten von Lehrvideos sind *Videos in Lege- und Zeichentechnik* sowie *Screencasts* mit der Unterform *Slidecast*. Bei Videos in Lege- und Zeichentechnik wird eine horizontale Fläche gefilmt, auf der Objekte manuell platziert werden können sowie geschrieben und gezeichnet werden kann. Die sprechende Person ist meistens lediglich zu hören, aber nicht zu sehen. Bei Screencasts wird dagegen der Bildschirm aufgezeichnet, bei der Sonderform Slidecast konkret ein Präsentationsprogramm für Folien. Hierbei sind sowohl eine Audioaufnahmen als Begleitung, aber auch die Einblendung der sprechenden Person, möglich. Lehrvideos können in Form von Vorlesungsaufzeichnungen rein *substituierend* für eine Vorlesung eingesetzt werden. Bei kleinen Anpassungen wie der Möglichkeit die Geschwindigkeit zu regeln, Untertitel einzublenden oder ein längeres Video in kurze Abschnitte zu unterteilen, wird der Einsatz eines Videos als *Augmentation* bezeichnet. Beim Einsatz von Videos in aktiven Lernkontexten wie Flipped Classroom-Szenarien (siehe 2.1.1) handelt es sich um eine *Modifikation*. Die Lehrvideos sollten dabei entsprechend aktivierend und zum Selbstlernen geeignet aufbereitet sein. Außerdem sollte durch die Präsenzphasen und geeignete Aufgaben ein Rahmen für die eingesetzten Materialien geschaffen werden. Unter der *Redefinition* von Lehrvideos werden Prozesse verstanden, in denen das Video nicht mehr zum inhaltlichen Lernzweck durch Lehrende an Lernende gegeben wird, sondern das Video selbst zum Teil des Lernprozesses wird. Dies ist beispielsweise dann der Fall, wenn ein Video von den Lernenden als Ergebnis ihres Lernprozesses entstehen soll. [Per19]

Eine wichtige Grundlage für die Gestaltung von multimedialen Lerninhalten ist die *Cognitive Load Theory* nach Sweller [SVMP98]. Diese geht davon aus, dass beim Lernen drei Belastungen auf das Arbeitsgedächtnis wirken: die *intrinsische* kognitive Last, die *extrinsische* kognitive Last sowie die *lernbezogene* kognitive Last. Insgesamt dürfen die Lasten die Kapazität des Arbeitsgedächtnisses nicht überschreiten und sollten daher möglichst reduziert werden. Die intrinsische kognitive Last entsteht durch den Lerninhalt als solchen. Diese Last lässt sich verringern, wenn der Lerninhalt in kleinere Bestandteile aufgeteilt ist. Diese sind jeweils einfacher zu lernen und vermitteln dadurch ausreichend Vorwissen, um den komplexen Inhalt dennoch einfach zu verstehen. Wenn gleichzeitig viele neue Bestandteile eines Inhalts verarbeitet werden müssen, ist die intrinsische kognitive Last hoch. Die extrinsische kognitive Last ergibt sich aus der Gestaltung der Lernmaterialien. Sind diese besonders lernförderlich gestaltet, ist die extrinsische Last geringer. Ablenkende Elemente erhöhen die extrinsische kognitive Last. Durch Senken der intrinsischen und extrinsischen kognitiven Last kann möglichst viel Kapazität für die lernbezogene kognitive Last geschaffen werden. Diese liegt im kognitiven Aufwand des Lernens und dem Aufbau entsprechender kognitiver Strukturen. Die lernbezogene kognitive Last, auch *Germane Cognitive Load* genannt, ist nur bedingt beeinflussbar [Swe06]. Für multimediale Lernmaterialien ist die Reduktion der extrinsischen kognitiven Last die relevanteste Ansatzstelle, da die kognitive Belastung beim Lernen durch geeignete Aufbereitung des Materials verringert werden kann.

Auf dieser und weiterer psychologischer Lernforschung bauen eine Reihe von Gestaltungsprinzipien für multimediale Lernmaterialien auf. Als übergreifende Regel kann das Multimediaprinzip nach Mayer [May21] verstanden werden. Dabei ist zentral, dass Lernen mit unterstützenden grafischen Elementen besser funktioniert als mit Texten alleine.

Für die weitere Gestaltung relevant sind auch das Kohärenzprinzip, das Redundanzprinzip, das Modalitätsprinzip, das Signalisierungsprinzip und das Kontinguitätsprinzip. Das Kohärenzprinzip bezieht sich darauf, dass nicht inhaltsrelevante Aspekte unnötige Fokussierung auf sich ziehen können und daher weggelassen werden sollten. Ähnlich bedeutet das Redundanzprinzip, dass wiederholende Texte oder Elemente entfernt werden sollten. Abseits von sprachunterstützenden Untertiteln, die insbesondere beim Erlernen von Sprachen und Begriffen mehrere Ebenen ansprechen können [Dan04], sollte gesprochener Text nicht noch im Video sichtbar sein. Nach dem Modalitätsprinzip ist gesprochener Text mit Ausnahme von Sprachunterricht auch grundsätzlich wirkungsvoller als geschriebener. Unter dem Signalisierungsprinzip wird das Hervorheben besonders relevanter Inhalte verstanden, wobei dies sowohl verbal als auch visuell erfolgen kann. Das Kontinguitätsprinzip thematisiert die zeitliche und räumliche Passung verschiedener Bestandteile des multimedialen Inhalts. Für die zeitliche Kontinguität ist es wichtig, dass das Audio mit dem Video zusammen passen und der gesprochene Text das aktuell sichtbare erläutert. Als räumliche Kontinguität wird direkt in Grafiken platzierter Text bezeichnet. Dabei soll die Situation umgangen werden, dass Beschriftungen erst umständlich beispielweise über Nummern den Bestandteilen einer Grafik zugeordnet werden müssen [May21].

Konkret aufbauend auf die Cognitive Load Theory sind das Segmentierungsprinzip und das Pre-Trainingsprinzip [May21]. Beim Segmentierungsprinzip wird ein umfangreicher Inhalt in kleinere Einheiten unterteilt, wobei diese den intrinsischen Cognitive Load senken können. Ähnlich wirkt das Pre-Trainingsprinzip, bei dem Vorwissen aufgebaut und aktiviert wird, um ebenfalls die intrinsischen kognitive Last zu senken. Für eine stärkere Motivation und ein angenehmeres Lernen zur Optimierung der lernbezogenen kognitiven Last kann der Einsatz des Personalisierungs-, Verkörperungs- und des Stimmenprinzips hilfreich sein. Dabei sollte der Lernende direkt angesprochen werden (Personalisierungsprinzip) und bei Aufnahmen von Tätigkeiten sollte eine Perspektive gewählt werden, die diese aus Sicht des Lernenden zeigt (Verkörperungsprinzip). Audioaufnahmen sollten mit menschlichen Stimmen statt mit technisch generierten erfolgen (Stimmenprinzip) [May21].

In der Praxisforschung sind große Datenmengen zur Nutzung von Lernvideos häufig über *MOOCs* (*Massive Open Online Courses*)-Plattformen analysiert worden. Kim et al. [KGS⁺14] haben Videos aus vier MOOCs zu Themen aus den Fächern Informatik, Chemie und Statistik im Hinblick auf die Interaktionen wie Wiederholungen, Pausen und Abbrüche der Lernenden analysiert. Ein relevanter Teil an Abbrüchen findet dabei bereits während der ersten 3 % der Länge eines Videos statt, mutmaßlich aufgrund unpassender Erwartungen an die Inhalte oder die Aufmachung des Videos. Bei kurzen Videos mit einer Länge von etwa fünf Minuten finden dabei 35 % der Abbrüche direkt zu Beginn statt, bei längeren Videos von bis zu 20 Minuten sogar 47 %. Insgesamt wurden 53 % der kurzen Videos und 71 % der langen Videos im Verlauf von den Lernenden abgebrochen. Die stärksten Interaktionen in Form von Unterbrechungen sowie Wiederholungen der Wiedergabe fanden bei Wechseln in der Visualisierung oder im Thema statt. Die Autoren vermuten, dass Inhalte vor den Wechseln wiederholt werden sollten und Pausen dabei helfen wichtige Aspekte notieren zu können. Insgesamt empfehlen sie daher kurze Videos, die klar strukturiert sind, wobei auch Sprungmarken hilfreich sein können. Außerdem sollten schnelle Visualisierungswechsel vermieden werden. Eine interaktive Gestaltung und Zusammenfassungen der zentralen Aspekte wirken

dagegen lernförderlich. Im Kontext von Flipped Classroom-Szenarien gibt es jedoch auch Untersuchungen, bei denen Studierende eine Videolänge von etwa 20 Minuten für angemessen halten, wobei für die Studie Videos von bis zu 50 Minuten Länge einbezogen wurden [ZLM⁺09].

Zur Strukturierung von Videos stellen Altinpulluk et al. [AKFY19] sowie Zhi et al. [ZPM⁺19] fest, dass Segmentierung positiv wirkt. De Koning et al. [dTRP09] haben in einer Metastudie verschiedene Arten von Strukturierung durch Animationen, Pfeile und andere aufmerksamkeitslenkende Strukturelemente betrachtet. Dabei liegen gemischte Ergebnisse vor, die die Autoren darauf schließen lassen, dass das volle Potential einiger Effekte möglicherweise noch nicht genutzt wurde. Eine weitere Erklärung für negative Ergebnisse bestünde darin, dass in Videos häufig bereits genug Animationen durch das bewegte Bild selbst vorhanden sind und zu viele weitere Effekte daher ablenkend wirken können. Für die Praxis empfehlen sie eher Farben zur Markierung einzusetzen als weitere Objekte wie z. B. Pfeile und dabei insgesamt nicht zu übertreiben, sondern die visuelle Strukturierung in Maßen an sinnvollen Stellen einzusetzen.

Die Bedeutung der Kürze von Lernvideos bestätigen auch Guo et al. [GKR14]. Sie analysierten 6,9 Millionen Videosichtungen auf einer MOOC-Plattform mit mathematisch-naturwissenschaftlicher Ausrichtung. Dabei wurden Videos mit einer Länge bis sechs Minuten bevorzugt. Bei den konkreten Arten der Videos interagierten die Nutzer länger mit Videos, die neben Folien noch einen Talking Head zeigten. Auch dynamische Videos, in denen beispielsweise etwas geschrieben wurde, wiesen eine höhere Interaktion auf statische Screencasts. Als besonders wichtige Faktoren wurden jedoch Authentizität und Enthusiasmus festgestellt. Eine qualitativ besonders hochwertige Studioproduktion oder ein besonders langsames Sprechtempo seien dagegen nicht notwendig. Eine reine Vorlesungsaufzeichnung wäre allerdings ebenfalls kein gut geeignetes Format für Lehrvideos. Insgesamt müssten Kontext und Zweck beachtet werden. Ein bei der Studie von Guo et al. [GKR14] gemessener Indikator für Interaktion mit den Lehrvideos bestand in der Bearbeitung von materialbezogenen Aufgaben, die beispielweise als Quizfragen bei etwa einem Drittel der betrachteten Materialien der Sichtung folgten. Nach Koedinger et al. [KKJ⁺15], die Videos in einem MOOC in der Psychologie analysiert haben, hat Interaktivität in Form solcher Aufgaben grundsätzlich positive Auswirkungen auf das Lernen. Zur Bildqualität liegt außerdem eine ältere Studie vor, in der Ghinea und Thomas [GT05] das Verständnis und die Qualitätswahrnehmung für Videos mit verschiedenen Farbeinstellungen und Bildwiederholraten gemessen haben. Dabei stellten sie fest, dass diese keinen signifikanten Einfluss haben und eine besonders hohe Bildqualität bei Lehrvideos nicht notwendig ist.

Für die Auswirkungen der Darstellung der redenden Person in einem Lehrvideo liegen unterschiedliche Forschungsergebnisse vor. Nach Bateman und Schmidt-Borcherding [BSB18] sowie Wang und Antonenko [WA17] hat der Talking Head positive Auswirkungen auf den Cognitive Load für die Lernenden, die sich aber nicht beim Lernerfolg bemerkbar machen. Colliot und Jamed [CJ18] konnten keine Auswirkungen feststellen. Schmidt-Borcherding und Drendel [SBD21] verfolgten die Hypothese, dass Köherenz und dynamische Darstellung allgemein wichtiger sind als die Sprecherdarstellung und diese nur eine Möglichkeit für die Umsetzung dieser Kriterien wäre. Dabei konnten sie in einer Studie mit Videos mit und ohne Sprecheransicht sowie geringer und hoher

Kohärenz bestätigen, dass eine hohe Kohärenz einen positiven Effekt auf die kognitive Belastung hat. Beim Wissenszuwachs wirkte jedoch nur die Kombination aus Kohärenz und Talking Head positiv. Für den Lerntransfer ergaben sich keine Unterschiede. Nach Llyod und Robertson [LR12] sowie Griffin et al. [GMT09] ist ein kohärentes Audio zum Videobild grundsätzlich lernförderlich.

Viele Lehrende sorgen sich bei der Aufnahme digitaler Lernmaterialien darum, dass Fehler mit aufgezeichnet werden könnten und diese einen noch höheren Produktionsaufwand verursachen. Nach Oser [OHS99] und Weinert [Wei99] können Fehler unterschiedlichster Art sein und dadurch auch unterschiedlich wirken. Eine positive Wirkung ist dabei durchaus auch möglich, wenn Fehler bewusst zu pädagogischen Lerngelegenheiten werden. Dadurch kann eine positive Fehlerkultur vermittelt werden. Zudem kann die Lösung typischer Fehler direkt erläutert werden, sodass die Lernenden den Umgang mit diesen ebenfalls erlernen können.

Eine besondere Art von Lernmaterialien sind *Worked Examples*. Diese stellen die Lösung eines zuvor formulierten Problems dar und sollen vom Lernenden nachvollzogen werden. Worked Examples werden grundsätzlich als positiv für das Lernen bewertet [Swe06, Ren05]. Nach Sweller [SVMP98] enthalten sie bereits die Zustände, die im Prozess des Problemlösens erreicht werden müssen. Der Fokus läge daher eher auf den Schritten zwischen den Zuständen und entlastete damit das Arbeitsgedächtnis der Lernenden. Worked Examples für Programmierung sollten neben der Problemstellung eine kurze Analyse des Problems, den Lösungscode und Testläufe enthalten. Außerdem kann eine zusätzliche Visualisierung der wichtigsten Aspekte hilfreich sein [NB19]. Auch explizit für Codebeispiele in der Informatik liegt bereits Forschung insbesondere zur Einbettung dieser vor [SLR14]. Dabei sei es grundsätzlich förderlich, wenn Lösungen von ähnlichen Beispielen begleitet werden, die zum aktiven eigenen Lösen anregen. Insgesamt seien Worked Examples dabei entlastend nach der Cognitive Load Theorie, eine geeignete Einbettung in das Lehrkonzept jedoch von enormer Wichtigkeit [SLR14].

Damit existiert zwar bereits einige Forschung zu digitalen Lernmaterialien, durch die stetige technische Weiterentwicklung und die teilweise spezielleren fachlichen Anforderungen bestehen allerdings auch noch einige Unsicherheiten bei Designentscheidungen für digitales Lernmaterial. Diese sollten unter Anwendung der bereits bekannten Prinzipien weiter untersucht werden.

Aus den Erkenntnissen der allgemeinen Lehr-Lern-Forschung und der bisherigen informatikdidaktischen Forschung kann eine theoretische Basis für ein Modulkonzept abgeleitet werden. Einige Fallstudien geben zudem Hinweise auf konkrete Gestaltungsmöglichkeiten, die jedoch insbesondere in der praktischen Wirksamkeit in der Informatik bislang nicht ausreichend erforscht sind. Um zu weiteren Erkenntnissen beizutragen, wurde auf Basis der in diesem Kapitel beschriebenen Grundlagen (Praxis-)Forschung durchgeführt, deren Methodik und Rahmenkontext im folgenden Kapitel beschrieben wird.

3 Methodik

Dieses Kapitel erläutert den Kontext der durchgeführten Forschungsarbeiten und gibt einen Überblick über die Veröffentlichungen. Aufgrund der individuellen Schwerpunkte der Studien wurden jeweils für die Beantwortung der Forschungsfragen unterschiedliche Methodiken ausgewählt, die in der Übersicht kurz dargestellt werden. Insgesamt wurden sowohl qualitative als auch quantitative Auswertungen durchgeführt. Während bei den eingesetzten qualitativen Methoden in Form von mündlichen und schriftlichen Interviews sowie offenen Fragen in digitalen Fragebögen im Vordergrund stand konkrete inhaltliche Aspekte bezüglich der Meinungen von Studierenden und deren Ideen sowie Erfahrungen von Tutoren zu erfassen, fokussierten die eingesetzten quantitativen Verfahren eher empirische Erkenntnisse zum Erfolg der eingesetzten Konzepte, erfassten aber auch quantitativ Erkenntnisse über größere Gruppen an Studierenden. Eine ausführliche Beschreibung der Methodik für jede Studie findet sich in der jeweiligen Veröffentlichung. Eine Sammlung der ausgewerteten Daten befindet sich unter [Oss23].

3.1 Kontext der durchgeführten Studien

Die Forschungen fanden vollständig in einem studiengangübergreifenden Einführungsmodul der Informatik an der Universität Osnabrück zwischen 2019 und 2022 statt. Das Modul *Einführung in Algorithmen und Datenstrukturen*¹ wird in jedem Wintersemester angeboten und von etwa 400 bis 550 Studierenden unterschiedlicher Studiengänge besucht. Es ist verpflichtendes Grundmodul in allen informatiknahen Studiengängen, zu denen neben Informatik und Embedded Systems auch Cognitive Science und Anwendungswissenschaften wie Geoinformatik, Wirtschaftsinformatik oder (Umwelt-)Systemwissenschaften gehören. Auch für Studierende des Fachs Mathematik als fachwissenschaftlicher Bachelor oder im 2-Fächer-Bachelor als Lehramtsstudiengang für das Lehramt an Gymnasien ist der Besuch des Moduls vorgesehen. Studierende weiterer Fächer wie z. B. Physik können das Modul im Rahmen eines gewählten Nebenfachs ebenfalls absolvieren.

Die meisten Teilnehmenden belegen das Modul in ihrem ersten Semester, da es die Grundlage für weitere Informatikmodule bildet. Studierende der Mathematik wählen häufiger das dritte Semester für das Informatikmodul. Einen kleinen Anteil der Teilnehmenden bilden Studierende in Masterstudiengängen mit Informatikbezug und Anwendungswissenschaft, die zuvor einen auf das Anwendungsfach ausgelegten Bachelor absolviert haben und für die Zulassung zum Masterstudium Auflagen durch das Belegen von Informatikmodulen erfüllen müssen.

Durch den Status des Fachs Informatik als überwiegend Wahl- oder Wahlpflichtfach an Schulen [SHF21] sowie studiengangübergreifende Module ergibt sich eine starke

¹Da das Modul in diesem Fall nur aus einem Kurs besteht, werden die Begriffe Modul und Kurs im folgenden weitgehend synonym verwendet.

Heterogenität im Vorwissen sowie in der Motivation der Studierenden. Während Studierende der Informatik zwar neben dem Modul auch ein Modul der technischen Informatik sowie ein Mathematikgrundlagenmodul belegen müssen, fokussieren sie sich dennoch als Basis für weitere programmierbezogene Kurse stark auf *Einführung in Algorithmen und Datenstrukturen*. Studierende der Mathematik können dagegen häufig nicht nachvollziehen zu welchem Zweck sie ein Informatikmodul belegen müssen und konzentrieren sich überwiegend auf die parallel belegten etwa zwei Mathematikmodule, in denen sie in der Regel ebenfalls wöchentliche Abgaben leisten müssen. In Umfragen des Dozenten über das Audience-Response-System in der ersten Vorlesung gaben in den Jahren 2018 und 2019 12 bzw. 11 % an, dass sie nicht glauben, dass Programmieren und Algorithmen für ihr Berufsleben relevant sein werden. Weitere 19 bzw. 18 % wüssten nicht, ob die Inhalte für sie relevant sein werden ($n = 412$ bzw. 374). Auf dieser Grundlage kann davon ausgegangen werden, dass etwa ein Drittel der Studierenden trotz ihres Studiengangs keine fachliche Motivation für Informatik aufweist.

Auch die unterschiedlichen Vorkenntnisse spiegelten sich in den Umfragen über das Audience-Response-System in der ersten Vorlesung wieder. In den Jahren 2018 und 2019 gaben 42 bzw. 44 % an Informatik in keiner Form in der Schule gehabt zu haben. 7 bzw. 10 % haben dagegen einen Leistungskurs bzw. Kurs auf erhöhtem Niveau im Abitur belegt ($n = 375$ bzw. 385). 28 bzw. 25 % haben bisher noch nie programmiert, weitere 17 % in beiden Jahren nur im Vorkurs direkt vor Studienbeginn ($n = 413$ bzw. 373).

Die Veranstaltung bestand bis zum Wintersemester 2018/19 aus den drei Komponenten Vorlesung, Übung und Abgabenaufgaben. In der Vorlesung am Montag und Dienstag wurden die Inhalte grundlegend mit Fokus auf der theoretischen Basis und Herleitung vermittelt, während die Übung an vier verschiedenen, deckungsgleichen Terminen am Donnerstag sich mit der Vor- und Nachbereitung der Abgabenaufgaben beschäftigte. Diese bestanden aus diversen Anwendungen der Inhalte und mussten wöchentlich in Gruppen von zwei Studierenden bearbeitet werden. Die Lösungen wurden individuell mit einem studentischen Tutor in einer 30-minütigen Kurzprüfung mit dem Namen *Testat* zwischen Montag und Mittwoch diskutiert.

Inhaltlich deckt das Modul, mit dem die Studierenden neun ECTS erwerben, sowohl grundlegende Programmierung als auch Algorithmik und Datenstrukturen ab. Zu Beginn liegt der Schwerpunkt vor allem auf Elementaranweisungen in einer Programmiersprache, Kontrollstrukturen wie Bedingungen und Schleifen sowie auf der Darstellung einfacher Ausgaben. Im weiteren Verlauf werden Sortieralgorithmen, Laufzeiten und Datenstrukturen wie Arrays bzw. Vektoren, Listen, Stacks und Queues thematisiert. Abschließend werden auch Bäume und Graphen und darauf operierende Algorithmen wie z. B. der Dijkstra-Algorithmus zur Bestimmung der kürzesten Wege, erlernt.

Obwohl bereits einige Veranstaltungsteile im bisherigen Konzept früh Möglichkeiten wie z. B. Vorlesungsaufzeichnungen und eine hohe Betreuungsquote zur besseren Unterstützung der Studierenden nutzten, ergab sich durch mehrere Entwicklungen ein Bedarf nach einer Neukonzeption der Modulstruktur. Der Fokus dieser sollte nicht primär auf den Inhalten liegen, sondern auf dem Veranstaltungsangebot und dem Einbezug der Studierenden. Zum Wintersemester 2017/18 erfolgte aufgrund von Pensionierung ein Dozentenwechsel. Im den folgenden zwei Durchgängen wurden - mutmaßlich aufgrund der didaktischen Prägung der übernehmenden Arbeitsgruppe - zunehmend Praxisbeispiele,

die bisher in den Übungen diskutiert wurden, bereits in der Vorlesung thematisiert. Dadurch nahm die Menge der Inhalte der Übung ab, diese bestand daraufhin vor allem aus der Besprechung der Musterlösung der vorherigen Abgabenaufgaben sowie aus einem Vorrechnen von aufgabenbezogenen Beispielen. Wie auch bisher fand die Übung daher stark passiv statt. Studierende, die keine Schwierigkeiten mit der Lösung der Aufgaben gehabt haben, langweilten sich häufig und fragten an, ob ein zeitlicher Tausch mit den Beispielen für die neuen Aufgaben möglich sei. Auch für Studierende, die noch Schwierigkeiten mit dem Lösen der Aufgaben gehabt haben, war das Vorstellen der fertigen Lösung insbesondere bei Programmieraufgaben wenig hilfreich. Diesen fehlte häufig die Einsicht in den Lösungsprozess. Ein zeitlicher Tausch führte dann effektiv zu einer weiteren Verkürzung der Übung, da die Teilnehmenden überwiegend kein Interesse am zweiten Teil zeigten. Dies führte auch unter den Übungsleitenden zur Unzufriedenheit und dem Wunsch nach einem neuen Übungskonzept.

Parallel dazu erfolgte auch eine weitere Entwicklung, die sich auf das Fortschreiten der Technik und der damit verbundenen einfachen Verfügbarkeit dieser bezieht. Während bis etwa zum Beginn der 2010er Jahre noch viele Studierende zum Bearbeiten der Abgabenaufgaben die Rechner in den universitären CIP-Pools aufsuchten und dort automatisch auf Kommilitoninnen und Kommilitonen trafen, verfügten 2019 bereits 86% von 367 teilnehmenden Studierenden in einer Umfrage mittels Audience-Response-System in der ersten Vorlesung über ein eigenes Notebook. Um Übertragungsprobleme zu vermeiden, bevorzugten die meisten Studierenden das Lösen der Aufgaben auf ihren eigenen Endgeräten. Dafür war die physische Anwesenheit in der Universität nicht mehr nötig, allerdings führte dies auch zu einer Reduktion der Zusammenarbeit mit anderen Studierenden. Tutorinnen und Tutoren beobachteten daher zunehmend, dass sich Studierende in einer Abgabe- und Testatgruppe erst kurz vor dem Testat absprachen. Dadurch waren einzelne Studierende immer wieder nicht ausreichend über zumindest die Hälfte der Inhalte informiert.

Da außerdem wie in Kapitel 2 dargestellt kooperatives Lernen grundsätzlich positiv wirkt und mit dem Pair Programming auch ein informatikbezogenes Konzept existiert, dass dieses umgesetzt, sollte die Zusammenarbeit zwischen den Studierenden zumindest innerhalb der Gruppe gestärkt werden.

Die Modulgestaltung fand innerhalb eines dreijährigen Projekts zusammen mit anderen Arbeitsgruppen, die ebenfalls die Verbesserung der Lehre durch Flipped Classroom-Elemente und digitale Hilfsmittel anstrebten, statt. Für die Neugestaltung der Übung als *Coding Classes* konnte außerdem ein weiterer Mitarbeiter innerhalb der ein Jahr dauernden *LehrZeit* [Uni] gewonnen werden. Aufgrund von Vertragsfortsetzungen nach einer Elternzeit standen dadurch im ersten Semester des neuen Konzepts insgesamt fünf statt den üblichen zwei Mitarbeitenden für die Übungen zur Verfügung. Als nachhaltige Maßnahme wurden jedoch innerhalb dieser Zeit auch viele Aufgaben und Übungen entwickelt, die weiter genutzt werden können. Auch auf der Ebene der Tutoren, die durch angestellte Studierende besetzt werden, bestand ein Einsparbedarf. Obwohl weiterhin eine möglichst gute Betreuungsquote angestrebt war, sollten durch das Projekt auch Möglichkeiten gefunden werden die Anzahl an Tutorinnen und Tutoren zu reduzieren.

Im zweiten Projektjahr begann die Corona-Pandemie. Dadurch musste im Wintersemester 2020/21 vollständig digitale Lehre stattfinden und im Wintersemester 2021/22

waren nur kleinere Veranstaltungen wieder in Präsenz erlaubt. Unter Berücksichtigung der verfügbaren Räume hätte etwa ein Siebtel der regulären Teilnehmenden an einer Veranstaltung für die gesamten Teilnehmenden, wie z. B. einer Vorlesung, teilnehmen können. Für Übungen bzw. Coding Classes stand ein Raum zu Verfügung, in dem sich zeitgleich 20 Studierende unter Einhaltung der Richtlinien aufhalten konnten. Für die digitale Lehre stand das Videokonferenzsystem BigBlueButton zur Verfügung. Im ersten Jahr waren dabei Onlinekonferenzen mit bis zu etwa 300 Teilnehmenden ohne große Schwierigkeiten möglich. Veranstaltungen für eine höhere Anzahl teilnehmender Personen konnten über ein Streaming und einen angeschlossenen Chat realisiert werden. Im zweiten Jahr waren dann auch für sehr große Personenzahlen Onlinekonferenzen möglich. Außerdem wurde sowohl vor der Umgestaltung als auch im neuen Konzept StudIP [vir] für die grundsätzliche Lehrveranstaltungsorganisation genutzt.

Für Fragebögen, die im Rahmen der Evaluation des Modulkonzepts oder andere Studien eingesetzt wurden, wurde die Software von LimeSurvey [Lim] verwendet. Diese ermöglichte neben der Erstellung verschiedener Fragetypen für offene und geschlossene Fragen eine einfache digitale Bearbeitung, einen angemessenen Datenschutz und weitgehend störungsfreie Daten, da beispielsweise eine Antwort bei geschlossenen nur zwischen vorgegebenen Möglichkeiten erlaubt wurde und von vorherigen Antworten abhängige Fragen individuell ein- und ausgeblendet werden konnten.

Im Rahmen aller Datenerhebungen wurde stets auf die soweit mögliche Anonymität der Teilnehmenden hingewiesen. Auch wurde betont, dass die gegebenen Antworten keinerlei Auswirkungen auf die Behandlung in der Lehre und insbesondere auch auf die Prüfung und Benotung haben. Zur Teilnahme wurde sowohl inhaltlich durch die Möglichkeit zur Verbesserung der Lehre als auch teilweise extrinsisch, wie in den einzelnen Studien beschrieben, motiviert.

Die Auswertung quantitativer Daten erfolgte mit R [The] bzw. RStudio [Pos]. Als Signifikanzniveau wurde dabei $\alpha = 0,05$ (entsprechend [DB16, S. 665]) zugrunde gelegt.

3.2 Übersicht über Veröffentlichungen

Im Folgenden werden die Veröffentlichungen, zu denen diese Dissertationsschrift den Mantelbeitrag darstellt, kurz zusammengefasst und der Anteil an der Arbeit daran erläutert. Die Volltexte befinden sich im Anhang A.

Heterogenitätsbewusste digitale Gestaltung eines Einführungsmoduls in der Informatik [OBB⁺22]

In diesem Beitrag werden das Gesamtkonzept des Einführungsmoduls der Informatik und dessen Entwicklung über zwei Jahre beschrieben. Dabei wird vor allem auf die Bestandteile der aktiven Übung, die Entwicklung der Abgabenaufgaben und Zeiträume für die Bearbeitung sowie auf die digitalen Lernmaterialien eingegangen. Zentrale Fragestellungen für diese Arbeit waren die Konzeption eines Modulkonzepts, das für die heterogene, große Studierendengruppe anwendbar ist und zu einer guten Zufriedenheit sowie gutem

Lernerfolg führt. Methodisch wurde dabei ein Mixed-Methods-Ansatz gewählt. Im ersten Jahr wurden neben der Gesamtevaluation am Ende der Veranstaltung wöchentliche Kurzevaluationen in den Coding Classes durchgeführt und die studentischen Tutoren befragt. Im zweiten Jahr wurde auf die wöchentliche Befragung verzichtet, die beiden anderen Evaluationen jedoch weiter geführt. Während die Befragungen der Studierenden sowohl quantitative als auch qualitative Bestandteile enthielten, fokussierten sich die schriftlichen Befragungen und Gruppeninterviews mit den Tutoren vor allem auf qualitative Aspekte. Die Kapitel 5 und 6 erläutern den weiteren Verlauf im dritten Jahr sowie zugehörige Ergebnisse. Alle Autoren des Beitrags haben an der Konzeptentwicklung und Durchführung mitgewirkt, die Evaluation und Verschriftlichung der Ergebnisse wurden von der Autorin der Dissertation durchgeführt.

Cooperative Gamification in a Computer Science Introductory Module [OKHB21]

Der Gamificationansatz als ein weiterer Aspekt des Modulkonzepts wird in diesem Beitrag erläutert. Dabei stehen vor allem die Entwicklung des kooperativen Konzepts und die Überlegungen zu angemessenen Punktwerten im Vordergrund. Die Evaluation fand vor allem auf quantitativer Basis statt. Dazu wurden zum einen die Entwicklungen der gesammelten Punkte sowie die Art und die Erstellenden der Einträge ausgewertet, um eine Einschätzung zur praktischen Umsetzung entwickeln zu können. Zum anderen wurden die Studierenden innerhalb der Gesamtevaluation am Semesterende quantitativ, aber auch in einzelnen Fragen qualitativ nach ihren Erfahrungen mit den beschriebenen Elementen befragt. Alle Autoren des Beitrags haben an der Konzeptentwicklung und Durchführung mitgewirkt, die Evaluation und Verschriftlichung der Ergebnisse wurden von der Autorin der Dissertation durchgeführt.

Warum Onlinelehre sowie Open-Books-Klausuren und das Fach Informatik zusammenpassen [OBK21]

Dieser Diskussionsbeitrag erläutert fünf Thesen, die sich während der Reflexion zur Onlinelehre und den digitalen Open-Books-Klausuren entwickelt haben. Die Thesen behandeln dabei diverse Aspekte aus dem Fachkontext, die Onlinelehre und Open-Books-Klausuren begünstigen. Als zentrale Folgerung aus dem Beitrag ergibt sich der Wunsch für Lehrende der Informatik als besonders aktive Ausprobierende im Feld der digitalen Lehre, da diese sowohl inhaltlich als auch aufgrund der einfacheren Verfügbarkeit passender digitaler Werkzeuge häufig bessere Möglichkeiten dafür haben. Die drei Autoren haben den Beitrag nach der Idee von der Autorin der Dissertation in Zusammenarbeit entwickelt.

Comparing Video and Interactive Learning Material Styles for Programming [OB22]

In dieser Studie wurden fünf verschiedene Arten digitaler Lernmaterialien im Hinblick auf ihren Lerntransfererfolg und die Bewertung durch Studierende verglichen. Durch

diesen Vergleich sollte im Rahmen der Fallstudie ermittelt werden welchen Einfluss die Vermeidung von Fehlern und unterschiedliche Arten einen Programmcode als Lernmaterial aufzubereiten für Lernerfolg und Zufriedenheit haben. Für ein Anwendungsproblem aus den Graphalgorithmen wurden drei Videos und zwei interaktive Materialien erstellt, die sich im Produktionsaufwand und in der Darstellung des Prozesses unterschieden. Während zwei Videos und ein interaktives Material an Live Coding angelehnt den Programmierprozess und die Codeentwicklung zeigten, wurde in einem Video und einem interaktiven Material lediglich der vollendete Code als Worked Example erläutert. Die Teilnehmenden haben in der Studie zunächst ein zufälliges dieser Materialien erhalten. Nach der Bearbeitung sollten sie eine Aufgabe mit einem ähnlichen Graphproblem lösen und neben der Angabe ihres Lösungserfolges einen Fragebogen ausfüllen. Dadurch wurde zum einen quantitativ festgestellt, ob Unterschiede zwischen dem Lernen mit den verschiedenen Materialien im Bezug auf den Lerntransfer feststellbar sind und welche Materialien die Teilnehmenden grundsätzlich bevorzugen. Zum anderen konnten qualitativ weitere Aspekte festgestellt werden, die für die Studierenden wichtig sein könnten. Die Autorin der Dissertation hat das Konzept in Zusammenarbeit mit dem Zweitautor entwickelt und anschließend durchgeführt, ausgewertet sowie verschriftlicht.

Frustration Tolerance among Computer-Science-related Novice University Students (eingereicht)

Dieser Beitrag beschäftigt sich mit dem Verhältnis des psychologischen Begriffs der Frustrationstoleranz zum alltagssprachlichen und der Bedeutung von Frustrationstoleranz für die universitäre Einführungslehre in der Informatik. Dabei wurde die Frustration Discomfort Scale [Har05] übersetzt und von Studierenden im Einführungsmodul zu Beginn und zum Ende der Vorlesungszeit neben weiteren Fragen zur Selbsteinschätzung und demografischen Daten sowie einer Vorkenntniseinschätzung ausgefüllt. Diese wurden quantitativ ausgewertet, um die Forschungsfragen zu beantworten welchen Bezug der psychologische und der allgemeinsprachliche Begriff haben, ob die Teilnahme an einem Informatikmodul zu signifikanten Änderungen führt und welche Faktoren Auswirkungen auf die Frustrationstoleranz haben können. Die Autorin hat das Konzept in Zusammenarbeit mit dem Zweitautor entwickelt, bei der Durchführung Unterstützung durch den Zweitautor und eine weitere Person (Lara McDonald) zum Übersetzen des verwendeten Teilfragebogens gehabt und die Ergebnisse anschließend ausgewertet und verschriftlicht.

Teil II

Teil A: Voraussetzungen – heterogene Gruppen an Hochschulen

4 Studentische Erwartungen an ein studiengangübergreifendes Einführungsmodul der Informatik

Dieses Kapitel enthält eine unveröffentlichte Studie zu den Erwartungen der Studierenden an das Modul, in dessen Kontext die Forschung dieser Arbeit stattfand. Das Konzept wurde dabei von der Autorin der Dissertation in Zusammenarbeit mit Michael Brinkmeier entwickelt, die Durchführung wurde zusammen mit Laura Hembrock realisiert. Bei der Auswertung unterstützte Agnes Mainka die Prüfung der Reliabilität, ansonsten wurden Auswertung und Verschriftlichung von der Autorin der Dissertation durchgeführt.

In der Studie wurden ausgewählte Studierende interviewt, um deren Ansichten qualitativ explorieren zu können und daraus prototypische Erwartungen zu konstruieren, die bei der Gestaltung der Lehre berücksichtigt werden sollten. Dadurch können die Lehrangebote auf die Erwartungen der Studierenden eingehen und deren Zufriedenheit steigern.

4.1 Einleitung

Einstellungen und Erwartungen spielen aus psychologischer Sicht eine zentrale Rolle für das Handeln [Fea92]. Auf Grundlage der Bewertung von erwarteten Folgen einer Tat wird diese ausgeführt oder von einem solchen Verhalten abgesehen. In Bezug auf Studieninhalte, Module und Fächer haben so auch Studierende Erwartungen, die ihr Handeln innerhalb dieser Kontexte beeinflussen. Realistische Erwartungen führen außerdem zu einer höheren Zufriedenheit mit dem Studium [HSA13]. Besonders in großen Modulen mit einer hohen Heterogenität unter den Studierenden verfolgen diese unterschiedliche Ziele. Während für Studierende der Informatik das hier untersuchte Einführungsmodul zur Algorithmik und Programmierung die Basis für ihr weiteres Studium darstellt, ist es für einige Studierende anderer Fächer lediglich ein durch die Prüfungsordnung vorgesehener notwendiger Exkurs in ein Fach, dessen Bedeutung und Zweck für ihr Studium häufig nicht direkt ersichtlich ist. Obwohl einige quantitative Erkenntnisse über die Vorkenntnisse in solchen Modulen existieren und Lehrende eher geringe fachbezogene Vorstellungen bei Studierenden vermuten [VCEV15], gibt es bisher keine bekannte aktuelle Forschung innerhalb dieses fachlichen Kontextes zu Erwartungen qualitativer Art. Im Rahmen dieser Studie wurden daher Interviews mit Studierenden in einem solchen Modul geführt, um einen Einblick darüber zu erhalten, welche Vorstellungen zu den Inhalten fachlicher und überfachlicher Art die Befragten haben und welche Quellen und Ursachen dafür ihnen bewusst sind. Dabei wurde das Ziel verfolgt folgende Forschungsfragen explorativ beantworten zu können:

F1: Welche Erwartungen haben die Studierenden in der Anfangsphase des Informatiklernens an die Inhalte und den Lernerfolg im ersten Modul?

F2: Welche gefühlten Einstellungen haben Studierende in der Anfangsphase des Informatiklernens, wenn sie an dieses denken?

F3: Durch welche Quellen glauben Studierende bei ihren Einstellungen und Erwartungen beeinflusst worden zu sein?

4.2 Theoretischer Hintergrund

Vor dem Hintergrund, dass bisher keine bekannten qualitativ gewonnenen Erkenntnisse zu den Erwartungen und Einstellungen von Studierenden in der Anfangsphase des Informatiklernens vorliegen, wurde ein Modell ausgewählt, nach dem diese explorativ erfasst werden sollten. Zudem wurde auch Forschung aus anderen Fächern betrachtet, um dortige Erkenntnisse einbeziehen zu können.

4.2.1 Should- und Will-Expectations nach Boulding et al. [BKSZ93]

Vor dem Hintergrund des Marketings in den Wirtschaftswissenschaften entwickelten Boulding et al. 1993 [BKSZ93] ein dynamisches Stufenmodell für den Zusammenhang von Erwartungen, geliefertem und empfundenem Service, der sich wiederum auf die Handlungsintention auswirkt. Nach diesem Modell existieren bei einer Person Erwartungen, die nach *should* und *will* aufgeteilt werden. Als Soll-Erwartung wird dabei ein Wunsch bezeichnet. Die Wird-Erwartung besteht aus einer realistischeren Einschätzung dazu, was die Person glaubt tatsächlich erwarten zu können. Zusammen mit den tatsächlichen Angeboten führen diese Erwartungen zu einer Wahrnehmung des Services. Entspricht diese mindestens den Erwartungen, so ist garantiert, dass die Person ihre Erwartungen entsprechend anpasst und in einer nächsten Stufe den gelieferten Service vor dem Hintergrund angepasster Erwartungen betrachtet. Die Wahrnehmung des Services führt über eine Gesamtwahrnehmung der Situation dann auf die Intention zu handeln. Abbildung 4.1 zeigt das Wirkungsgefüge einer Stufe des Modells. Auf einen Kauf bezogen würden damit beispielsweise Erwartungen an den Preis oder die Qualität des Artikels existieren. In einer ersten Stufe erfolgen Anpassungen durch neue Informationen aus Quellen wie Bekannten oder Angeboten. Diese erzeugen dann eine möglicherweise realistischere Preisvorstellung, die nach ggf. weiteren Iterationen mit zusätzlichen Informationen und dem erneuten Abgleich mit den aktuellen Erwartungen - je nachdem, ob diese akzeptabel erfüllt werden - zum Kauf oder zum Verwerfen der Kaufabsicht führen. Im Gegensatz zum weit verbreiteten SERVQUAL [PZB88], das als standardisiertes Verfahren die wahrgenommene Qualität eines Angebots misst und dabei ebenfalls die Erwartungen der Kunden einbezieht, wird hier explizit zwischen unterschiedlichen Erwartungen bezüglich der Wünsche und der tatsächlichen Vorstellungen differenziert.

Bezogen auf die Qualität von Lehre lässt sich das Modell ebenfalls anwenden. Studierende werden von diversen Quellen beeinflusst, die sie mit Informationen zu einer Veranstaltung versorgen. Dies können sowohl Personen (z. B. Kommilitonen, Lehrende, Gesellschaft) als auch Medien (z. B. Zeitungen, Prüfungsordnung) sein. Mit diesen Informationen entstehen wieder angepasste Erwartungen, deren Erfüllung sich auf die Handlungsintention und damit das Studierverhalten auswirken kann. Die Zufriedenheit der Studierenden mit

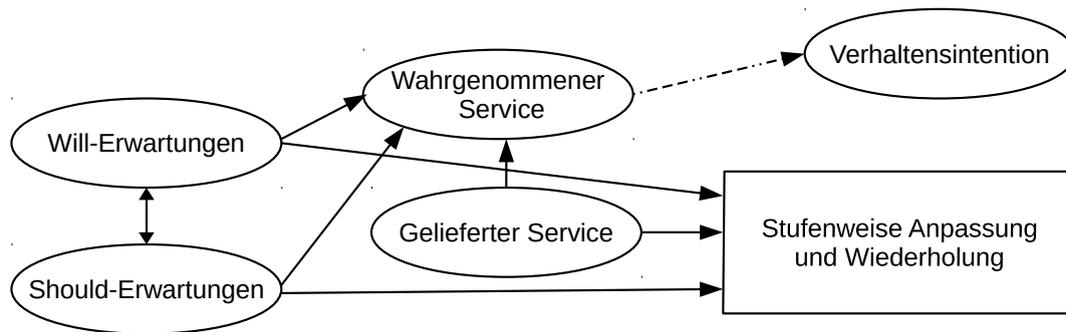


Abb. 4.1: Ausschnitt einer Stufe aus Modell nach Boulding et al. [BKSZ93], eigene Darstellung

dem Lehrangebot steht dabei im Zusammenhang mit ihren Leistungen [WLR97]. Im Rahmen der Theorie kann davon ausgegangen werden, dass Studierende vor allem dann so lernen wie im Konzept vorgesehen, wenn ihre Erwartungen auch zu den Erwartungen an sie selbst passen. Da bisher zwar einige serviceorientierte Modelle auf Lehrqualität im Allgemeinen übertragen wurden, aber noch keine fachdidaktische Forschung in der Informatik dazu existiert, sollte mit dieser Studie ein exploratives Erforschen der Erwartungen in einer frühen Stufe realisiert werden, auf das weitere Anwendungen aufbauen können. Entsprechend der Erkenntnisse können die Lehrenden die Erwartungen der Studierenden mit dem eigenen Angebot abgleichen und daraus schließen, ob diese zueinander kompatibel sind.

4.2.2 Bisherige Studienlage

Zum Zeitpunkt der Studie war keine direkt vergleichbare, aktuelle und qualitative Forschung im Rahmen der hochschulischen Informatik bekannt. Für die quantitative Einordnung der Häufigkeit bestimmter Einstellungen und Erwartung ist eine qualitative Voruntersuchung jedoch wichtig. Aus qualitativen Erkenntnissen können dann im Folgenden quantitative Messinstrumente entwickelt werden.

Eine qualitative Befragung von Studierenden informatikbezogener Studiengänge vor dem ersten Studienabschluss liegt lediglich zu deren Berufszielen vor [KGM18]. Für das Fach Physik existiert eine qualitative Untersuchung bezogen auf das Fähigkeitsselbstkonzept [WB19], wobei diese stark auf den Übergang zwischen Schule und Studium fokussiert ist. Für andere Fächer existieren jedoch teilweise Studien mit Studienanfängern zu Erwartungen oder fachlichen Vorerfahrungen quantitativer Art [KHBP14]. Zu Vorkenntnissen und damit Hinweisen zur Intention bei der Teilnahme an Vorkursen wurde im Fach Mathematik bereits quantitativ geforscht [AB18]. Im Rahmen der quantitativen Untersuchungen wurde insbesondere für Erwartungen und Einstellungen die Notwendigkeit weiterer qualitativer Forschung herausgestellt [KHBP14]. Eine weitere quantitative Studie, die auch Erwartungen an Studieninhalte berücksichtigt, wurde unter Studierenden der Sozialen Arbeit durchgeführt [Müh10]. Jedoch wird auch hier ein qualitatives Vorgehen als zusätzlich sinnvoll erachtet. Vorgefertigte Kategorien wie in diesem Fall können dazu führen, dass nicht die gesamte Bandbreite der Erwartungen innerhalb des Fragebogens erfasst werden kann. Im Rahmen der Informatik wurde bisher innerhalb der Forschung zu Voraussetzungen in der Studieneingangsphase nur quantitativ ein

Messinstrument zur Selbsteffizienz entwickelt [SPV20]. Dabei stellt diese jedoch nur einen Teil möglicher relevanter Voraussetzungen zu Studienbeginn dar. Eine qualitative Analyse von der Gesamtheit relevanter Faktoren fehlt dagegen.

Das in Abschnitt 4.2.1 beschriebene Modell wurde bereits im hochschulischen Lehrkontext angewandt [LM99]. Dabei stellte sich im Bezug auf die Erwartungen an die zum Studium gewählte Universität heraus, dass die Should-Expectations grundsätzlich höher und idealistischer sind als die realistischeren Will-Expectations. Außerdem konnten vor dem Kontext der Universitätswahl Unterschiede zwischen ethnischen Gruppen festgestellt werden. Der Einfluss von Vorkenntnissen und konkrete thematische Bezüge wurden nicht erfasst. In einem ähnlichen Szenario wurden mit qualitativen Interviews Studierende nach einem Jahr in ihrem Studium zu ihren Einstellungen dazu befragt [LADU19], wobei die Wahl der Hochschule ebenfalls relevant war. Ziel der Studie war allerdings eher hochschulstrategische Informationen zu gewinnen als fachbezogene Voraussetzungen zu explorieren.

In einer weiteren Studie [SLS15] wurden Studienanfänger zu Beginn und zum Ende eines CS1-Kurses quantitativ mit einem Fragebogen nach unterschiedlichen Einstellungen und Einschätzungen auch im Bezug auf ihre eigenen Kompetenzen zur Informatik befragt. Dabei wurden zwischen den beiden Zeitpunkten signifikante Veränderungen zwischen einzelnen Antworten festgestellt, die vor allem die eigenen Kompetenzen betrafen sowie die Möglichkeit in der Informatik kreativ zu sein und gute Lehrer gehabt zu haben. Dabei schätzten sich die Studierenden zum Kursende als kompetenter und weniger herausgefordert ein. Für viele Fragen bestanden signifikante Unterschiede zwischen männlichen und weiblichen Studierenden. Die weiblichen Studierenden trauten sich das Fach insgesamt weniger zu und schätzten sich als weniger kompetent im Programmieren ein. Während ethnische Gruppen im Untersuchungskontext einer deutschen Universität im Vergleich zu einer amerikanischen als weniger einflussreich einzustufen sind, könnten Zugehörigkeiten zu bestimmten Gruppen grundsätzlich Einfluss auf die Erwartungen und Einstellungen zum Fach haben und sollten daher bei der Erfassung und Auswertung berücksichtigt werden.

Um qualitativ und auf den konkreten Kontext ausgerichtet Erwartungen zum Modul analysieren zu können, wurden daher Interviews mit ausgewählten Studierenden durchgeführt. Die genaue Methodik wird im Folgenden beschrieben.

4.3 Methodik

Zur Erhebung der Erwartungen wurden zu Beginn eines Einführungsmoduls der Informatik Interviews mit Studierenden durchgeführt. Ziel war dabei eine explorative qualitative Erfassung der Erwartungen und Einstellungen zu diesem Modul.

4.3.1 Datenerhebung und Fragen

Die Datenerhebung erfolgte mittels Interviews über eine Onlinekonferenz mit einem Interviewer und einem Protokollanten. Da beide Personen auch an der Lehre im Modul

beteiligt waren und eine größtmögliche Sicherheit bezüglich der Anonymisierung garantieren wollten sowie primär die Aussageninhalte statt die Art der Äußerung von Bedeutung waren, wurde im Rahmen aktueller Sozialforschung [VF18] auf das klassische Transkribieren verzichtet. Auf Basis des in Abschnitt 4.2.1 erläuterten zugrundeliegenden Modells wurden dabei nach dem Erfassen des Studiengangs sowie thematisch passender Vorerfahrungen folgende Fragen gestellt:

- Welches Gefühl haben Sie grundsätzlich, wenn Sie an *Einführung in die Algorithmen und Datenstrukturen* denken? (F2)
- Sehen Sie *Einführung in die Algorithmen und Datenstrukturen* eher als Belastung oder als Bereicherung für Ihr Studium? (F2)
- Was möchten Sie in *Einführung in die Algorithmen und Datenstrukturen* lernen? (Should-Expectations, F1)
- Was glauben Sie werden Sie in *Einführung in die Algorithmen und Datenstrukturen* lernen? (Will-Expectations, F1)
- Welche konkreten Kompetenzen fachlicher und überfachlicher Art möchten Sie in *Einführung in die Algorithmen und Datenstrukturen* erwerben? (Should-Expectations, F1)
- Welche konkreten Kompetenzen fachlicher und überfachlicher Art glauben Sie werden Sie in *Einführung in die Algorithmen und Datenstrukturen* erwerben? (Will-Expectations, F1)
- Woher glauben Sie, haben Sie Ihre bisherigen Erwartungen? Welche Einflüsse könnten bei Ihren Antworten eine Rolle gehabt haben? (F3)

Dabei wurden bei Unsicherheit jeweils die Unterschiede zwischen den Should- und Will-Erwartungen sowie der Bezug zu den Kompetenzen erläutert. Zum Einstieg in das Interview wurde erneut nach dem Studiengang, Semester sowie nach Vorkenntnissen und deren Quelle gefragt. Die Interviews hatten eine Dauer zwischen vier und elf Minuten und wurden aufgrund der Organisation und Auswahl der Studierenden am Ende der zweiten sowie am Anfang der dritten Vorlesungswoche des Moduls durchgeführt. Zuvor wurden im Rahmen von Pretests mit drei Personen aus der Gruppe der Studierenden die Formulierungen der Fragen auf Verständnis und mögliche Mehrdeutigkeiten geprüft und entsprechend angepasst.

4.3.2 Auswahl der Interviewten

Die Interviews wurden mit 14 Studierenden des in Kapitel 3.1 beschriebenen Moduls durchgeführt. Zur Auswahl der Interviewten wurden zunächst alle Studierenden im Modul aufgefordert sich bei grundsätzlicher Interviewbereitschaft in einem Fragebogen einzutragen. Dabei wurden neben Geschlecht, Studiengang, Fach- und Gesamtsemesteranzahl und Alter auch die Vorerfahrungen abgefragt, wobei diese nach zwei Kategorien aufgeteilt waren. Zum einen sollte für eine Auswahl relevanter Themen eingeschätzt werden, ob und wie gut die Studierenden diese bereits kennen und zum anderen wurden

Pseudonym	Geschlecht	Studiengang (Fachsemester/ggf. Gesamtsemester)	Vorerfahrungen
FSIL	weiblich	2FB Lehramt Mathematik / Chemie (3)	keine
FSIM	weiblich	MSc Umweltsysteme und Ressourcenmanagement (3 / 11)	2 Jahre Unterricht in Mittelstufe, thematisch unpassend
FMeL1	weiblich	2FB Umweltsystemwissenschaften / Politikwissenschaften (1)	wenig Informatikunterricht in Mittelstufe, kaum Vorwissen
FMeL2	weiblich	BSc Cognitive Science (1)	keine
FMeM	weiblich	BSc Cognitive Science (1)	3 Monate Schulunterricht, eigenes Hobbyprojekt
FStL	weiblich	Bachelor Berufl. Bildung Kosmetologie / Informatik (1 / 7)	keine
FStM	weiblich	BSc Informatik (1)	Kurs an VHS, Arbeit im IT-Fachgeschäft
MSIL	männlich	2FB Lehramt Mathematik / Sport (3)	keine
MMeM1	männlich	MSc Geoinformatik (3 / 9)	Einsatz eines Skripts in BA
MMeM2	männlich	BSc Wirtschaftsinformatik (1)	Leistungskurs
MMeM3	männlich	BSc Cognitive Science (1)	Leistungskurs am berufl. Gymn.
MMeH	männlich	BSc Physik (3)	5 Jahre Schulunterricht
MStH1	männlich	BSc Embedded Systems (1)	Fachinformatiker-Ausbildung, Hobbyprojekte zur Schulzeit
MStH2	männlich	BSc Informatik (1)	Leistungskurs, Fachinformatiker-Ausbildung, 7 Jahre Programmiererfahrung

Tabelle 4.1: Übersicht der interviewten Studierenden

konkrete Vorerfahrungen mit Informatik aus schulischen und weiteren Bereichen abgefragt. Aus 134 grundsätzlich interessierten Studierenden wurden 13 mit unvollständigem Fragebogen sowie weitere 18 Studierende aussortiert, die bereits einmal das betreffende Modul oder ein vergleichbares an einer anderen Universität belegt hatten und damit

bereits konkrete Erfahrungen im Modul selbst gemacht hatten, die ihre Erwartungen beeinflussen. Die übrigen 103 Studierenden wurden im ersten Schritt nach Studiengängen und im zweiten nach Vorerfahrungsscore sortiert.

Die Auswahl verfolgte das Ziel in Studiengängen mit vielen Teilnehmenden möglichst unterschiedliche Vorerfahrungen einzubeziehen und in kleineren Studiengängen eine dafür möglichst typische Vorerfahrung abzubilden. Innerhalb der kleineren Studiengänge wurden einige thematisch zusammengefasst, insbesondere bei wenig Anmeldungen. Um insgesamt aber eine möglichst hohe Bandbreite an unterschiedlichen Studierenden interviewen zu können, wurden die Studierenden nicht exakt proportional zur Anzahl der Studierenden in ihrem Studiengang ausgewählt. So wurden für die beiden besonders stark vertretenen Studiengänge B. Sc. Informatik sowie B. Sc. Cognitive Science jeweils drei Studierende ausgewählt, wobei jeweils ein Interviewpartner mit einem niedrigen, einem mittleren und einem hohen Vorerfahrungsscore aus dem Spektrum dieser Studiengänge ausgewählt wurde. Von den anderen Studiengängen wurden lediglich für die ebenfalls höher vertretene Gruppe der Mathematik-Lehramtsstudierenden zwei Interviewpartner gewählt. Da hier nur ein Spektrum von niedrigen und mittleren Vorerfahrungsscores zu finden war, wurden die beiden Studierenden auch in diesen Bereichen gesucht. Insgesamt wurden so 16 Studierende sowie jeweils bis zu zwei Nachrückende mit ähnlichem Profil ausgewählt, von denen 14 Studierende inkl. der Nachrückenden tatsächlich einen Termin für ein Interview wählten und wahrnahmen.

4.4 Ergebnisse

Mit Hilfe von qualitativer Inhaltsanalyse sowie anschließend quantitativer Betrachtung relevanter Aspekte und einer Typenbildung konnte ein Eindruck über die Erwartungen der Studierenden gewonnen werden.

4.4.1 Kategorien

Im Rahmen der qualitativen Inhaltsanalyse konnten zunächst deduktiv sechs Oberkategorien anhand der Fragen und zu diesen induktiv jeweils bis zu zehn Unterkategorien und teils weiterer Spezifizierung herausgebildet werden, die in der Übersicht 4.2 dargestellt sind.

Die höchste Anzahl an Aussagen (129) wurde der Kategorie *Erwartungen fachlicher Art* zugeordnet. Zur Kategorie *Quellen/Einflussfaktoren* konnten 104 Aussagen zugeordnet werden. Die anderen Kategorien waren - teilweise auch aufgrund der eingeschränkteren Bandbreite - seltener vertreten. Elf der 14 Teilnehmenden wiesen eine mindestens partielle, bewusste Deckungsgleichheit von Should- und Will-Expectations (5.1) auf. Eine Person bezieht die im Modul verwendete Programmiersprache explizit zur Unterscheidung ihrer Vorstellungen ein, andere erkennen abweichende Should-Expectations als teilweise nicht realistisch an. Unter den fachlichen Erwartungen überwiegt die Nennung tatsächlich relevanter Themen mit 48 Aussagen gegenüber nicht relevanten Themen (18). Nur eine Aussage ist als inkorrekt einzustufen. Bei dieser gab die interviewte Person an, dass die Fähigkeiten in einer Programmiersprache keine Auswirkungen auf allgemeine

1 Modulbezogene Gefühle	
1.1 positiv	
1.2 negativ	
1.3 Schwierigkeiten mit der Frage	
1.3.a Frage ausweichen	
1.3.b Schwierigkeiten beim Benennen	
1.4 differenzierte Vorstellungen	
1.4.a Zeitmanagement/Arbeitsbelastung	
1.4.b Schwierigkeit	
1.5 Wertung von Erlebtem	
1.5.a deutliche Vergangenheit	
1.5.b nähere Vergangenheit	
2 Erwartungen fachlicher Art	
2.1 Modulname	
2.2 tatsächlich relevante Themen	
2.2.a Programmierkompetenzen	
2.2.b Algorithmmikkompetenzen	
2.2.c konkreter Sach-/Problem-/Programmiersprachenbezug	
2.3 nicht relevante Themen	
2.3.a niedrigere Abstraktionsebenen (technische Informatik)	
2.3.b im thematisch korrekten Bereich (über das Modul hinausgehend)	
2.3.c allgemein digitale Kompetenzen	
2.3.d Themen anderer Informatikmodule	
2.4 korrekte Aussagen	
2.5 inkorrekte Aussagen	
2.6 Bezug zum Studienfach/Beruf	
2.6.a Bezug zum Studiumsverlauf	
2.7 überfachliche Kompetenzen	
2.8 Bezug zum Alltag/Digitalisierung	
2.9 Explorative Erwartungen	
2.10 Schwierigkeiten beim Benennen	
2.10.a Allgemeines	
2.11 Bezug erste 2 Wochen	
3 Erwartungen überfachlicher Art	
3.1 tatsächlich relevante Kompetenzen	
3.1.a soziale Kompetenzen	
3.1.b Problemlösekompetenzen	
3.2 Schwierigkeiten beim Benennen	
3.2.a Allgemeines	
4 Quellen/Einflussfaktoren	
4.1 Bildung, Schule	
4.1.a Ausbildung	
4.1.b Praktika, Nebenjobs	
4.2 Medien und allgemeines Bild	
4.3 Familie/Freunde, gleiche Generation	
4.4 Kommilitonen, die Modul bereits belegt hatten	
4.5 Personen aus anderen Studiengängen	
4.6 Vorkurs	
4.7 bisherige 2 Wochen	
4.8 sonstige Quellen	
4.9 nicht eindeutig benennbar	
4.10 Fachliche Rahmung	
4.10.a Institutionelle Rahmung	
4.11 Hochschulinformationstag	
4.12 bisheriges Studium	
5 Deckungsgleichheit	
5.1 Should & Will	
5.2 freie Frage/Kompetenzen	
6 Sonstiges	
6.1 Beurteilung eigener Erwartungen	
6.1.b als unrealistisch im Bezug auf Modul	

Abb. 4.2: Aus den Interviewantworten abgeleitete Kategorien

Kompetenzen hätten und bezweifelte damit die Sinnhaftigkeit diese zu lernen. Zwölf Aussagen nehmen Bezug auf den Modulnamen. Zur Überprüfung der Reliabilität wurden drei der 14 Interviews noch von einer zweiten Person kategorisiert, was 21,43 % des Materials entspricht. Dabei konnte eine Inter-Rater-Reliabilität von $\kappa = 0,55$ (Cohens Kappa) festgestellt werden, was als akzeptabel anzusehen ist [LK77, DB16, S. 346].

4.4.2 Festgestellte Typen

Aus den Interviews konnten anhand der Einordnung in die Kategorien sowie der Gesamtbetrachtung der Interviews sechs Typen bezüglich der Erwartungen herausgebildet werden, die in zwei Kategorien aufgeteilt werden können. Dabei unterscheiden sich diejenigen mit Vorwissen und/oder starkem Interesse am Modul, häufig Studierende in Studiengängen mit eher hohem Informatikbezug, von Studierenden, für die weniger Informatik im Studium vorgesehen ist.

Kategorie 1: Vorwissen und/oder starkes Interesse

Der interessierte und offene Programmierer hat ein realistisches Bild vom Modul, ist über die Inhalte und intendierten Kompetenzen informiert und offen für das Modul. (Eine Person: MMeH)

Der Programmierausbauer verfügt bereits über Vorkenntnisse und möchte vor allem die Programmierkompetenzen weiter ausbauen, ist dabei insgesamt positiv gestimmt und vorrangig praktisch orientiert. Einige Programmierausbauer zeigen Tendenzen einer allgemein informatikinteressierten Person, die unter „echter“ Informatik Algorithmik versteht und dies als Steigerung der Programmierkenntnisse sieht. (Drei Personen: FStM, FMeM, MMeM1)

Der euphorische Systematisierer hat bereits einige Programmierkenntnisse, möchte diese nun jedoch systematisieren und mit überfachlichen Kompetenzen anreichern. Auch diese Personen sind durchweg positiv gestimmt. (Drei Personen: MStH1, MStH2, MMeM3)

Programmierausbauer sowie Systematisierer können als Untertypen der interessierten und offenen Programmierer betrachtet werden.

Kategorie 2: Bezugsinformatiker

Der Digitalisierungsfan möchte vor allem deshalb Grundkenntnisse in der Informatik erwerben, um der fortschreitenden Digitalisierung folgen zu können, hat dabei aber kaum konkrete Vorstellungen, welche Kenntnisse dafür benötigt werden. (Eine Person: FMeL2)

Der Alltagsinformatiker und explorative Realist möchte vor allem für den Alltag nützliche Kenntnisse erwerben, gleichzeitig aber unter Berücksichtigung seiner Zeitkapazität einen möglichst breiten Einblick in das Fach bekommen. Alltagsnützliche Kenntnisse können dabei Kenntnisse in der Reparatur von genutzten technischen Geräten, in der allgemeinen Nutzung von Software oder berufsbezogen nützliche Kenntnisse sein. Ein Teil dieser Personen erkennt jedoch, dass ein studiengangübergreifendes Modul keine berufsspezifischen Kompetenzen für alle Studiengänge vermitteln kann. (Vier Personen: FMeL1, MMeM2, FStL, MSIL)

Der ängstliche Programmiersprachenbezogene bezieht die erwarteten Kompetenzen strikt auf die zu lernende Programmiersprache, sieht dabei kaum die tatsächlichen Kompetenzen und befürchtet vor diesem Hintergrund einen hohen Aufwand sowie ein mögliches Scheitern. (Zwei Personen: FSIM, FSIL)

Insgesamt sind die Erwartungen selten eindeutig falsch, ein höheres Vorwissen sorgt aber in der Regel für detailliertere thematische Erwartungen. Fast alle Interviewten sind zum Modul grundsätzlich eher positiv eingestellt, wobei einige Befürchtungen zu einem hohen zeitlichen Aufwand äußern. Kein Studierender stellte im Interview grundsätzlich den Sinn des Moduls für das persönliche Studium in Frage. Einige erwarten auch einen Schwierigkeitsgrad, den sie als problematisch einschätzen.

4.4.3 Einflüsse auf Einstellungen und Erwartungen

Zur Frage nach bewussten Einflüssen auf die Einstellungen und Erwartungen wurden am häufigsten die Erfahrungen der bisherigen zwei Wochen genannt (acht Aussagen). Vier weitere Aussagen bezogen sich auf den Vorkurs, der wenige Wochen vor dem Vorlesungsbeginn stattgefunden hat. In fünf Aussagen wurde vorherige außeruniversitäre Bildung (z. B. Schule, Ausbildung) erwähnt. Auch andere Personen im Umfeld der Studierenden werden als Einflüsse genannt. Am häufigsten wurden von vier Personen Mitstudierende erwähnt, die das Modul bereits in einem vorherigen Semester belegt hatten. Vereinzelt wurden Familie, Freunde sowie Studierende anderer Studiengänge genannt. Drei Aussagen beziehen sich auf die Vorstellung der fachlichen oder institutionellen Rahmung. So erwartet ein Studierender einen hohen fachlichen Anspruch im Modul mit der Begründung, dass die universitäre Lehre sich von schulischer deutlich unterscheiden sollte. Drei Befragte hatten Schwierigkeiten damit Einflüsse zu benennen und zwei Aussagen beziehen sich auf das Bild der Informatik in den Medien.

4.5 Diskussion

Zunächst sollten bei der Interpretation der Ergebnisse einige Limitierungen beachtet werden. Die insgesamt eher positive Stimmung ist sicherlich zu einem Teil auch dadurch bedingt, dass sich grundsätzlich bereits eher positiv eingestellte und motivierte Studierende für ein solches Interview melden. Die negativ oder belastend empfundenen Aspekte bezogen sich vor allem auf die Schwierigkeit und die zeitliche Belastung bzw. die Befürchtung eines hohen Arbeitsaufwands. Dies sollte vor dem Hintergrund des Drucks durch die Regelstudienzeit [Mey10, GBHW12, SM11] und damit verbundene finanzielle Schwierigkeiten betrachtet werden. Grundsätzlich steht es insbesondere Studierenden mit Informatik als Nebenfach frei weniger Module als vorgesehen im empfohlenen Semester zu belegen. Angesichts der damit verbundenen möglichen Verzögerungen und des entsprechenden Anstiegs der Lernlastung in höheren Semestern kann dies ebenfalls zu einem höheren Stressempfinden führen. In der Tat berichten Studierende auch nach Abschluss des Moduls oft, dass der zeitliche Aufwand höher war als sie für angemessen hielten. In manchen Studiengängen sind Studierende es jedoch nicht gewohnt auch während der Vorlesungszeit regelmäßig Leistungsnachweise erbringen zu müssen. Durch die Heterogenität des Vorwissens sowie aufgrund von unvorhersehbaren Schwierigkeiten bei Programmieraufgaben kann es jedoch vorkommen, dass Studierende mehr Zeit für die Aufgaben benötigen als vorgesehen. Diese Erfahrungen scheinen sich schnell über die Jahrgänge der Studierenden zu verbreiten.

Bei den weiteren Einflussfaktoren wird deutlich, dass Studierende grundsätzlich noch offen für neue Eindrücke sind, da die meisten angeben bereits aus den ersten zwei Wochen der Vorlesung sowie dem Vorkurs Erwartungen an das Modul generiert zu haben. Dies könnte ein möglicher Grund für das Fehlen eindeutig falscher Erwartungen sein. Auch bisherige Bildung im informatischen Bereich hat einen starken Einfluss, der sich sowohl auf die den Studierenden bewussten Quellen als auch auf das tatsächliche Vorwissen auswirkt. Zur seltenen Nennung der Medien als Quelle lässt sich vermuten, dass diese eher unterbewusst einen Einfluss auf die Erwartungen der Studierenden besitzen. Dies lässt sich durch Aussagen zu erwarteten Inhalten stützen, da hier durch Medien verbundene Inhalte genannt wurden (Lernspiele für den Unterricht, allgemeine digitale Kompetenzen, Reparaturkenntnisse). Durch die relativ hohe Übereinstimmung bei Should- und Will-Expectations zeigt sich, dass an dieser Stelle nicht pauschal erhöhte Wunschanforderungen existieren. Jedoch wurden stellenweise Wünsche geäußert, die von den Studierenden selbst als nicht realistisch im Bezug auf das Modul bewertet wurden. Der häufig erwähnte Einfluss der bisherigen Vorlesungszeit zeigt jedoch eine grundsätzliche Kompatibilität zum Modell nach Boulding [BKSZ93]. Die Studierenden konnten diese neuen Informationen bereits nutzen, um ihre Erwartungen anzupassen.

4.6 Folgerungen

Die Ergebnisse zeigen, dass an ein Einführungsmodul der Informatik einige Erwartungen existieren, die sowohl fachlicher als auch organisatorischer Art sind. Dabei haben die Studierenden mehrheitlich entweder überwiegend passende oder fehlende Einschätzungen zu den thematischen Inhalten der Veranstaltung, aber eine meist positive Grundeinstellung. Lediglich zur zeitlichen Belastung und vereinzelt zum Schwierigkeitsniveau existieren Sorgen. Als Einflussfaktoren für die Erwartungen sind den Studierenden vor allem eigene Erfahrungen vom Beginn des Moduls, aber auch Vorkenntnisse aus früherer Bildung sowie durch Kommilitonen, die das Modul bereits belegt hatten, bewusst. Vor diesem Hintergrund ist eine dauerhafte Verankerung des Informatikunterrichts an allgemeinbildenden Schulen von Bedeutung, da hier in einem geschützten Rahmen ein erster Kontakt zur Informatik erfolgen kann. Dadurch könnten bereits im Vorfeld realistische Erwartungen an einen universitären Zugang zur Informatik aufgebaut werden und Befürchtungen bezüglich der eigenen Fähigkeiten für das angedachte Niveau vermieden werden. In diesem Kontext können auch bereits Schüler dafür sensibilisiert werden, welche Bereiche und Kompetenzen die Informatik umfasst. Zusätzlich sollten Studierende in Studiengängen mit niedrigem Informatikbezug, wie z. B. Lehramtsstudierende im Fach Mathematik, darüber informiert werden, wieso auch für sie das Absolvieren eines Informatikmoduls sinnvoll ist und welche dabei intendierte Kompetenzen für ihren weiteren Werdegang von Bedeutung sind. Idealerweise sollte sogar eine Individualisierung des Moduls nach Studiengang stattfinden, jedoch ist dies aufgrund des Personalbedarfs dafür nicht realistisch. Überraschend ist, dass kein Studierender eindeutig falsche Vorstellungen geäußert hat. Hier bleibt zu ergründen, ob ein solcher Typ Studierender tatsächlich nicht existiert, sich nicht zum Interview gemeldet hat oder ob der Einfluss der ersten zwei Wochen bereits zuvor existierende falsche Vorstellungen beeinflusst hat. Zu diesen qualitativ explorativen Ergebnissen bietet sich besonders zur Festigung der gefunden

Typen und zur Konkretisierung der Verhältnisse eine Fortsetzung quantitativer Art an. Dabei sollten Messinstrumente entwickelt werden, anhand derer die Verteilung der gefundenen Typen in der Gesamtheit besser eingeschätzt werden kann. In Bezug auf das verwendete Modell könnte außerdem auch erfasst werden, welche Faktoren und Quellen genau auf die einzelnen Vorstellungen wirken.

Teil III

Teil B: Praktische Gestaltung von Einführungsmodulen in der Informatik unter Berücksichtigung von Heterogenität und Digitalisierung

5 Modulkonzept

Den Kern der Promotionsarbeit bildet die Umstrukturierung eines Einführungsmoduls der Informatik an der Universität Osnabrück. Dabei wurde über mehrere Jahre iterativ ein Modulkonzept auf Basis des Flipped Classroom und weiterer in Kapitel 2 beschriebener Grundlagen entwickelt und kontinuierlich mit einem Mixed-Methods-Ansatz quantitativ und qualitativ auf verschiedenen Ebenen evaluiert. Eine Beschreibung der Voraussetzungen, des ersten und des weiterentwickelten Konzepts sowie die Evaluationsergebnisse der ersten zwei Durchführungen wurden in [OBB⁺22] publiziert.

Die Veranstaltung *Einführung in Algorithmen und Datenstrukturen* besteht nach Modulbeschreibung [Fac20] aus den drei wesentlichen Bausteinen der Vorlesung mit vier Semesterwochenstunden, der Übung mit zwei Semesterwochenstunden und der „erfolgreiche[n] Bearbeitung der Übungsblätter“ [Fac20, S. 12]. Inhaltlich liegt der Fokus auf Algorithmen wie Suchen, Sortieren und auf Datenstrukturen wie abstrakten Datentypen arbeitenden Algorithmen sowie auf Eigenschaften von Programmen wie z. B. der Effizienz [Fac20]. Während die Vorlesung für die gesamte Gruppe der Studierenden mit bis zu 550 Personen jeweils montags und dienstags zwischen 14 und 16 Uhr stattfand, wurde die Übung zu vier Terminen donnerstags jeweils um 8, 10, 12 und 14 Uhr durchgeführt. Die erfolgreiche Teilnahme am Übungsbetrieb wurde durch eine ausreichende Anzahl bestandener Testate erreicht. Dabei handelt es sich um etwa 30-minütige mündliche Kurzprüfungen, in denen eine Testatgruppe einem studentischen Tutor die eingereichten Lösungen zu einem Übungsblatt erläutert und Fragen zu diesen beantwortet. Eine Beratungszeit, bei der studentische Tutoren zu bestimmten Zeiten für individuelle Fragen ansprechbar waren, ergänzte die Lehrveranstaltungen.

Die Motivation zur Umgestaltung des Moduls begründete sich auf mehreren Ebenen, die im Folgenden erläutert werden. Obwohl das Modul bereits seit vielen Jahren über die Basislehre hinausgehende Angebote, wie beispielsweise eine Vorlesungsaufzeichnung, zur Verfügung gestellt hat und durch Beratungszeiten mit studentischen Tutoren auch individuelle Lerngelegenheiten gegeben wurden, führten ein Dozentenwechsel und einige technische Entwicklungen zum Wunsch die Lehre anhand aktueller Forschung neu und innovativ zu gestalten. Vermutlich aufgrund der von den Lehrenden wahrgenommenen breiten Verfügbarkeit von portablen Arbeitsgeräten wie Notebooks bei Studierenden wurde die bisher häufige Arbeitsweise in den Computerräumen der Universität zunehmend seltener beobachtet (siehe auch Kapitel 3.1). Das Bearbeiten der Aufgaben in den Computerräumen trug dazu bei, dass Studierende, meistens innerhalb einer Testatgruppe, gemeinsam mit Kommilitonen die Aufgaben bearbeitet haben. Dabei kommunizierten sie häufig miteinander und halfen sich gegenseitig bei Schwierigkeiten. Durch die einfachere Verfügbarkeit von privaten Computern verlagerte sich die Arbeit an den Aufgaben allerdings zunehmend in den außeruniversitären Raum. Die Studierenden mussten zwar im Testat die Lösungen auf einem universitären Rechner präsentieren, konnten ihre Abgabe jedoch online auf diese übertragen. Folglich wurde zunehmend beobachtet, dass

die Kommunikation der Studierenden kaum noch stattfand und die gemeinsame Abgabe durch die Aufteilung der Aufgaben gelöst wurde. Eine kurze Absprache der zu diesem Zeitpunkt aus zwei Studierenden bestehenden Testatgruppen fand häufig erst unmittelbar vor der Aufgabenpräsentation statt. Mit weiterer Verbreitung portabler Computer und der Bitte vieler Studierender ihre Lösungen auf ihren eigenen Geräten zeigen zu dürfen, bestand nach einigen weiteren Jahren kein Grund mehr dafür die universitären Rechner zu nutzen. Das Problem der mangelnden Zusammenarbeit zwischen den Studierenden existierte jedoch weiterhin. Ein Ziel der Umstrukturierung bestand folglich darin die soziale Komponente des gemeinsamen Programmierens zu fördern und die Studierenden dazu zu motivieren zusammen an den Aufgaben zu arbeiten. Dabei wurde auch die berufliche Realität als Hintergrund gesehen, da in vielen Bereichen, in denen die Studierenden nach ihrem Studium arbeiten können, Kommunikation ein zentraler Aspekt ist.

Viele Studierende hatten grundsätzlich Schwierigkeiten mit dem Programmieren, das als schwierige Tätigkeit eingeordnet wird [Jen02, RRR03]. Insbesondere die eigentlich auf die Praxis ausgerichtete Übung wurde dabei von vielen Studierenden allerdings als wenig hilfreich empfunden. Zum einen wurden nach dem Dozentenwechsel in der Vorlesung die Inhalte der Übung geringer, da viele praktische Beispiele bereits in der Vorlesung thematisiert wurden. Zum anderen wünschten sich viele Studierende, dass die Vorstellung der Aufgabenlösungen der vorherigen Woche zum Ende der Übung erläutert werden, um die Übung vorher verlassen zu können. Dies betraf vor allem Studierende, die die Aufgaben erfolgreich gelöst hatten und daher kein Interesse an einer Wiederholung hatten. Aber auch Studierende, die Schwierigkeiten mit dem Lösen der Aufgaben hatten, profitierten kaum von der Vorstellung des fertigen Programms. Sie konnten zwar häufig nachvollziehen, wieso der vorgestellte Code das gegebene Problem lösen kann, jedoch fehlte der Einblick in den Prozess des Programmierens selbst. Dadurch hatten sie bei der nächsten Aufgabe erneut große Hürden zu bewältigen, um selbstständig eine Lösung zu entwickeln. Aus diesem Grund sollte die Übungszeit stärker dazu genutzt werden, um die Studierenden zu aktivieren und die Prozesse beim Programmieren sichtbar zu machen. Außerdem sollte Hilfe direkt beim Problem und zeitlich synchron ermöglicht werden.

Eine weitere Beobachtung, die ebenfalls Relevanz für die Umgestaltung hatte, besteht in der Vor- und Nachbereitung der Veranstaltungen durch Studierende. Besonders in den ersten Studiensemestern fiel vielen Studierenden dabei der Wechsel aus der Schule mit Anwesenheitspflicht und klar festgelegten Hausaufgaben zur Universität mit dem freiwilligen Besuch von Veranstaltungen und der Erwartung der Vor- und Nachbereitung dieser ohne explizite Prüfung schwer. Einige Studierende hatten trotz Anwesenheit in der Vorlesung, vermutlich aufgrund der hohen Passivität des Formats, keinen ausreichenden Lernerfolg und dadurch Schwierigkeiten beim Bearbeiten der Aufgaben. Als begleitende Materialien zur Vorlesungen wurden den Studierenden ein Skript sowie Vorlesungsfolien zur Verfügung gestellt. Es existierten aber keine weiteren Materialien, an denen die eher theoretischen Inhalte der Vorlesung konkret angewendet und das eigene Verständnis geprüft werden konnten. Somit mussten die verpflichtenden Aufgaben zur Abgabe direkt auf Basis des theoretischen Wissens bearbeitet werden. Obwohl viele Studierende über den hohen Zeitaufwand des Moduls klagten, zeigen die ZeitLast-Studie [SM11] und eigene Untersuchungen mit der Methode des Zeitbudgetprotokolls, dass viele Studierende nicht

die vorgesehene Zeit für die Vor- und Nachbereitung der synchronen Veranstaltungsteile aufwenden. Eine Hypothese der Lehrenden im umgestalteten Modul war dabei, dass die Studierenden zwar die Bereitschaft zu einer weiteren Auseinandersetzung mit den Inhalten hätten, allerdings aufgrund bisher geringer Studienerfahrung nicht wissen würden wie sie außerhalb der Veranstaltungen lernen sollten. Daher sollten im Rahmen des neuen Konzepts Materialien entstehen, die das Selbstlernen und insbesondere die Vor- und Nachbereitung der Präsenzveranstaltung unterstützen könnten. Diese sollten gleichzeitig der Heterogenität im Modul (siehe 3.1) gerecht werden, indem sowohl auf unterschiedliche Vorkenntnisse eingegangen als auch zeitliche und räumliche Flexibilität ermöglicht wird.

Zusätzlich sollten die Aufgaben, die weiterhin verpflichtend abgegeben werden müssen, offener und umfangreicher werden, um die Motivation und Kreativität der Studierenden zu fördern. Ein höherer Umfang erlaubt dabei auch eine stärkere Realitätsnähe, da größere Projekte und Anwendungen ermöglicht werden. Zur Kreativität sollte ein graphischer Ansatz verfolgt werden, wie dieser in der Literatur häufig mit projektartigen Aufgaben verbunden wird (siehe 2.1.2).

Als Rahmenbedingung bestand außerdem die Beschränkung des Personalaufwands. Obwohl insbesondere durch Studienqualitätsmittel eine verhältnismäßig hohe Zahl von bis zu ca. 25 studentischen Tutoren für die Betreuung der etwa 550 Studierenden ermöglicht wurde, bestand seitens des Instituts ein Interesse daran den Personalaufwand dauerhaft zu reduzieren. Auf Ebene der Übungsleitenden konnte die Umgestaltung zwar personell unter anderem durch zwei Projekte wesentlich unterstützt werden, jedoch sollte ein Konzept entstehen, das dauerhaft mit dem regulär verfügbaren Personal sinnvoll durchführbar ist.

Insgesamt wurde dabei ein Konzept angestrebt, das zumindest Teile des Flipped Classrooms (siehe 2.1.1) umsetzt. Dieses Konzept fokussiert das Selbstlernen mit geeigneten Materialien außerhalb der Präsenzveranstaltungen und die aktive Übung während der synchronen Termine mit Anwesenheit der Studierenden. Aus diesen Eigenschaften wurde eine Eignung für die genannten Ziele abgeleitet. Aus Forschungssicht bestanden dabei die zentralen Fragen, wie ein Flipped Classroom im Kontext der Veranstaltung und der Rahmenbedingungen wie Teilnehmendenzahl, Heterogenität und Ressourcen sinnvoll umgesetzt werden kann und welche Wirkungen das entstehende Konzept auf die Studierenden hat. Dabei waren die Auswirkungen auf die Bearbeitung der Aufgaben sowie der Umgang mit Vor- und Nachbereitung von Veranstaltungen durch Studierende von besonderem Interesse.

Auf Basis der Anforderungen und der Literatur wurde iterativ ein Konzept entwickelt, das zunächst nur einen partiellen Flipped Classroom als Ersatz für die bisherige Übung enthielt. Im weiteren Verlauf entstand daraus ein vollständiger Flipped Classroom, bei dem auch die klassische Vorlesung ersetzt wurde. Zur Vorlesungszeit wurde eine Live-Sitzung angeboten, die während der pandemischen Einschränkungen in verschiedenen technischen Formaten digital stattfand. Zur zentralen Inhaltsvermittlung wurden interaktive Lernmaterialien auf der Plattform JupyterNotebooks zur Verfügung gestellt, die gleichzeitig als Entwicklungsumgebung für die Programmieraufgaben genutzt wurde. Die Lernmaterialien, die innerhalb der Veranstaltung als *Lessons* bezeichnet wurden, setzten sich dabei aus folgenden Bestandteilen zusammen:

- kurze Hinweisen zu Inhalten, die für die jeweilige Lesson vorausgesetzt werden, in Form von verlinkten Lessons
- Slideshows in Form von vertonten Folien, die im Gegensatz zu einem klassischen Video auch einzeln betrachtet werden können
- Markdown- sowie Codezellen, die konkrete Codebeispiele zeigen und erläutern
- kurze Hinweise zu weiterführenden Inhalten und möglichen Ergänzungen, in Form von verlinkten Lessons

In den Codebeispielen können auch Aufgaben von geringem Umfang enthalten sein, bei denen der gegebene Code ergänzt oder verändert werden soll. Dabei wurden wie auch bei vollständigen Codebeispielen Markdownzellen für Erläuterungen verwendet. Für Codezellen ist die Ausführung und Modifikation des enthaltenen Codes möglich. Für den Großteil der Aufgaben konnten im Anschluss Lösungen eingeblendet werden.

Die Übungen wurden zu *Coding Classes* umgestaltet, in denen der Fokus auf der aktiven Anwendung der Inhalte aus den Lessons liegt. Dafür wurden gesonderte Aufgaben erstellt, die ab dem zweiten Jahr der Umstellung in das reguläre Aufgabenblatt integriert wurden und bis zum Projekt jeweils direkt auf die Aufgaben zur Abgabe vorbereiten sollen. Diese werden in den Coding Classes nach einem kurzen Intro, in dem Grundbegriffe besprochen und im Rahmen von gemeinsamen Live-Coding vorbereitende Aufgaben angefangen werden, von Kleingruppen bearbeitet. Die Coding Class Leitenden helfen dabei, indem sie Fragen beantworten und Tipps bei auftretenden Schwierigkeiten geben. Zum Abschluss erfolgt eine Diskussion von Lösungen und Lösungsansätzen vorzugsweise durch die Studierenden, wobei diese gegebenenfalls durch die Lehrende ergänzt werden, wenn hierfür ein Bedarf besteht. Durch das kooperative Gamificationkonzept [OKHB21] ab dem zweiten Jahr der Umstellung erhielten die Studierenden für Aktivität in den Coding Classes wie auch in den Live-Sitzungen Algos, die zum Freischalten von vor allem organisatorischen Vorteilen und unterhaltsamen Inhalten führten. Diese wurden von allen Studierenden gemeinsam gesammelt, damit der kooperative Aspekt im Fokus steht und keine Konkurrenz gefördert wird. Für das Vorstellen von Lösungen oder Lösungsansätzen in Coding Classes wurde dabei eine gegenüber regulärer Aktivität erhöhte Zahl von Algos vergeben, um die besondere Kooperationsbereitschaft der Studierenden zu belohnen. Im Rahmen der Belohnung konnte die Möglichkeit freigeschaltet werden bis zu 5 % der Modulnote in Form von Bonuspunkten erreichen zu können. Diese mussten durch besonders kreative Lösungen des Projekts noch individuell erreicht werden, sodass zunächst eine Gruppenleistung nötig war, die genaue Bewertung aber individuell erfolgte. Schon zuvor wurden Elemente aus der Popkultur vereinzelt in die Materialien und in die Aufgabengestaltung einbezogen. Mit dem Gamificationkonzept wurde dies als Gestaltungsprinzip stärker flächendeckend angestrebt. Durch versteckte unterhaltsame Inhalte mit Bezug zu den Lerninhalten sollten die Studierenden zusätzlich motiviert werden.

Nach den in der Veröffentlichung [OBB⁺22] beschriebenen Iterationen folgten zwei weitere Durchgänge, in denen die Änderungen wesentlich geringer ausfielen, da das Konzept aus dem zweiten Jahr im Grundsatz als erfolgreich bewertet wurde. Eine Übersicht über die Konzepte im Verlauf der Zeit und die jeweils durchgeführten Änderungen in den Modulbestandteilen zeigt Tabelle 5.1.

	Vor 2019/2020	2019/2020 (1. Jahr)	2020/2021 (2. Jahr)	2021/2022 (3. Jahr)	2022/2023 (4. Jahr)
Vorlesung	klassische Vorlesungen mit Aufzeichnung	klassische Vorlesungen mit Aufzeichnung	Online-Streaming, Fragen per Chat	Online-Konferenz , Möglichkeit für Fragen	Live-Sitzungen , Möglichkeit für Fragen
Übung - Inhalt	klassisch: Besprechung von Lösungen, Tipps für nächstes Blatt	Coding Class: Aufgabenbearb., Vorstellung durch Stud.	Coding Class: Intro , Aufgabenbearb. Besprechung mit Stud.	Coding Class: Intro , Aufgabenbearb., Besprechung mit Stud.	Coding Class: Intro , Aufgabenbearb., Besprechung mit Stud.
Übung - Terminanzahl	4 Termine, alle donnersdays, in Präsenz	13 Termine, Mittwoch bis Freitag , in Präsenz	10 Termine , Mittwoch bis Freitag, online	11 Termine , davon 6 in Präsenz und 5 online, Mittwoch bis Freitag	8 Termine , Montag bis Donnerstag , davon 2 online
Abgabeaufgaben - Rhythmus	wöchentlich	dreiwöchentlich	dynamisch wachsend: 4 x wöchentlich, 2 x zweiwöchentlich, 1x dreiwöchentlich	dynamisch wachsend: 4x wöchentlich, 2x zweiwöchentlich, 1x dreiwöchentlich	dynamisch wachsend: 4x wöchentlich, 2x zweiwöchentlich, 1x dreiwöchentlich
Abgabezeitpunkt	zum Testat	Sonntag 18 Uhr	Sonntag 18 Uhr	Sonntag 18 Uhr	Sonntag 18 Uhr
Gruppengröße	2	3	3	3	3
Testate - Termine	Montag bis Mittwoch	Montag bis Freitag	Montag bis Freitag	Montag bis Freitag	Montag bis Freitag
Programmiersprache	Java	C++	C++	C++	C++
Programmierungsumgebung	Vim, Atom	QtCreator mit OpenFrameworks	JupyterNotebooks mit AlgoViz	JupyterNotebooks mit AlgoViz	JupyterNotebooks mit AlgoViz
Materialien zur Vorbereitung auf Übung	keine	Courseware-Inhalte mit Zusätzen	Interaktive Materialien in JupyterNotebooks für alle Inhalte	Interaktive Materialien in JupyterNotebooks für alle Inhalte	Interaktive Materialien in JupyterNotebooks für alle Inhalte
Einschränkungen durch Pandemie	keine	keine (außer Klausur)	ausschließlich Online-lehre	Teilnahmebeschränkungen	keine
Beratungszeit	Donnerstag/Freitag	Mittwoch bis Freitag	Mittwoch bis Freitag	Mittwoch bis Freitag	Montag bis Donnerstag Präsenz, Freitag online
Ausgabe Aufgabenteil/Material	Montag	CC: Mittwoch f. Folgewoche , Rest Montag	Montag, CC integriert	Montag, CC integriert	Donnerstag f. Folgewoche , CC integriert

Tabelle 5.1: Übersicht über Modulkonzepte zwischen 2018 und 2022, Änderungen in Fettdruck

Im dritten und vierten Jahr wurden die Bestandteile vor allem den geänderten Bedingungen aufgrund der Pandemie angepasst, sodass zunehmend mehr Veranstaltungen in Präsenz stattfinden konnten. Außerdem wurden im vierten Jahr einige Termine im Wochenplan geändert, da einige Studierende den im Verhältnis unpassenden Termin für die Live-Sitzung kritisierten. Dieser lag in den vorherigen Jahren thematisch häufig nach einer Abgabe, sodass für die Aufgaben relevante Fragen bis dahin bereits geklärt waren. Aufgrund der Durchführung der Live-Sitzung und grundsätzlich auch anderer Veranstaltungen in Präsenz wurden im vierten Jahr wieder beide Vorlesungstermine montags und dienstags angeboten, um möglichst vielen Studierenden trotz möglicher zeitlicher Überschneidungen mit anderen Veranstaltungen die Teilnahme zu ermöglichen. Die Änderungen ab dem dritten Jahr, die Ergebnisse der zugehörigen Evaluation sowie der Ausblick auf das vierte Jahr werden im folgenden Kapitel 6 beschrieben.

6 Weitere Ergebnisse

Nach der iterativen Konzeptentwicklung für die Durchführungen im Wintersemester 2019/20 sowie 2020/21 folgte noch eine weitere Durchführung im Wintersemester 2021/22 mit weitgehend zum Wintersemester 2020/21 identischem Konzept. Durch die fortgeschrittene Pandemie konnte in diesem zu Beginn eine anteilige Präsenz für kleine Gruppen ermöglicht werden, die zu Mitte Dezember jedoch aufgegeben werden musste. Durch die anteilige Präsenz konnten Erkenntnisse für die auf Dauer geplante Umsetzung in überwiegender Präsenz gesammelt werden. Diese und weitere grundsätzliche Ergebnisse zur weiteren Stützung der bisherigen Ergebnisse werden in diesem Kapitel vorgestellt.

6.1 Modulkonzept im Wintersemester 2021/22

Das Modulkonzept im Wintersemester 2021/22 beruhte weitgehend auf dem weiterentwickelten Konzept mit Coding Classes als aktive Übung zu interaktivem Lernmaterial in Form von JupyterNotebooks wie in der Veröffentlichung [OBB⁺22] beschrieben. Auch das Gamifikationskonzept [OKHB21] und die synchrone Fragestunde in digitaler Form wurden weitergeführt. Wie im Jahr zuvor wurde den Studierenden in der ersten Woche der Vorlesungszeit durch die Lehrenden in einer Videokonferenz das Konzept ausführlich erläutert und im Lern-Management-System dokumentiert. In diesem ist auch ein Wochenablauf enthalten, der aufzeigt welche Veranstaltungsteile an welchem Tag angeboten werden und dass eine selbstständige Arbeitsweise besonders beim Bearbeiten der Lessons erforderlich ist.

Aufgrund der Pandemielage durften zwischen Ende Oktober 2021 und Mitte Dezember 2021 allerdings Coding Classes in Präsenz mit einer maximalen Teilnehmendenzahl von 20 Studierenden mit der Betreuung von zwei Coding Class-Leitenden stattfinden. Um dennoch allen Studierenden die Möglichkeit zur Teilnahme an einer Coding Class zu gewähren, fanden nur sechs Termine pro Woche in Präsenz statt und fünf weitere Termine wie im Wintersemester 2020/21 als Onlinekonferenz. Alle Termine sollten unter Berücksichtigung individueller Interessen der Teilnehmenden inhaltlich deckungsgleich sein und waren auf die Tage Mittwoch bis Freitag verteilt. Einige Coding Classes in Präsenz fanden direkt parallel zu digitalen Coding Classes statt. Ab Mitte Dezember bis Anfang Februar 2022 fanden die Coding Classes ausschließlich digital statt, wobei parallel stattfindende Termine häufig zusammengelegt und dann von vier Lehrenden begleitet wurden.

Für die Präsenztermine war eine Anmeldung über das Lern-Management-System erforderlich, wobei die Studierenden diese jeweils ab Freitag Nachmittag der Vorwoche bis zum Beginn der jeweiligen Coding Class selbst vornehmen konnten. Es war dabei möglich andere bereits für die Coding Class angemeldete Studierende zu sehen. Aufgrund der Dokumentationspflicht mussten alle Teilnehmenden beim Betreten einer in Präsenz

durchgeführten Coding Class ihre Kontaktdaten hinterlassen und die Anforderungen an ihre Gesundheit erfüllen ¹. Die Präsenztermine fanden im 2019 dafür speziell eingerichteten PC-Raum statt. Dieser bietet neben Computern am Rand Gruppentische in der Mitte sowie ein Smartboard für Präsentationen und Skizzen.

Die Live-Sitzung fand weiterhin online als Fragestunde statt, aufgrund technischer Ausrüstung allerdings in einer normalen Videokonferenz statt wie bisher als Streaming mit Chat über eine andere Plattform. Dies führte dazu, dass der im vorherigen Jahr verwendete Chat kaum genutzt wurde. Im Rahmen der offiziellen Kommunikationskanäle sollten Fragen außerhalb der Veranstaltungstermine daher im Forum im Lern-Management-System gestellt werden. Als inoffiziellen und nicht moderierten Kommunikationsweg wurde aber auch die Verbreitung von Links zu von Studierenden erstellten Gruppen in Kurznachrichtendiensten unterstützt. Wie bereits im Vorjahr wurden die Live-Sitzungen auch aufgezeichnet, um Studierenden, die zeitlich parallele Veranstaltungen besuchen, die Inhalte dennoch zugänglich zu machen.

Auch die Beratungszeit wurde weiterhin digital angeboten. Testate durften bis Mitte Dezember bei Zustimmung aller Beteiligten in Präsenz stattfinden, wobei die meisten Testate online erfolgten. Zum Zeitpunkt der Umstellung auf vollständig digitale Lehre absolvierten lediglich noch vereinzelte Testatgruppen ihre Testate in Präsenz. Die Abgabebefragungen und einzelne Lessons wurden basierend auf den Rückmeldungen der Tutoren in den wöchentlichen Besprechungen des vorherigen Jahres in geringem Umfang überarbeitet. Insbesondere wurden bei einigen Aufgaben Formulierungen angepasst, die zuvor vereinzelt zu Missverständnissen geführt hatten.

Studierende, die bereits in einem vergangenen Jahr die Zulassung zur Klausur erhalten hatten, mussten nicht erneut am Testatsystem teilnehmen. Jedoch hatten diese die Möglichkeit das Projekt zu wiederholen, um erneut Bonuspunkte zu erhalten, da diese lediglich für die zwei Klausurtermine im zugehörigen Semester gültig sind. Dabei mussten sie aus den fünf Projektthemen ein anderes wählen als sie im Vorjahr für die Zulassung zur Klausur bearbeitet hatten.

Im Wintersemester 2021/22 wurde eine etwas geringere Anzahl an Teilnehmenden festgestellt, die in etwa bei 450 lag, von denen sich 392 zu Beginn in das Testatsystem eingetragen haben. In den Jahren zuvor haben etwa 550 bis 600 Studierende am Modul teilgenommen. Bei der Abschlussevaluation im Wintersemester 2020/21 haben 204 Studierende vollständig teilgenommen, im Wintersemester 2021/22 waren es 130 vollständig ausgefüllte Fragebögen. Wie auch bisher fand diese in den letzten zwei Wochen der Vorlesungszeit, an die sich die Klausur anschloss, online über das Tool *LimeSurvey* [Lim] statt. Für die Teilnahme wurden innerhalb des Gamificationkonzepts Punkte vergeben, die zu Belohnungen führen konnten und der Zugang wurde über eine Ankündigung im Lern-Management-System sowie Nachrichten an alle zu diesem Zeitpunkt in der Veranstaltung eingetragenen Studierenden verbreitet. Durch eine Instanz des Evaluationstools auf universitätseigenen Servern sowie entsprechende Hinweise zu Beginn der Evaluation konnten Datenschutz und Anonymität gewährleistet werden.

¹Zum Betreten der Räume der Universität wurden ein 3G-Status (geimpft, genesen oder getestet) und das Fehlen von Erkrankungssymptomen vorausgesetzt.

6.2 Quantitative Ergebnisse

Aus den Evaluationen und Anwesenheitsstatistiken ergibt sich eine Reihe quantitativer Daten, von denen die wichtigsten Auswertungsergebnisse im Folgenden vorgestellt werden. Dabei ist insbesondere das Wintersemester 2021/22 als Fortsetzung des Konzepts vom Wintersemester 2020/21, dem zweiten Jahr der Umstellung, von besonderer Relevanz. Trotz gering veränderter äußerer Bedingungen durch die fortgeschrittene pandemische Lage können die zusätzlichen Daten die bisherigen ergänzen, um ein umfassenderes Bild der Wirkung des Konzepts zu erhalten.

Tabelle 6.1 zeigt eine Erweiterung der Ergebnisse aus der Veröffentlichung [OBB⁺22] um das dritte Jahr mit verändertem Modulkonzept wie in Abschnitt 6.1 beschrieben. Für die Bearbeitung der Abgabenaufgaben ergeben sich keine signifikanten Veränderungen gegenüber den Ergebnissen im zweiten Jahr (Mann-Whitney-U-Test ²: $W = 12075, p = 0,8069$ für reguläre Testataufgaben, $W = 12279, p = 0,5224$ für das Projekt). Bei den Lessons ergibt sich eine geringe, aber signifikante Verschlechterung des eingeschätzten Lernerfolgs (Mann-Whitney-U-Test: $W = 13862, p = 0,0176$). Für das Livestreaming (Mann-Whitney-U-Test: $W = 12534, p = 0,012$) sowie die Coding Classes in Präsenz (Mann-Whitney-U-Test: $W = 8011,5, p < 0,001$), aber nicht in digitaler Form (Mann-Whitney-U-Test: $W = 10209, p = 0,1248$) können signifikante Verbesserungen zwischen den beiden betrachteten Jahren bei der subjektiven Lerneinschätzung durch die Studierenden festgestellt werden. Beim Vergleich innerhalb des Wintersemesters 2021/22 gaben die Studierenden an in den Coding Classes in Präsenz durchschnittlich mehr gelernt zu haben als in der digitalen Variante, der Unterschied ist jedoch nicht signifikant (Mann-Whitney-U-Test: $W = 7554, p = 0,1217$). Bei der Bearbeitung der Abgabenaufgaben ergibt sich wie auch im Vorjahr bereits ein signifikanter (Mann-Whitney-U-Test: $W = 8599, p = 0,018$) Unterschied zwischen den regulären Aufgaben im Rhythmus von einer bis zwei Wochen und dem Projekt.

Für die Lessons wurden die Studierenden auch um eine Einschätzung der Zeit, die sie wöchentlich im Durchschnitt damit gelernt haben, gebeten. Dabei sollte die Wahl auf einer achtstufigen Skala zwischen den Möglichkeiten 1 für «<10 Minuten» bis 8 für «>180 Minuten» in Schritten von 30 Minuten mit Ausnahme der beiden ersten Möglichkeiten (2 für «11 - 30 Minuten») getroffen werden. Während der Mittelwert für das Wintersemester 2020/21 bei 5,7846, entsprechend etwa 120 bis 150 Minuten liegt, sank dieser für das Wintersemester 2021/22 leicht auf 5,472, entsprechend etwa 120 Minuten. Es handelt sich dabei nicht um eine signifikante Änderung (Mann-Whitney-U-Test: $W = 13278, p = 0,1707$).

Für die erfolgten Zulassungen zur Klausur durch Absolvieren der Testate ergibt sich für das Wintersemester 2021/22 ein ähnliches Ergebnis wie im Wintersemester 2020/21. 351 der zu Beginn eingetragenen 392 Studierenden haben die Zulassung erreicht. Dies entspricht 89,54 %. Im Vorjahr hatten 90 % die Testate erfolgreich bestanden. In der Gesamtbetrachtung liegen diese Ergebnisse etwa im Bereich der Zulassungsquote aus dem klassischen Testatsystem mit wöchentlichen Rhythmus (86,29 % im Wintersemester

²entspricht dem Wilcoxon-Rangsummentest für unabhängige Stichproben

Tabelle 6.1: Mittelwerte der Items zur eigenen Lerneinschätzung in den Gesamtevaluationen zur Frage «Wie viel haben Sie bei folgenden Veranstaltungsteilen gelernt?», erweitert aus [OBB⁺22]

Items Gelernt	Mittelwert vor Umstellung	Mittelwert 1. Jahr	Mittelwert 2. Jahr	Mittelwert 3. Jahr
Vorlesung	2,3	2,54	-	-
Vorlesungs- aufzeichnung	2,38	2,65	-	-
Lessons	-	-	4,46	4,24
Livestreaming	-	-	2,76	3,22
Übung / Coding Class	2,82	3,31	3,27	4,38 (Präsenz)/ 3,99 (digital)
Bearbeitung der Abgabeaufgaben	1,85	2,33	4,29 / 4,01 (Projekt)	4,27 / 3,94 (Projekt)

Skala von 1 – «gar nichts» bis 5 – «sehr viel», n = 104 / 124 / 204 / 130

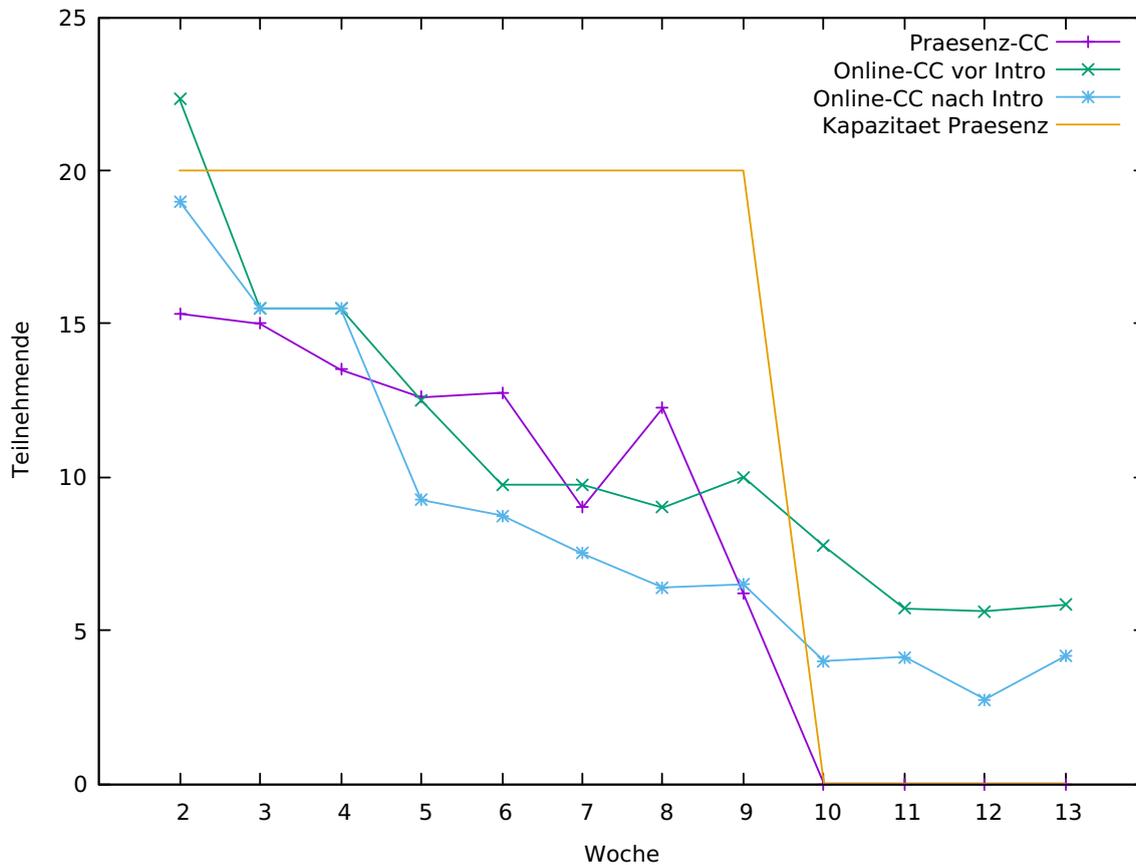
2018/19) und weiterhin deutlich über der Zulassungsquote im ersten Jahr der Umstellung mit einem durchgehend dreiwöchigen Rhythmus (78,01 %).

Abbildung 6.1 zeigt die durchschnittlichen Teilnehmendenzahlen in den Präsenz- sowie Online-Coding Classes im hybriden Wintersemester 2021/22, die von den jeweiligen Tutoren in den Coding Classes dokumentiert wurden. Bis einschließlich zur neunten Veranstaltungswoche durften pro Coding Class in Präsenz maximal 20 Studierende teilnehmen. Diese mussten sich vorher jeweils ab einem festgelegten Zeitpunkt am Ende der vorherigen Woche über die Veranstaltung im Lern-Management-System anmelden. Sobald alle Plätze in einer Coding Class belegt waren, konnte eine Anmeldung erst erfolgen, falls sich ein angemeldeter Studierender wieder abgemeldet hatte. Coding Classes, die online stattfanden, unterlagen keiner Teilnahmebeschränkung und erforderten auch keine Anmeldung. Es wurden sechs Coding Classes in Präsenz sowie fünf weitere Coding Classes online angeboten.

Die Anzahl an Teilnehmenden ist in Präsenz zunächst geringer, bleibt aber im Verlauf konstanter. Unter Berücksichtigung der Kapazitätsgrenze wurden in sechs von acht Wochen mit Präsenz-Coding Class durchschnittlich mehr als die Hälfte der verfügbaren Plätze besucht. Ausschließlich in online durchgeführten Coding Classes trat die Situation auf, dass Studierende diese nach dem Intro, das aus dem gemeinsamen Bearbeiten einer kurzen Aufgabe besteht, wieder verließen. Dieses Phänomen trat in 34 von 53 (64,15 %) dokumentierten Online-Coding Classes auf.

Die Studierenden wurden in der Abschlussevaluation auch danach gefragt, ob sie ohne pandemiebedingte Einschränkungen eine Coding Class lieber in Präsenz oder digital besuchen würden. 42,44 % gaben an, dass sie «definitiv in Präsenz» bevorzugen würden, 19,66 % würden eine Coding Class «eher in Präsenz» wählen und 15,53 % geben an «unentschlossen» zu sein. Von den restlichen Studierenden würden 10,26 % eine Coding

Abb. 6.1: Durchschnittliche Anzahl an Teilnehmenden pro Coding Class in den Präsenz- und Online-Coding Classes im Wintersemester 2021/22 sowie Kapazitätsgrenze für Coding Classes aufgrund pandemiebedingter Maßnahmen



Class «eher digital» und 8,55 % «definitiv digital» besuchen. 4,27 % der Befragten gaben an grundsätzlich keine Coding Class besuchen zu wollen.

Zur Erfassung der Einschätzung der Studierenden zur konkreten Gestaltung der Programmierumgebung, der Inhalte und der Coding Classes wurden einige Konzepte in der Abschlussevaluation zur Bewertung gestellt. Tabelle 6.2 zeigt die Ergebnisse für die Wintersemester 2020/21 sowie 2021/22. An der konzeptionell-inhaltlichen Gestaltung hatten sich zwischen diesen beiden Jahren keine Änderungen ergeben. Insgesamt werden die meisten Aspekte eher zustimmend für die Sinnhaftigkeit bzw. Angemessenheit bewertet. Insbesondere die handlungsorientierten Modelle, die Vorstellung von Lösungen durch Kommilitonen in den Coding Classes sowie die inhaltliche Abfolge von Programmieren und Algorithmik im Kursverlauf wurden positiv bewertet. Die Studierenden stimmten eher nicht der Aussage zu, dass es ihnen an weiteren Ideen fehlte, wie sie die Selbstlernzeit besser nutzen könnten. Ebenfalls stimmen sie eher nicht zu sich mehr Arbeitszeit in den Coding Classes gewünscht zu haben. Signifikante Unterschiede zwischen den zwei betrachteten Jahren bestehen nur beim Einsatz von JupyterNotebooks und AlgoViz. Bei JupyterNotebooks fiel der Mittelwert dabei um eine halbe Stufe auf der vierstufigen Skala, bei AlgoViz um eine Viertelstufe. Dennoch bleiben die Zustimmung bei beiden Items weiterhin auf dem Niveau einer tendenziellen Zustimmung.

Tabelle 6.2: Mittelwerte und Mann-Whitney-U-Test für Konzeptitems für die Durchführungen im zweiten und dritten Jahr

Item	Mittelwert 2020/21	Mittelwert 2021/22	Mann-Whitney-U
Das Live Coding [...]³ war sinnvoll.	3,375	3,2959	$W = 8397,5$, $p = 0,2917$
Den Einsatz von JupyterNotebooks empfand ich als sinnvoll.	3,3177	2,8291	$W = 14400$, $p < 0,001*$
Ich empfand den Einsatz von AlgoViz als sinnvoll.	3,5	3,2672	$W = 12691$, $p = 0,0071*$
Die verschiedenen handlungsorientierten Modelle für Algorithmen haben zum Verständnis beigetragen.	3,5029	3,3469	$W = 9397$, $p = 0,0939$
Ich bin damit zufrieden, dass Kommilitonen in den Coding Classes Lösungen vorgestellt haben.	3,4621	3,5783	$W = 5518,5$, $p = 0,2286$
Das Verhältnis zwischen Arbeitsphasen und Besprechungsphasen in den Coding Classes halte ich für angemessen.	3,2378	3,1429	$W = 5820,5$, $p = 0,4525$
Ich hätte mir mehr Arbeitszeit in der Coding Class gewünscht.	2,0417	2,1667	$W = 4977$, $p = 0,612$
Ich hätte mir mehr Zeit für Besprechungen in der Coding Class gewünscht.	2,6806	2,9014	$W = 4485,5$, $p = 0,1283$
Ich empfand es als sinnvoll, dass zunächst der Fokus auf Programmierkompetenzen und erst danach auf Algorithmen lag.	3,5167	3,4466	$W = 9662$, $p = 0,4957$
Ich empfand die gewünschte Vorbereitung auf die Coding Class grundsätzlich als belastend.	2,3603	2,5513	$W = 4747,5$, $p = 0,1856$
Mir fehlte es an weiteren Ideen, wie ich die Selbstlernzeit besser nutzen kann.	2,1905	2,4184	$W = 7252,5$, $p = 0,0934$

Skala: 1- «stimmt nicht», 2- «stimmt eher nicht», 3- «stimmt eher», 4- «stimmt»
n = 204 / 130, * signifikant

Bei der Frage, ob das Konzept für alle Studierenden gleichermaßen erfolgreich ist, kann eine Näherung über die als bereits bekannt angegebenen Themen in der Evaluation am Ende des Moduls erfolgen. Es kann davon ausgegangen werden, dass Studierende, die angeben viele Themen bereits vor dem Besuch der Veranstaltung gekannt zu haben,

³An dieser Stelle folgte eine Erklärung des Begriffs *Live Coding*.

über höhere Vorkenntnisse verfügen. Dabei ist von besonderem Interesse, ob die dadurch angenommenen Vorkenntnisse mit der Einschätzung zum Lernerfolg oder der Gesamtbewertung der Veranstaltung korrelieren. Da kein eindeutig metrisches Skalenniveau vorliegt, wird der Spearman-Rang-Korrelationskoeffizient berechnet. Tabelle 6.3 zeigt die Ergebnisse für die Korrelationen zwischen der Anzahl bereits im Vorfeld bekannter Themen und ausgewählten Items für die Durchführungen in den Wintersemestern 2020/21 (zweites Jahr) sowie 2021/22 (drittes Jahr).

Tabelle 6.3: Spearman-Korrelationen zwischen der Anzahl bereits im Vorfeld bekannten Themen und ausgewählten Items für die Durchführungen im zweiten und dritten Jahr

Korrelation	Wintersemester 2020/21	Wintersemester 2021/22
Bereits bekannte Themen x Gelernt Lessons	$S = 1273509$ $p = 0.8365$ $\rho = -0.0148$	$S = 377765$ $p = 0.0357 *$ $\rho = -0.1889$
Bereits bekannte Themen x Gelernt Live-Sitzung	$S = 970240$ $p = 0.2757$ $\rho = 0.0806$	$S = 278410$ $p = 0.4535$ $\rho = -0.0703$
Bereits bekannte Themen x Gelernt Coding Class (für 2021/22: Präsenz/Digital ⁴)	$S = 1193644$ $p = 0.7019$ $\rho = -0.0279$	$S = 251589/261445$ $p = 0.8412/0.4555$ $\rho = -0.019/0.069$
Bereits bekannte Themen x Gelernt Testat	$S = 1160699$ $p = 0.2958$ $\rho = 0.0751$	$S = 285333$ $p = 0.7148$ $\rho = 0.0336$
Bereits bekannte Themen x Gelernt Bearbeitung der Aufgaben (Aufgaben/Projekt)	$S = 1022141/1176607$ $p = 0.0157 * /0.5062$ $\rho = 0.1729/0.0479$	$S = 317490/286162$ $p = 0.591/0.7378$ $\rho = -0.0491/0.0307$
Bereits bekannte Themen x Teilnahme an Live-Sitzung	$S = 1409445$ $p = 0.0855$ $\rho = -0.1232$	$S = 328857$ $p = 0.9091$ $\rho = -0.0103$
Bereits bekannte Themen x Teilnahme an Coding Class (für 2021/22: Präsenz/Digital ⁴)	$S = 1498627$ $p = 0.0064 *$ $\rho = -0.1942$	$S = 368578/380559$ $p = 0.1412/0.0115 *$ $\rho = -0.1323/ -0.2271$
Bereits bekannte Themen x Schulnote für Veranstaltung	$S = 1185713$ $p = 0.5739$ $\rho = 0.0405$	$S = 371461$ $p = 0.0043 *$ $\rho = -0.2582$

Skalen: Themenanzahl: 0 bis 11, Gelernt von 1 - «gar nichts» bis 5 - «sehr viel», Teilnahme von 1 - «nie» bis 6 - «mehr als 80 %», Schulnoten von 1 - «sehr gut» bis 6 - «ungenügend»,
n = 204 / 130, * signifikant

Dabei ergeben sich für Items zur Lerneinschätzung zwei signifikante Korrelationen. Für das Wintersemester 2020/21 besteht zwischen der Anzahl bereits bekannter Themen und

⁴2020/21 wurden ausschließlich digitale Coding Classes angeboten.

der Einschätzung des Lernerfolgs durch Bearbeitung der wöchentlichen und zweiwöchentlichen Aufgaben ein signifikanter, schwach positiver Zusammenhang. Studierende, die für mehr Themen angeben diese bereits gekannt zu haben, schätzen ihren Lernerfolg durch das Bearbeiten dieser Aufgaben daher als tendenziell höher ein. Im Wintersemester 2021/22 liegt eine andere signifikante Korrelation vor. Die Anzahl bereits bekannter Themen korreliert dabei signifikant schwach negativ mit dem eingeschätzten Lernerfolg durch die Lessons. Dies bedeutet, dass Studierende, die bereits viele Themen vor der Veranstaltung kannten, ihren Lernerfolg durch die Lessons tendenziell schlechter einschätzen als Studierende, denen bisher keine oder wenige Themen bekannt waren.

In beiden betrachteten Semestern liegen jeweils signifikante schwache bis mittlere negative Korrelationen zwischen der Anzahl bereits bekannter Themen und der angegebenen Teilnahme an digitalen Coding Classes vor. Dies bedeutet, dass Studierende, denen mehr Themen im Vorfeld bekannt waren, die digitalen Coding Classes tendenziell weniger häufig besucht haben. Für die Coding Classes in Präsenz, die im Wintersemester 2021/22 teilweise parallel angeboten wurden, liegt keine signifikante Korrelation zur Anzahl bereits bekannter Themen vor.

Für das Wintersemester 2021/22 liegt außerdem bei der Schulnote, die die Studierenden zum Ende der Evaluation für die Veranstaltung insgesamt vergeben haben, eine signifikante Korrelation vor. Da die Skala für Schulnoten umgekehrt zur Skala für die Lerneinschätzung verläuft, bedeutet die vorliegende signifikant schwache bis mittlere negative Korrelation, dass Studierende, denen mehr Themen bekannt waren, tendenziell eine bessere Schulnote in der Evaluation vergeben haben als Studierende, denen bisher keine oder wenige Themen bekannt waren.

Tabelle 6.4: Bewertungen des Algo-Konzepts und Easter-Eggs für die Durchführungen in den Wintersemestern 2020/21 (zweites Jahr) sowie 2021/22 (drittes Jahr)

	A1	A2	A3	A4	A5	Mittelwert
Algo-Konzept 2020/21	46,63 %	36,27 %	8,81 %	5,18 %	3,11 %	1,8187
Algo-Konzept 2021/22	36,89 %	35,25 %	18,85 %	6,56 %	2,25 %	2,0246
Easter-Eggs 2020/21	44,62 %	36,56 %	18,28 %	0,54 %	0 %	1,7473
Easter-Eggs 2021/22	34,21 %	38,6 %	26,32 %	0,88 %	0 %	1,9386

Skalen: Bewertung Algo-Konzept von A1- «sehr gut» bis A5 - «sehr schlecht», Easter-Eggs von A1 - «Sehr positiv/motivierend» bis A5 - «Sehr negativ/störend», n = 204 / 130

Das Gamificationkonzept wurde in den Abschlussevaluationen quantitativ in jeweils zwei Items bewertet. Diese differenzierten zwischen den kooperativ sammelbaren Punkten (Algos) für aktives und hilfreiches Verhalten sowie der Gestaltung der Materialien. Gesammelte Algos führten dabei zum Freischalten von Belohnungen unterhaltsamer oder nützlicher Art. Zu den gestalterischen Elementen in den Materialien gehören dagegen thematische Einkleidung von Aufgaben mit Kontexten aus der Popkultur, Verlinkungen zu Memes und ähnliche humorvolle Einschübe in Form von Easter-Eggs. Diese sollten ebenfalls eine motivierende Wirkung haben, allerdings wurden durch die Interaktion keine Folgen ausgelöst.

Tabelle 6.4 zeigt die Ergebnisse der Items für das Wintersemester 2020/21, in dem das Konzept zuerst angewendet wurde, sowie für das Wintersemester 2021/22. Dabei sind signifikante Unterschiede zwischen den beiden Jahren zu erkennen (Mann-Whitney-U-Test: $W = 10244, p = 0.0378$ für das Algo-Konzept, $W = 9212, p = 0.0412$ für Easter-Eggs). Bei Betrachtung der konkreten Antwortanteile fällt auf, dass bei beiden Items für das Wintersemester 2021/22 eine Verschiebung aus dem Bereich einer sehr positiven Bewertung in die neutrale Bewertung erfolgt ist.

6.3 Qualitative Ergebnisse

Durch offene Fragen in den Evaluationen sowie durch schriftliche Interviews mit den Tutoren des Moduls wurden auch qualitative Daten gesammelt, die konkrete inhaltliche Erfahrungen widerspiegeln. Am Ende des Fragebogens der Gesamtevaluation für das Wintersemester 2021/22 wurden die Studierenden darum gebeten Anmerkungen, Kritik, Wünsche und Erläuterungen mitzuteilen. In vorherigen Fragen waren ebenfalls bereits einige qualitative Anmerkungen möglich, diese Fragestellung umfasste allerdings einen globaleren Blick.

Unter diesen Freitexten gehörten zu den am häufigsten genannten Themen Lob für Tutoren (13 Nennungen), eine positive Arbeitsatmosphäre (zehn Nennungen), allgemeines Lob für das Konzept mit vielseitigen Möglichkeiten und den dafür betriebenen Aufwand (zehn Nennungen). Mit jeweils neun Erwähnungen wurden ein zu hoher Arbeitsumfang sowie technische Schwierigkeiten mit JupyterNotebooks kritisiert. Je sieben Nennungen lobten die Eastereggs und die Lessons als Lernmaterialien. Sechs Teilnehmende merkten an, dass die pandemischen Umstände kein Problem für den Kurs gewesen seien und der Kurs unter den digital angebotenen positiv aufgefallen ist. Häufig im Zusammenhang mit der hohen Arbeitsbelastung merkten fünf Teilnehmende an, dass sie sich einen speziellen Kurs für Studierende, die nicht Informatik als Hauptfach studieren, wünschen. Für die Slideshows wurde von vier Personen um eine Möglichkeit zur Geschwindigkeitskontrolle gebeten. Weitere Kommentare bezogen sich auf zeitliche Strukturen wie zu kurze Testate oder nach der Abgabe am Sonntag terminierte Live-Sitzungen. Einige stellten einzelne Elemente wie z. B. die Algos oder die Offenheit der Aufgaben als überwiegend positiv dar. Zwei Studierende merkten an, dass es ihnen wesentlich leichter fiel Fragen in privaten Chatgruppen zu stellen statt im Forum der Veranstaltung, da dort sowohl die Handhabung beim Einfügen von Medien einfacher als auch Antworten schneller zu erwarten seien.

Während eines Zeitraums von zwei Wochen vor und einer Woche nach dem ersten Klausurtermin wurden die Tutoren des Kurses zu ihren Erfahrungen befragt. Dabei wurden die Bestandteile der Lehrveranstaltung wie Testate, Coding Classes, Aufgaben oder die Lernumgebung JupyterNotebooks als Kategorien vorgegeben und um einige allgemeine Aspekte wie beispielsweise die Kommunikation ergänzt. Für das Wintersemester 2021/22 wurden dabei sieben ausführliche Beiträge diskutiert. Ein Hauptthema unter den Tutoren ist dabei die JupyterNotebooks-Umgebung, die häufig zu Schwierigkeiten beim Testen von Programmen führt. Die Studierenden wären bei einem Ausfall des Kernels, der zeitweise auftritt, unsicher, ob dies ein Fehler von ihrem Programm wäre

oder ein technischer Fehler ist. Die Tutoren wünschen sich eine Entwicklungsumgebung wie diese für größere Programmierprojekte genutzt wird. Ein Vorschlag bezieht sich dabei konkret darauf VSCode mit einem JupyterNotebooks-Plugin zu verwenden.

Laut den Tutoren hatten einige Studierende insgesamt Schwierigkeiten damit die Arbeitsumgebungen wie auch das Testatsystemtool angemessen zu bedienen. Die Tutoren plädieren hier teilweise für eine gebildete Anleitung. Bei den Beratungszeiten halten sie flexible Zeitgestaltungen im Kursverlauf für sinnvoll, allerdings auch in geringer besuchten Zeiten eine Besetzung mit mehreren Tutoren. Diese wäre insbesondere dann von Vorteil, wenn Studierende die Beratungszeit mit komplexeren Problemen aufsuchen und dadurch zwei Einschätzungen erhalten würden, die die Wahrscheinlichkeit steigern, dass auch schwierigere Fragen zufriedenstellend gelöst werden. Außerdem sprechen sich die Tutoren dafür aus Testate auch weiterhin online zu ermöglichen, um vor allem im Krankheitsfall mehr Flexibilität zu ermöglichen. Viele leicht erkrankte Studierende würden sich wünschen die Inhalte des Testats nicht zu verpassen, könnten aber insbesondere aufgrund der pandemischen Lage nicht in Präsenz teilnehmen.

6.4 Diskussion

Zunächst lässt sich feststellen, dass sich der Lerneffekt der angebotenen Veranstaltungen im dritten Jahr nach Einschätzung der Studierenden gesteigert hat. Das Livestreaming und die Coding Class wurden dabei im Bezug auf die Lerneinschätzung signifikant besser bewertet. Ein wahrscheinlicher Erklärungsansatz dafür besteht in zunehmender Erfahrung der Lehrenden, aber auch in zunehmender Verbreitung des aktiven Lernkonzepts, sodass sich vermutlich auch mehr Studierende darauf einlassen. Fortgeschrittenere Studierende können Studienanfängern von ihren eigenen Erfahrungen berichten, sodass diese ihre Erwartungen anpassen können (vgl. auch Kapitel 4) und tendenziell eher passende Arbeitsweisen innerhalb des Konzepts anwenden. Durch die Wiederholung der konkreten Gestaltung der Inhalte, Aufgaben und Intros in den Coding Classes konnten außerdem bereits einige der Erfahrungen aus dem vorherigen Durchgang in eine leicht überarbeitete Version einfließen. Dabei wurden einige Formulierungen zum besseren Verständnis angepasst und in den Intros konnte stärker darauf eingegangen werden welche Themen für die Studierenden besonders herausfordernd waren.

Es ist erwartbar, dass der Lernerfolg im Projekt grundsätzlich als geringer bewertet wird als in den regulären Aufgabenblättern. Im Projekt liegt das Ziel weniger auf der Ebene der Anwendung neuer Inhalte, sondern stärker darauf die einzelnen Themen miteinander zu verknüpfen und durch das Programmieren in größerem Umfang programmierbezogene, aber stärker inhaltsunabhängige Kompetenzen zu erlernen. Dieser Kompetenzen sind sich Studierende allerdings wahrscheinlich geringer bewusst, sodass den Inhalten selbst ein höherer Lerneffekt zugesprochen wird. Dennoch ist die Motivation im Projekt meistens höher als bei den regulären Aufgaben, da hier mehr Kreativität gefragt ist und der zeitliche Druck durch einen längeren Bearbeitungszeitraum von drei Wochen abnimmt.

Obwohl auch die Lessons im dritten Jahr gegenüber dem zweiten Jahr bereits bei Meldung von Fehlern durch Studierende oder Tutoren in geringem Umfang überarbeitet wurden, bewerteten die Studierenden den Lernerfolg mit diesen als geringer als im

Vorjahr. Für dieses Ergebnis könnten mehrere Faktoren gemeinsam relevant sein, die sich nicht isoliert überprüfen lassen. Zum einen kann ein Effekt durch die höheren Lerneffekte für die anderen Modulbestandteile entstanden sein. Da die Lernmenge grundsätzlich durch den Umfang der Inhalte begrenzt ist, könnte sich hier die Verteilung des Erlernten auf die einzelnen Bestandteile der Veranstaltung verändert haben. Wenn die Studierenden durch andere Bestandteile der Veranstaltung einen höheren Lernerfolg haben, könnte der Anteil der Lessons am Lernerfolg geringer ausfallen. Dies würde die Lessons nicht abwerten, sondern lediglich andere Bestandteile aufwerten. Ein weiterer Erklärungsansatz berücksichtigt die geringere für die Lessons aufgewendete Zeit, wobei dieser Unterschied statistisch nicht signifikant ist. Im Zusammenhang mit der signifikant geringeren Zustimmung zum sinnvollen Einsatz von JupyterNotebooks und AlgoViz liegt es im Bereich des Möglichen, dass die Probleme durch die Entwicklungsumgebung entweder im Wintersemester 2021/22 in größerem Umfang aufgetreten sind oder als problematischer empfunden wurden. Dabei könnten höhere Erwartungen an die digitale Lehre aufgrund der fortgeschrittenen Pandemie entstanden sein, da das Wintersemester 2021/22 bereits das vierte zumindest überwiegend digitale Semester war und nicht mehr von einem *Emergency Remote Teaching* nach Hogdes et al. [HML⁺20] als Notlösung im Vergleich zu bewusstem Einsatz digitaler Lehre ausgegangen werden kann.

Beim Vergleich von Coding Classes in Präsenz und online deuten die Ergebnisse darauf hin, dass Coding Classes in Präsenz insgesamt als nützlicher empfunden werden. Die signifikante Verbesserung der subjektiven Lerneinschätzung der Studierenden zwischen den rein digitalen Coding Classes im Wintersemester 2020/21 und den Coding Classes in Präsenz im folgenden Jahr bestätigt die Hypothese. Zu den digitalen Coding Classes konnte kein signifikanter Unterschied für diese Einschätzung festgestellt werden, obwohl die Frage zu einem Zeitpunkt erfolgte als bereits seit über einem Monat keine Coding Classes in Präsenz mehr stattfanden. Die Teilnehmendenzahlen liegen für die Präsenz-Coding Classes zwar teilweise unter denen für digitale Coding Classes, jedoch müssen hier die Bedingungen beachtet werden, unter denen die Coding Classes in Präsenz stattfinden mussten. Insbesondere ist dabei die Deckelung der Teilnehmenden auf 20 pro Coding Class zu berücksichtigen. Häufig waren für eine Coding Class in Präsenz 20 Personen angemeldet, von denen aber nicht alle tatsächlich teilgenommen haben. Neben spontanen Entscheidungen gegen einen Besuch liegen mögliche Gründe dafür auch in der Pandemiesituation. Bei Krankheitssymptomen oder Quarantäne, die häufig bereits bei Kontakt zu Corona-Infizierten eingehalten wurde, war der Besuch einer Veranstaltung in Präsenz nicht erlaubt. Für einzelne Coding Classes fehlen darüber hinaus teilweise genaue Dokumentationen zur Teilnehmendenzahl. Direkt vor und nach den Weihnachtsferien hat außerdem aufgrund unvollständiger Lehrwochen nur ein Teil der regulär für eine Woche vorgesehenen Coding Classes stattgefunden.

In den digitalen Coding Classes verließen zudem einige Studierende bereits nach dem Intro die Veranstaltung und verpassten damit die eigentlich vorgesehene wichtige Kleingruppenarbeit. Damit reduzierten sie den Effekt der Coding Class wieder auf einen stärker zentralisierten Teil und zogen sich aus der Anforderung der eigenen Aktivität. In Präsenz wurde dieses Verhalten nicht beobachtet. Aus den bisherigen Erfahrungen im Wintersemester 2022/23 mit sechs Präsenzterminen ohne Teilnahmebeschränkung sowie zwei digitalen Coding Classes pro Woche bestätigen sich die Ergebnisse. Die Teilnehmendenzahlen der Präsenz-Coding Classes sind durchgehend höher und nur

einzelne Studierende verlassen die Coding Classes vorzeitig vor oder während der Arbeitsphase. Für einige Studierende ermöglichen die Online-Coding Classes jedoch eine größere Flexibilität, sodass wenige Termine als zusätzliches Angebot beibehalten werden können.

In den Testaten scheint es dagegen weiterhin sinnvoll einen Teil der Testate auch online anzubieten oder die Möglichkeit Studierende online dazuzuschalten zu gewährleisten. Hier zeigen sich nur geringere Nachteile, falls sich einzelne Studierende z. B. aufgrund technischer Probleme zurückziehen. Dagegen bestehen Vorteile in der nicht notwendigen Anfahrt für die vergleichsweise kurze Testatzeit von 30 Minuten und die Möglichkeit auch bei Krankheitssymptomen die Inhalte zumindest nachzuvollziehen.

Die Materialbereitstellungsplattform und Entwicklungsumgebung JupyterNotebooks wurde gehäuft von Studierenden und Tutoren kritisiert. Obwohl die angesprochenen Probleme zu einiger Frustration und Enttäuschung führen, ist ein Wechsel hier nicht einfach zu vollziehen und ein Teil der Kritik erfolgt von einer eingeschränkten Perspektive aus. JupyterNotebooks wird dabei häufig als rein didaktisches Tool aufgenommen, dessen Praxisrelevanz dabei vernachlässigt wird. JupyterNotebooks wird in diversen Bereichen auch außerhalb von Lehre eingesetzt und wurde bereits für eine wichtige Rolle in der Datenverarbeitung ausgezeichnet [Jup18]. Eine umfassende, nicht für die Bildung entwickelte, Entwicklungsumgebung kann auf Programmieranfänger dagegen überfordernd wirken [VHI14] und nach den Erfahrungen im ersten Jahr der Umstellung mit openFrameworks [LWC23] ebenfalls zu Problemen bei der Ausführung von Programmen führen. Studierende der Informatik und stark informatikbezogener Fächer werden sich in folgenden Kursen auch noch näher mit Entwicklungsumgebungen auseinandersetzen, während für Studierende entfernterer Fächer eine Beschäftigung damit nicht notwendig erscheint. Durch das Zusammenspiel mehrerer Komponenten wie der Bereitstellung von Materialien über NB-Git-Puller [The21] und AlgoViz [Bri21] ist außerdem eine Kompatibilität bei der Nutzung einer Entwicklungsumgebung mit einem JupyterNotebooks-Plugin nicht automatisch gegeben. Zur Vermeidung von Schwierigkeiten mit JupyterNotebooks sollte eher mehr Hilfestellung zur Interpretation von Fehlermeldungen gegeben werden, sodass die Lernenden besser einordnen können, ob es sich um einen Programmierfehler oder einen Ausfall des JupyterNotebook Kernels handelt.

Zur Bedienung von JupyterNotebooks und anderer verwendeter Tools sollten einige Möglichkeiten in den synchronen Veranstaltungen gezeigt werden. Aus der Forschung zum Umgang mit Informationstechnologie gibt es Hinweise darauf, dass der bewusste Umgang mit Computern zugunsten der Automatisierung abnimmt [FPD18] und damit nicht unter dem Begriff der *Digital Natives* davon ausgegangen werden sollte, dass die heutigen Studierenden sich problemlos selbst in neue Werkzeuge einarbeiten können. Eine sehr detaillierte Schulung könnte jedoch dazu führen, dass keine eigene Auseinandersetzung mit der Bedienung und den Möglichkeiten stattfindet, sodass hier ein geeignetes Maß gefunden werden muss, um auch die Kompetenzen im Umgang mit digitalen Geräten zu schulen.

Bei der Betrachtung des Zusammenhangs zwischen Vorkenntnissen und den Auswirkungen auf Lernerfolg, Teilnahme und Zufriedenheit lässt sich für den Lernerfolg lediglich eine signifikante Korrelation pro Jahr feststellen, diese tritt jedoch für verschiedene Items auf. Ein geringerer Lernerfolg durch Lessons ist bei höheren Vorkenntnissen grundsätzlich

erwartbar, da viele Themen aus den Lessons bereits bekannt sind. Für die Bearbeitung der Aufgaben im Wintersemester 2020/21 können ähnliche Gründe angenommen werden. Positiv zu bewerten ist jedoch, dass offenbar alle Studierenden beim Bearbeiten der Projektaufgaben ähnlich viel lernen konnten. Dies spricht dafür, dass das Projekt diverse Kompetenzen fördert und für alle Studierenden unabhängig von deren Vorkenntnissen eine angemessene Herausforderung darstellt. Auch für die restlichen Bestandteile der Veranstaltung weisen die fehlenden Korrelationen darauf hin, dass diese für diverse Studierende lernförderlich waren. Jedoch ist bei den Veranstaltungen zu beachten, dass durch den Besuch ein Selbstselektionseffekt auftreten kann, sodass diejenigen, die nicht glauben von einer Veranstaltung zu profitieren, diese tendenziell gar nicht besuchen.

Bei der Teilnahme an Veranstaltungen findet sich in beiden betrachteten Semestern eine signifikante Korrelation zwischen den bereits bekannten Themen und dem Besuch von digitalen Coding Classes. Für die Präsenz-Coding Classes liegt keine Korrelation vor. Im Zusammenhang mit dem häufig beobachteten Verlassen der digitalen Coding Classes nach dem Intro lässt sich vermuten, dass Studierende mit höheren Vorkenntnissen durch das Intro weniger lernen als durch die anschließende Übungsphase. Dies ist auch zu erwarten, da das Intro einen einfachen Einstieg darstellen soll, bei dem die Inhalte der Lessons reflektiert und erste Verknüpfungen zur praktischen Anwendung geschaffen werden sollen. Studierende, die bereits über Programmiererfahrung verfügen, können hier zwar zur Lösung beitragen, lernen aber häufig keine neuen Inhalte.

Für das Wintersemester 2021/22 liegt eine Korrelation zwischen den bereits bekannten Themen und der Gesamtnote für die Veranstaltung in der Abschlussevaluation vor. Dies bedeutet, dass Studierende mit weniger Vorkenntnissen tendenziell schlechtere Noten vergeben haben. Dabei ist es wahrscheinlich, dass Studierende mit geringen Vorkenntnissen sich in diesem Semester, das als Semester mit deutlichem Präsenzanteil angekündigt war, aber letztendlich in ein komplett digitales überführt wurde, überfordert gefühlt haben. Insgesamt stimmt es mit dem Eindruck der Lehrenden überein, dass viele Studierende dieses Semester als grundsätzlich belastender empfunden haben als bisherige Semester während der Pandemie, wie auch im Bezug auf Emergency Teaching bereits diskutiert wurde.

Trotz der zeitlichen Belastung sollte an regelmäßigen Abgaben und Überprüfungen der aktiven und kontinuierlichen Mitarbeit festgehalten werden. Sowohl die Literatur (siehe 2.1.2) als auch die Ergebnisse der Konzeptentwicklung für das Modul befürworten verpflichtende Aufgaben. Auch der Aufbau und das Zusammenspiel aus Lessons, Coding Class-Aufgaben und Abgabenaufgaben sowie Testaten erwies sich als förderlich. Besonders im Gegensatz zu dem ersten Konzept mit zur Vorlesung ergänzenden Materialien als Grundlage der Vorbereitung für die Coding Classes und jeweils getrennten Aufgabenblättern sind im aktuellen Konzept die Zusammenhänge und der Umfang sowie die Art der Vorbereitung deutlich klarer. Auf den ersten Blick scheint die Bearbeitung der Coding Class Aufgaben daher zunächst eine zusätzliche Belastung darzustellen, jedoch erkennen die Studierenden zunehmend besser, dass ein einfacherer Einstieg bei den schwereren Aufgaben auch zeitsparend wirken kann wie auch in anderen Untersuchungen zum Flipped Classroom bereits festgestellt wurde [ES17]. Durch die Steigerung der Effizienz können so in kürzerer Zeit auch mehr Aufgaben bearbeitet werden und die Erfahrung im Programmieren steigt.

Im fachlichen Kontext wird ein hoher Zeitaufwand grundsätzlich häufig in Veranstaltungsevaluationen kritisiert. Die angebotenen Veranstaltung sollten dabei als Angebote begriffen werden, die die Auseinandersetzung mit den Inhalten der Veranstaltung erleichtern und damit die Effizienz bei den verpflichtenden Teilen steigern können. Bei den vorgesehenen Semesterwochenstunden handelt es sich dabei eher um eine grobe Abschätzung der Zusammensetzung der Veranstaltung als um eine eindeutige Angabe dazu wie viel Zeit für die Veranstaltung im individuellen Fall aufgewendet werden muss. Zudem sind auch aus verwandten Fächern wie z. B. der Mathematik Veranstaltungen bekannt, die als Angebote über die vorgesehenen Semesterwochenstunden hinaus gehen. So werden häufig neben Vorlesungen und Übungen noch Tutorien angeboten, die ebenfalls bei der Bearbeitung von Aufgaben helfen sollen und nicht in die Semesterwochenstunden einberechnet werden. Daher kann erwartet werden, dass das Durcharbeiten der Lessons die eigentlichen Vorlesungsstunden zuzüglich Vor- und Nachbereitung dieser ersetzt und trotzdem bei Bedarf ein Besuch der Livesitzungen erfolgen kann. Durch die benötigte Selbstverantwortung beim Besuch der Veranstaltungen und dem eigenständigen Lernen entsteht auch ein entscheidender Unterschied zu Schulunterricht, der mit der Übungsform assoziiert werden kann. Insbesondere in den ersten Semestern kann ein weicherer Übergang allerdings durchaus sinnvoll sein.

Die zwar eher selten geäußerte Kritik zum Ablauf, der insbesondere in den ersten Wochen dazu führte, dass Fragen zu Lessoninhalten teilweise erst nach Abgabe des zugehörigen Übungsblatts diskutiert werden sollten, wurde als Anlass für einen überarbeiteten Wochenplan genommen. Im Wintersemester 2022/23 erfolgt die Ausgabe des Wochenplans mit den Lessons und dem zugehörigen Übungsblatt bereits donnerstags für die Folgewoche. In dieser finden dann weiterhin montags und dienstags Live-Sitzungen statt, in denen Fragen zu den Inhalten der Lessons besprochen werden. Montags bis donnerstags finden die Coding Classes statt. Damit wird Studierenden die Möglichkeit gegeben sowohl am Ende der Vorwoche mit der Vorbereitung auf die Coding Classes zu starten, um diese am Anfang der Woche wahrzunehmen oder dies auf den Anfang der Woche zu verlagern, um die späteren Coding Class Termine zu besuchen. In der Vorstellung der Veranstaltungsorganisation wurden explizit die Unterschiede zwischen den Live-Sitzungen und Coding Classes erläutert, um Studierenden zu helfen die für sie passende Veranstaltung zu Beginn der Woche zu wählen. Damit wird den Studierenden auch bewusst offen gelassen, ob und wie viele Veranstaltungen sie besuchen, um der individuell einsetzbaren Zeit gerecht zu werden.

Insgesamt lässt sich feststellen, dass das Konzept des Flipped Classrooms in Verbindung mit einer neuen Gestaltung der Aufgaben die Studierenden aktiviert und eine Vor- und Nachbereitung der Vorlesungsinhalte durch die Materialien zum Selbstlernen häufiger stattfindet. Als optimaler Rhythmus für Abgaben haben sich wachsende Zeiträume herausgestellt, sodass zu Beginn des Kurses eine engmaschige Kontrolle stattfindet, die zum Ende aber aufgeweitet wird und dadurch offenere und umfangreichere Aufgaben ermöglicht. Mit abnehmenden Einschränkungen durch die Pandemie ist für synchrone Veranstaltungen die Präsenzform zu bevorzugen, digitale Angebote können in Maßen ergänzend eingesetzt werden.

7 Lernmaterialien

Im Rahmen des in Kapitel 5 vorgestellten Modulkonzepts werden für die Selbstlernphasen des Flipped Classroom digitale Lernmaterialien vor allem in Form von interaktiven Jupyter Notebooks eingesetzt. Zur genaueren Untersuchung der Gestaltung digitaler Lernmaterialien wurde eine Fallstudie durchgeführt, die fünf verschiedene Materialien zu einem ausgewählten Thema vergleicht. Dabei wurden Worked Examples und Live Coding, klassische Videos und interaktive Materialien sowie Videos mit unterschiedlichem Perfektionismusanspruch bei der Erstellung miteinander bezüglich des Lerntransfers sowie der Präferenzen der Lernenden verglichen. Die Darstellung dieser Studie findet sich in der Veröffentlichung [OB22].

Digitale Lernmaterialien können dabei auf verschiedenste Weisen klassifiziert werden. Wie im Kapitel 2.2 beschrieben, existieren für einige Eigenschaften insbesondere beim Videoformat bereits Empfehlungen. So sollten Videos möglichst eher kurz gehalten werden, an den Zielkontext angepasst werden und aktivierend beim Selbstlernen wirken. Ablenkende Elemente sollten in digitalen Lernmaterialien vermieden werden, sofern sie keinen besonderen Zweck erfüllen. Unter Einbezug von Gestaltungsprinzipien lassen sich außerdem weitere Elemente optimieren, wie z. B. der Einsatz und die Beschriftung von Grafiken oder die Ansicht und Vertonung.

Für die Informatik konkret existieren durch Live Coding und Worked Examples zwei besondere Arten ein Programm zu erläutern. Während beim Live Coding der Prozess explizit gezeigt wird, wobei dies sowohl in einer synchronen Veranstaltung als auch auf Video aufgenommen geschehen kann, besteht ein Worked Example aus einer fertigen, geeignet kommentierten Codelösung. Aus der veröffentlichten Studie [OB22] zeigt sich dabei, dass zwar beide Varianten ähnlich gut auf den Lerntransfererfolg wirken, Studierende jedoch eher das Live Coding bevorzugen. Innerhalb der Evaluationen zum Modul (siehe Kapitel 6) wurde das in den Veranstaltungen eingesetzte Live Coding ebenfalls stets positiv bewertet.

Ein weiterer betrachteter Aspekt liegt im Perfektionismusanspruch bei der Aufnahme, den viele Lehrende anstreben. Dabei sollte jedoch beachtet werden, dass auch das Auftreten in den vortragsartigen Teilen synchroner Veranstaltungen nicht kontrolliert werden kann und Fehler sowohl inhaltlicher als auch gestalterischer Art passieren können. Lehrende sollten sich dabei bewusst werden, dass kleine Fehler in der Sprache oder nicht perfekt erscheinende Bewegungen häufig gar nicht explizit von den Lernenden wahrgenommen werden, da diese sich auf den Inhalt fokussieren. Erkenntnisse aus der Forschung zur Wirkungen von Veränderungen in Filmen unterstützen diesen Erklärungsansatz [LMIS00]. Insgesamt deutet bisherige Forschung in Verbindung mit der durchgeführten Studie darauf hin, dass Authentizität und Enthusiasmus eine wichtigere Rolle für den Erfolg von Lernmaterialien einnehmen als eine scheinbar perfekte Gestaltung durch vielfache Aufnahmen, ausgiebigen Videoschnitt oder professionelle Aufnahmetechnik.

Das Fach Informatik scheint durch die natürliche Nähe zu digitalen Werkzeugen gut dafür geeignet zu sein, um in diesem digitale Lehrformate auszuprobieren [OBK21]. Dies hat sich auch im Kontext der Projekte, im Rahmen dessen das Modulkonzept (siehe Kapitel 5) entwickelt wurde, gezeigt. Aus den fachbezogenen Projekten wie InClassPraxis ist das Projekt UOS.DLL [Fri21] entstanden. Dies bündelt die Erfahrungen der fachbezogenen Projekte und soll umfassend digitale Lehre unterstützen. Konzepte und Erfahrungen aus Veranstaltungen, in denen durch Projekte Innovationen in der Lehre ermöglicht wurden, können dadurch auch in andere Bereiche weitergetragen werden.

Im Rahmen des *Constructive Alignment* [Big96], das bedeutet die Lernziele, Lernprozesse und Prüfungen aufeinander abzustimmen, gehören zum Erlernen von Kompetenzen im Bereich der Informatik auch kompetenzorientierte Prüfungen. Neben digitalen Lehrmaterialien wurde daher im Rahmen der Entwicklung der Veranstaltung auch die Strategie von Open-Books-Klausuren verfolgt, um kompetenzorientierte Prüfungen in digitalen Lehrszenarien angemessen umsetzen zu können. Diese erlauben den Studierenden in der Prüfung die Verwendung von allen verfügbaren Materialien, sofern keine Kommunikation mit anderen Personen stattfindet. Dadurch werden Aufgaben, die lediglich Faktenwissen abfragen und bisher mit reinem Auswendiglernen beantwortet werden konnten, überflüssig und der Fokus liegt klar auf der Anwendung und höheren Kompetenzstufen. Durch die Bearbeitung an Computern ermöglichen sie auch das Lösen von Programmieraufgaben an Computern, wobei zeitliche Einschränkungen beachtet werden müssen. Grundsätzlich deuten jedoch erste Studienergebnisse darauf hin, dass beim Programmieren auf Computern und der damit verbundenen Möglichkeit Programme auszuführen und deren Ergebnisse zu analysieren bessere Lösungen erreicht werden können [LLH17] und diese Lösungsweise von den Studierenden bevorzugt wird [MA16].

Im Rahmen der Überarbeitung einer bereits fertigen klassischen Klausur mussten jedoch nur wenige Aufgaben tatsächlich umfangreich angepasst werden [OBK21]. Dabei existieren verschiedene Strategien, um Aufgaben angemessen zu gestalten und Täuschungsversuche zu vermeiden. Eine Gestaltungsregel besteht darin keine häufig eingesetzten Verfahren anzuwenden oder diese nicht erkennbar zu benennen. Für einige bekannte Algorithmen existieren Anwendungen, die Nutzereingaben individuell verarbeiten können. Durch die geeignete Gestaltung kann vermieden werden, dass solche online verfügbaren Tools zur Lösung genutzt werden. Eine zentrale Bedeutung hat darüber hinaus die nur begrenzt verfügbare Zeit. Wenn dabei die zu erreichenden Punkte mit der dafür benötigten Zeit abgewogen werden, kann daraus geschlossen werden wie hoch der Aufwand sein darf, um die Aufgabe mittels Recherche zu lösen. Ist letzterer groß genug, können zwar die Punkte an dieser Stelle möglicherweise erreicht werden ohne über die entsprechenden Kompetenzen zu verfügen. Jedoch fehlt dann an anderer Stelle die Zeit für die reguläre Bearbeitung der Aufgaben, sodass insgesamt eine schlechtere Bewertung folgt. Um Rechercharbeiten und die Anwendung von Rechnern bei Aufgaben, die lediglich händisch erledigt werden sollen, zu erschweren, können Bilder von Texten wie z. B. Code verwendet werden. Diese erlauben es nicht den darauf vorhandenen Text einfach zu kopieren, sondern verlangen in der Regel das mühsamere Abtippen, sofern keine Umgehungen über Smartphonekameras oder ähnliche weitere Werkzeuge verwendet werden. Vor dem Hintergrund der Entwicklung im Bereich der künstlichen Intelligenz und damit betriebener Chatanwendungen sollte jedoch darauf geachtet werden, dass

diese in Zukunft nicht auch solche Aufgaben lösen können, die bisher nicht durch einfache Recherche gelöst werden konnten.

Zur Vermeidung von Täuschungsversuchen eignen sich Klausurvarianten. Dabei reicht es für viele Fälle bereits aus für jede variierbare Aufgabe lediglich zwei Varianten zu konstruieren und daraus geeignet viele Permutationen zu erstellen. Studierende, die während der Klausur mit anderen Personen kommunizieren, stehen dadurch zusätzlich unter Zeitdruck und beachten besonders Details in Formulierung und Benennung häufig nicht. Dadurch besteht eine teilweise bessere Kontrollmöglichkeit als in klassischen Klausuren, in denen nach Erfahrung der Lehrenden viele Täuschungsversuche unentdeckt bleiben. In den vier durchgeführten Klausuren in den Jahren 2021 und 2022 wurden in jeder Klausur mehrere Täuschungsversuche eindeutig nachgewiesen. Der Anteil Studierender, die einen solchen Täuschungsversuch begangen haben, lag dabei zwischen 1,67 % und 3,38 % bei einem Durchschnitt von 2,26 % pro Klausur und bewegt sich dabei im niedrigen Bereich. Im Gegensatz dazu blieb es bei klassischen Klausuren, in denen keine Hilfsmittel erlaubt waren und die Studierenden bis auf Toilettengänge beaufsichtigt wurden, in der Regel lediglich bei Ermahnungen, obwohl davon auszugehen ist, dass auch in dieser Situation Täuschungsversuche stattgefunden haben.

Auch für Open-Books-Klausuren bietet die Informatik geeignete Ausgangsbedingungen [OBK21]. Insbesondere bei Programmieraufgaben und Anwendungen von Verfahren auf verschiedenen Beispiele sind ohne großen Aufwand viele verschiedene Instanzen und Varianten konstruierbar. Durch gut standardisierbare Antwortformate ist auch automatisierte Korrektur in diesem fachlichen Kontext für viele Aufgaben sinnvoll anwendbar.

Teil IV

Diskussion und Ausblick

8 Diskussion

In der vorliegenden Arbeit wurden ein digital unterstütztes Konzept für ein Einführungsmodul der Informatik unter Berücksichtigung heterogener Gruppen sowie weitere begleitende Studien vorgestellt. Diese sind in einem festgelegten Rahmen entstanden, wodurch sich Limitierungen ergeben. Diese zu beachtenden Einschränkungen der Übertragbarkeit sowie die Ergebnisse selbst werden in diesem Kapitel diskutiert.

8.1 Limitierungen

Als zentrale Limitierung der vorgestellten Ergebnisse ist zu berücksichtigen, dass das beschriebene und evaluierte Konzept sowie die weiteren Studien innerhalb einer Universität entwickelt wurden und dadurch konkrete Bedingungen als nicht kontrollierbare Faktoren Einfluss haben.

Ein Teil dieser Bedingungen ergibt sich durch die Vorgaben der Prüfungsordnung. Da innerhalb eines Moduls sowohl das Programmieren als auch Algorithmen und Datenstrukturen thematisiert und gelehrt werden müssen, ergab sich eine zusätzliche Herausforderung. Auch die Vorgaben des Umfangs der Lehrveranstaltungen, die Personalsituation sowie die Integration des Moduls innerhalb der Studienpläne beeinflussten Entscheidungen bei der Konzeptionierung sowie die Ergebnisse der Umgestaltung.

Das vorgestellte Konzept ist daher für den konkreten Fall optimiert und nicht allgemeingültig auf alle Situationen übertragbar. Es kann jedoch Anhaltspunkte bieten Teile des Konzepts an die eigene Situation angepasst weiterzuentwickeln.

Grundsätzlich sind auch die Bedingungen im Bildungssystem zu beachten. Zum einen betrifft dies den Status des Fachs Informatik im Schulsystem, zum anderen aber auch grundsätzliche formale Vorgaben für Hochschulen. Innerhalb des Schulsystems hat Informatik zum Zeitpunkt der Arbeit einen Status als Wahl- bzw. Wahlpflichtfach [SHF21]. Jedoch ist eine Entwicklung erkennbar, da eine zunehmende Zahl an Schulen Informatikunterricht anbietet und einige Bundesländer auch verpflichtenden Informatikunterricht zumindest für ein Schuljahr planen [SHF22]. Dennoch kann davon ausgegangen werden, dass noch über eine längere Zeit keine wesentlichen Vorkenntnisse bei Studienanfängern erwartbar sind. Die Inhalte und Herausforderungen durch die Diversität der Studierenden werden sich daher auf absehbare Zeit, abgesehen von den grundsätzlichen Entwicklungen im Fach wie z. B. neuen Programmiersprachen(versionen), nicht maßgeblich verändern.

Bei den Rahmenbedingungen für Lehrmodule müssen die Bedingungen der Hochschulen durch Prüfungsordnungen, Zeiten sowie Personal beachtet werden. Diese können diverse Auswirkungen haben. So ist es beispielweise nicht an jeder Hochschule möglich digitale Prüfungen durchzuführen. Auch vorgegebene Semestertermine und Kalenderbesonderheiten können Abweichungen in Plänen erforderlich machen, wenn Ferien und Feiertage die

zeitliche Wochenplanung beeinflussen und dadurch Termine ausfallen. Auch thematische Ausrichtungen können durch Jahreszeiten bei unterschiedlichen Semestern beeinflusst werden. Das eingesetzte Gamificationkonzept im vorgestellten Modul (siehe Kapitel 6.1) basiert auf einer Durchführung des Moduls im Wintersemester, indem weihnachts- und winterbezogene Einkleidungen und Belohnungen gewählt wurden. Eine wichtige Voraussetzung für die Umgestaltungsphase eines Moduls ist außerdem eine ausreichende Personalverfügbarkeit. In dem beschriebenen Modul wurde dies vor allem durch Projekte zur Verbesserung der Lehre ermöglicht. Besonders in den ersten Durchgängen steigt der Aufwand für die Erstellung neuer Materialien stark und relativiert sich erst mit mehrfacher Nutzung dieser. Jedoch sollte bei der Konzeptionierung insgesamt darauf geachtet werden, dass die geplanten Veranstaltungen personell durchführbar sind. Im beschriebenen Konzept hatten die personellen Vorgaben Einfluss auf die Gruppengrößen der Testatgruppen sowie auf die Anzahl angebotener Coding Class Termine.

Für einen internationalen Vergleich wären auch das grundsätzliche deutsche Bildungssystem und die Präsenz fortschrittlicherer Lehrkonzepte in diesem zu berücksichtigen. Ein großer Teil der zugrundeliegenden Literatur für den Einsatz von Flipped Classroom Szenarien bezieht sich auf andere Länder, in denen dieses Konzept teilweise bereits weiter verbreitet ist und daher möglicherweise auch von Lernenden schneller akzeptiert und verinnerlicht wird. Auch die grundsätzliche Lernkultur, übliche Gruppengrößen in den Lehrveranstaltungen und Rahmenbedingungen des Studiums selbst haben Einfluss auf diverse Aspekte wie die Beziehung zwischen Studierenden und Lehrenden, die Möglichkeiten für Aktivitäten sowie die Zeitbudgets der Studierenden.

Ein weiterer zu berücksichtigender Aspekt ist die Corona-Pandemie, die das Konzept ab dem zweiten Jahr sowie die meisten Ergebnisse der weiteren Studien maßgeblich beeinflusst hat. Obwohl dadurch einige insbesondere digitale Bestandteile sicherlich von einer größeren Akzeptanz profitierten, müssen Auswirkungen der Pandemie auch als Einschränkung der Übertragbarkeit der Ergebnisse betrachtet werden. Die Studierenden hatten im Gegensatz zum Studium vor der Pandemie in dieser Zeit zusätzliche Schwierigkeiten, da sie häufig nur digitale Lehrangebote nutzen konnten und durch Maßnahmen zur Pandemieeindämmung auch die sonst üblichen und gewünschten Kontakte unter Studierenden nicht mehr einfach umsetzbar waren. Rein digitale Angebote bieten dabei zwar Vorteile wie die örtliche Unabhängigkeit und oft einfachere zeitliche Planbarkeit, erfordern allerdings eine angemessene technische Ausstattung und können durch lange Bildschirmzeiten als anstrengend wahrgenommen werden [SFL20]. Außerdem ist in der ersten digitalen Umsetzung noch der Effekt des Emergency Remote Teaching [HML⁺20] zu beachten, durch den Studierende Schwierigkeiten in der digitalen Lehre zunächst aufgrund der besonderen Situation stärker toleriert haben könnten.

Zwischen dem Wintersemester 2020/21 und dem Wintersemester 2021/22 bestanden wiederum Unterschiede in den Beschränkungen, aber auch in den Ankündigungen zu den erlaubten Kontaktmöglichkeiten und der dahingehenden Stabilität in der Lehre. Während das Wintersemester 2020/21 als vollständig digitales Semester angekündigt und auch so durchgeführt wurde, wurde ein Jahr später bereits eine beschränkte Präsenz ermöglicht. Dabei mussten Regeln zum Infektionsschutz eingehalten werden, die insbesondere das Tragen von Masken, Nachweise von Präventionsmaßnahmen und Abstände zwischen den Personen betrafen. So durften an den Coding Classes in dem speziell

dafür eingerichteten Raum nur 20 Studierende pro Termin teilnehmen. Dies machte eine Anmeldung erforderlich und schloss Personen, die dabei nicht schnell genug oder zur Anmeldezeit verhindert waren, aus. Außerdem könnten sich einzelne Studierende aufgrund der geltenden Regeln oder der Befürchtung einer Infektion in dieser besonderen Situation gegen eine Teilnahme an einer Präsenzveranstaltung entschieden haben. Ab Mitte Dezember wurden aufgrund der Pandemie die Kontaktmöglichkeiten beschränkt und die angekündigte Teilpräsenz musste abgesagt werden. Durch diese Maßnahme wurde Studierenden, die das Angebot der Präsenzlehre gerne wahrgenommen haben, wieder genommen. Es kann positiv auf digitale Veranstaltungen wirken, wenn die Studierenden sich aus vorherigen Veranstaltungen persönlich kennen und dadurch im digitalen Raum weniger Schwierigkeiten bei der Kontaktaufnahme haben. Im vorliegenden Fall ist dieser Effekt jedoch nicht eingetreten und es wurde beobachtet, dass auf den Besuch der Coding Class häufig gänzlich verzichtet wurde.

Als grundsätzliche Limitierungen empirischer Evaluations- und Fragebogenstudien in der Lehre sind Biaseffekte zu berücksichtigen. So nehmen durch die Freiwilligkeit häufiger motivierte oder extrem zufriedene bzw. unzufriedene Personen an den Befragungen teil, während unentschlossene eher darauf verzichten. Durch den Kontext der Durchführung ergibt sich außerdem eine grundsätzlich beschränkte Zahl an Teilnehmenden, sodass einige Ergebnisse nur mit relativ geringer Teilnehmendenzahl vorliegen.

Konkrete weitere Limitierungen, die die einzelnen Studien betreffenden, sind außerdem in den jeweiligen Veröffentlichungen erläutert und diskutiert worden.

8.2 Diskussion der Ergebnisse

Im Rahmen dieser Arbeit wurde ein iterativ entwickeltes Modulkonzept vorgestellt, das mehrere aus der Literatur als überwiegend erfolgreich bewertete Konzepte miteinander vereint. Dabei ergänzen sich ein Flipped Classroom mit Ansätzen offener Projektaufgaben und der Idee von Gamification zu einem Modell, das Angebote für diverse Studierende bereit hält und diese aktiv einbezieht.

Wie die Ergebnisse der Evaluationen in Verbindung mit bisherigen Forschungsergebnissen insbesondere zum Flipped Classroom zeigen, müssen die Studierenden zunächst mehr Zeit für die aktive Mitarbeit einbringen und durch Selbstregulation die Materialien selbstständig bearbeiten [BS12, Mor19]. Bei näherer Betrachtung zeigt sich, dass zum einen erst durch die Materialien zum Selbstlernen die eigentlich vorgesehene Zeit investiert wird. Zum anderen kann der Zeitaufwand für begleitete Aufgaben hilfreich sein, um den Zeitaufwand für die eigenständig zu bearbeitenden Aufgaben zu verringern.

Eine Stärke der Arbeit besteht darin, dass ein sehr konkretes Modulkonzept vorgestellt wurde, dessen detaillierte Entwicklung ebenfalls nachvollziehbar ist. Dies wird in der bisherigen Literatur teilweise kritisiert [GKC14], weil einige Autoren sich stark auf die Evaluationsergebnisse oder weitere stark forschungsbezogene Ergebnisse fokussieren und dabei nur oberflächlich das Konzept selbst beschreiben. Zudem werden negative oder nicht signifikante Ergebnisse häufig nicht publiziert [DB16, S. 901]. Dabei konnten auch aus dem ersten Jahr der Umstellung, in dem die Ergebnisse nicht ausschließlich als

positiv bewertet werden können, einige Folgerungen gezogen werden, die für die weitere Konzeptentwicklung von Vorteil waren.

In Anbetracht der in Kapitel 5 erläuterten Ziele der Modulgestaltung und der Ergebnisse kann festgestellt werden, dass ein Großteil der Ziele zumindest teilweise erreicht wurde. Die Teilnehmenden der Übung arbeiten im neuen Konzept aktiv und gemeinsam an Aufgaben, die für die folgenden Abgabenaufgaben hilfreich sind und tauschen sich dabei zu den Inhalten und der Anwendung dieser aus. Die Übung bietet damit wieder für diverse Vorkenntnisstufen interessante Inhalte und einen Mehrwert durch die Anwesenheit von Übungsleitenden und Tutoren für direkte Hilfe bei der Anwendung der Inhalte. Die Integration eigener mobiler Geräte wird dabei durch die Raumeinrichtung unterstützt und ermöglicht sowohl Studierenden mit als auch ohne Notebook bequem während der Coding Class Lösungsideen auszuprobieren und Rückmeldungen zu erhalten. Nach dem ersten Durchgang mit weniger Studierenden, die das Testsystem erfolgreich bestanden haben, entwickelten sich die Statistiken zum Erfolg der Aufgaben und dem Sichtbarmachen des Programmierprozesses eher positiv. Vor dem Kontext der Open Books-Klausur mit etwa konstanten Ergebnissen in der Benotung im Vergleich zu den bisherigen klassischen Klausuren liegt die Vermutung nahe, dass die tatsächlichen Programmierkompetenzen der Studierenden gestiegen sind. Ein Ausgleich geringer anwendungsbezogener Kompetenzen durch einfache Aufgaben niedriger Kompetenzniveaus, die bisher im Wesentlichen durch Auswendiglernen zu bewältigen waren, ist im neuen Klausurformat nicht mehr möglich.

Durch die in den Evaluationen angegebenen Zeiten für die Arbeit mit den interaktiven Materialien und die daraus resultierende beobachtete Vorbereitung auf die Coding Classes ist davon auszugehen, dass im neuen Konzept eine intensivere Auseinandersetzung mit den Inhalten stattfindet. Die integrierten Aufgaben mit geringem Umfang stellen dabei die Vor- und Nachbereitung der vor allem in den Slideshows erklärten Inhalte dar. Zudem ist rein physische Anwesenheit ohne Mitarbeit zwar in einigen Teilen der Veranstaltung weiterhin möglich, beim grundlegenden Erfassen der Inhalte und in den Coding Classes wird allerdings zumindest aktives Mitdenken gefordert, da bis auf kleine Anteile der Veranstaltungen und Aktivitäten anderer keine rein passiv aufnehmbaren Inhalte mehr existieren.

Der Personalbedarf für die Durchführung der Veranstaltung stieg zwar zum Teil durch das Angebot von mehr Übungsterminen, wurde jedoch durch Maßnahmen im Bereich der Testate gesenkt, ohne die engmaschige Betreuung aufzugeben. Die leicht erhöhte Gruppengröße für die Abgabenaufgaben erwies sich als angemessen. Die zunächst angedachten durchgehend auf drei Wochen erweiterten Abgabezeiträume stellten sich zu Beginn als problematisch heraus, da die angestrebte regelmäßige Auseinandersetzung mit den Inhalten damit nicht gewährleistet werden konnte. Im veränderten Konzept wurden die Vorteile beider Rhythmen miteinander vereint. Dadurch wurde zu Beginn ein strikter Rhythmus geschaffen, der das kleinschrittige Lernen insbesondere für das Programmieren unterstützt, aber zum Kursende hin umfangreichere Aufgaben ermöglicht, die Kreativität und Realitätsnähe begünstigen. Durch die Gruppengröße kann dabei ein Drittel der Tutorenstunden eingespart werden, durch den neuen Abgaberrhythmus zusätzlich 41,67 %, da nur noch sieben statt bisher zwölf Testate pro Kleingruppe stattfinden. Insgesamt ergibt sich dadurch eine Einsparung bei den Tutorenstunden um

über 60 %. Dem gegenüber werden etwa 25 bis 30 % der Tutoren für die Coding Classes benötigt, sodass insgesamt noch eine Einsparung von ca. 30 bis 35 % vorliegt.

Die Pandemie, die die Entwicklung des Konzepts ab dem zweiten Jahr begleitet hat, ist sowohl als Limitierung (siehe Kapitel 8.1) zu sehen, aber auch als Katalysator für die Entwicklung der Materialien zum Selbstlernen [OBB⁺22] und damit einem vollständigen Flipped Classroom. Im ersten Jahr wurde ein partieller Flipped Classroom mit Vorlesung und aktiver Übung sowie zusätzlichen Materialien zur Vorbereitung auf diese umgesetzt. Viele Studierende sahen ihre Anwesenheit in der Vorlesung als ausreichend für die Vorbereitung auf die Übung an und bearbeiteten die Materialien daher nicht. Nahezu vollständig unvorbereitete Gruppen erschweren dabei massiv die Arbeit in den synchronen Terminen. Der vollständige Flipped Classroom trug zur Lösung dieses Problems bei.

Aus der Literatur [KP14, KNH⁺15a, KNBH16] und den Überlegungen zum konkreten Konzept entstanden zunächst zwei Grundregeln für die synchronen Termine, die jedoch im Verlauf an die reale Situation angepasst und dadurch gelockert wurden. Eine dieser Regeln verbietet dabei die Wiederholung vorbereitender Inhalte insbesondere zu Beginn synchroner Sitzungen, um den Teilnehmenden zu signalisieren, dass ihre möglicherweise fehlende Vorbereitung nicht während der synchronen Termine aufgeholt werden kann. Dies soll, eher extrinsisch, dazu motivieren die Inhalte tatsächlich vorab ausreichend gut zu bearbeiten, um sie zum synchronen Termin abrufen zu können, bevor die Anwendung stattfindet. Besonders im ersten Jahr der Umstellung führte dies allerdings dazu, dass unvorbereitete Studierende aufgrund fehlender Anwesenheitspflicht einfach auf den weiteren Besuch von Coding Classes verzichteten, da sie diese als für sich nicht sinnvoll empfanden und die Abwesenheit eine einfachere Lösung darstellte als die Beschäftigung mit den Materialien. Für einige Studierende war trotz Beschäftigung mit den Inhalten die Einstiegshürde der direkten Anwendung weiterhin zu hoch und die Möglichkeit Fragen zu den Inhalten zu stellen wurde ebenfalls kaum genutzt.

Ab dem zweiten Jahr begannen die Coding Classes daher mit einem Intro, bei dem interaktiv die wichtigsten Inhalte so wiederholt wurden, dass zwar eine zumindest oberflächliche Vorbereitung nötig war, aber Studierende, die noch Schwierigkeiten damit hatten das Gelernte anzuwenden, dennoch gewinnbringend teilnehmen konnten. Häufig wurde dabei nach der Abfrage einer Definition für die wichtigsten neuen Begriffe eine einfache Anwendungsaufgabe gemeinsam gelöst oder eine der für die Coding Class konzipierten Aufgaben gemeinsam begonnen. Dadurch bestand gleichzeitig ein Anknüpfungspunkt für die Bearbeitung des Aufgabenblatts. Zusätzlich hilfreich war die Ersetzung der klassischen Vorlesung durch die Materialien zum Selbstlernen. Dadurch war die Vorbereitung auf die Coding Classes direkt in der inhaltlichen Basis integriert und durch die enthaltenen kleinen Aufgaben zur Reflexion der Inhalte konnte eine erste Anwendung erfolgen. In einigen Fällen bestand dadurch bereits eine Gelegenheit Fragen konkret formulieren zu können, wenn eine der Aufgaben in den Materialien nicht wie erwartet gelöst werden konnte. Insgesamt ist diese Empfehlung, die in der Literatur als zentral für einen funktionierenden Flipped Classroom beschrieben wird, daher im Verlauf des Moduls leicht abgewandelt worden, da der universitäre Kontext und die damit verbundene Freiwilligkeit beim Besuch von Veranstaltungen vor allem die intrinsische und weniger die extrinsische Motivation ansprechen sollte.

Als zweite Regel wurde im initialen Konzeptentwurf darauf geachtet, dass vor allem die Zusammenarbeit der Studierenden untereinander und die eigenständige Arbeit dieser gefördert werden. Daher wurden möglichst geringe konkrete Antworten gegeben, sondern häufig bereits die Aufgaben so konzipiert, dass die Studierenden Sonderfälle bei der Programmierung entdecken sollten. Dies stellte sich jedoch besonders zu Beginn als zu schwierig heraus und überforderte viele Studierende schnell, sodass auch diese keinen ausreichenden Mehrwert im Besuch der Veranstaltung sahen. Ähnlich wirkten die durchgehend dreiwöchigen Bearbeitungszeiträume im ersten Konzept. Da diese besonders zu Beginn durch den langen Zeitraum bis zur ersten Kontrolle zu viele Freiheiten erlaubten, entzogen sich einige Studierende der regelmäßigen Mitarbeit. Sie begannen erst zum Ende dieser Zeiträume mit der tatsächlichen Bearbeitung der Aufgaben und verpassten dadurch den Anschluss. Die im Semesterverlauf wachsenden Abgabezeiträume stellten sich dabei als effektive und sinnvolle Lösung dieses Problems heraus, da sie sowohl eine engmaschigere Kontrolle zu Beginn ermöglichten und die Studierenden gezwungen waren parallel zu den Inhalten auch die Anwendung zu üben, als auch zum Ende des Kurses offene Aufgaben mit längerem Bearbeitungszeitraum ermöglichten. Bis zu diesem Zeitpunkt hatten die Studierenden zum einen die regelmäßige Arbeit an den Inhalten und der Anwendung dieser verinnerlicht. Zum anderen wurden im Projekt, dessen ausreichend gute Bearbeitung für die Zulassung zur Klausur benötigt wird, keine neuen Inhalte mehr abgefragt, da dieses als Programmierprojekt angelegt ist und parallel zu den restlichen eher theoretischen Inhalten zu Datenstrukturen, wie Bäumen und Graphen, bearbeitet wird. Diese Inhalte werden allerdings in der Klausur explizit thematisiert und in den Coding Classes geübt, sodass die Coding Classes hier eine besondere Rolle einnehmen.

Die Ergebnisse der Studie zur Frustrationstoleranz (eingereicht) weisen darauf hin, dass mit der Nutzung psychologischer Begriffe als angestrebte Kompetenz oder Voraussetzung vorsichtig umgegangen werden sollte. Insgesamt zeigt sich auch durch die Betrachtung der Forschungslage, dass die Fähigkeiten, die in einem Informatikstudium von Vorteil sein könnten und die überfachlichen Kompetenzen, die erlernt werden sollen, bisher noch nicht ausreichend erforscht sind. Die konkreten Ergebnisse der einzelnen Studien werden in den Veröffentlichungen diskutiert.

9 Folgerungen

Aus den geschilderten Ergebnissen ergibt sich eine Reihe von Folgerungen sowohl für die Zukunft des Moduls, das sich aufgrund äußerer Umstände weiter verändern wird, aber dennoch Teile des Konzepts fortführen könnte, aber auch für generelle Anwendungsszenarien der betrachteten Aspekte in der Lehre. Diese werden im Folgenden erläutert.

9.1 Folgerungen für Modul

Für das Modul *Einführung in Algorithmen und Datenstrukturen* ist in einem iterativen Ansatz ein Konzept entstanden, das durch die Verbindung von Flipped Classroom, wachsenden Bearbeitungszeiträumen für Abgabearbeiten und Gamification den Lernenden ein breites Angebot ermöglicht und sie gleichzeitig zur aktiven Mitarbeit animiert. Obwohl aufgrund von Änderungen auch im Prüfungsformat keine gesteigerten Leistungen in diesen nachweisbar sind, ergibt sich aus den Leistungen in dem stärker kompetenzorientierten Format sowie aus den Evaluationsergebnissen und Tutoreninterviews ein insgesamt guter Erfolg.

Durch die Entwicklungsschritte ist zu sehen, dass bedingt durch die Pandemie vor allem das Konzept zum Wintersemester 2020/21 die entscheidendsten Veränderungen gebracht hat, wobei das erste Konzept im Wintersemester 2019/20 als grundsätzlicher Beginn der Veränderung ebenfalls einen hohen Stellenwert einnimmt. In den folgenden Semestern betrafen die Anpassungen nur noch Details wie die teilweise Rückkehr zur Präsenz (Wintersemester 2021/22) oder veränderte Wochenpläne (Wintersemester 2022/23). Damit lässt sich feststellen, dass das Grundkonzept im Wesentlichen in zwei Entwicklungszyklen entstand und sich zu einem gut funktionierenden Modulkonzept entwickelt hat.

Obwohl das Modul aufgrund einer veränderten Studienordnung ab dem Wintersemester 2023/24 nicht mehr in der gleichen Form weitergeführt wird, ergeben sich aus den Erfahrungen und der Begleitforschung einige Aspekte, deren Fortführung in den Nachfolgemodulen erwägt werden sollte. Trotz geringerer erwartbarer Teilnehmendenzahlen in diesen werden sich auch dort einige der Herausforderungen wiederfinden, die die Umgestaltung des aktuellen Moduls motiviert haben. Durch die Aufteilung in ein Modul für stark und weniger informatikbezogene Studienfächer sowie für die stark informatikbezogenen Fächer in ein Programmier- und ein Algorithmenmodul können einige Schwierigkeiten im Umgang mit der starken Heterogenität der Studierenden bereits strukturell abgefangen werden. Für die inhaltliche Gestaltung des anwendungsbezogenen Moduls sollte die Gestaltung eines Einführungsmoduls von Malan [Mal10] beachtet werden, die einige Themen umfasst und durch einen grafikbasierten Einstieg auch Programmierunerfahrenen einen angenehmen Start ermöglicht. Auf Dauer wird auch die schrittweise Einführung von Informatik als Pflichtfach in einigen Bundesländern zu einem

zumindest geringen Vorwissen bei einem höheren Anteil Studierender führen. Dennoch ist davon auszugehen, dass weiterhin auch im Modul für stark informatikbezogene Studienfächer Studierende mit sehr geringen Vorkenntnissen auf Studierende treffen, die die Modulinhalte bereits zum Teil kennen. Im Modul für weniger informatikbezogene Studierende würde zum einen weiterhin die hohe Themenvielfalt bestehen, zum anderen können geringere Personalkapazitäten herausfordernd sein.

Bezüglich der Personalkapazitäten ergeben sich aus den Ergebnissen der Modulumgestaltung mehrere Überlegungen. Zunächst wurden verschiedene Wege gefunden, um die Anzahl der beteiligten Tutoren verringern zu können. Ein Weg war die Reduktion der Anzahl der Testate durch die Erweiterung des Bearbeitungszeitraums. Dies hat sich jedoch erst nach einigen Wochen mit dichterem Überprüfen des Lernfortschritts als tragfähig und sinnvoll erwiesen. Die zweite Möglichkeit war die Erhöhung der Gruppengröße auf drei statt wie bisher zwei Studierende. Diese Gruppengröße wurde als geeignet wahrgenommen. Eine weitere Erhöhung der Zahl an Gruppenteilnehmenden ist nicht als problemlos anzusehen, da in der Testatsituation bereits vier Personen gleichzeitig eine Lösung betrachten und eine weitere Erhöhung zu einer zunehmenden Anonymisierung innerhalb der Gruppe führen kann, bei der einzelne Gruppenmitglieder kaum zum Ergebnis beitragen. Insgesamt sollten die Gruppengrößen in den einzelnen Veranstaltungsteilen berücksichtigt werden und das Personal bedarfsgerecht fokussiert werden. Um der Heterogenität der Studierenden gerecht zu werden, sollte allerdings ein möglichst vielfältiges Angebot beibehalten werden. Das Angebot kann allerdings, neben betreuten Lernzeiten, auch daraus bestehen lernbegünstigende Räume zum Selbstlernen insbesondere in Kleingruppen bereitzustellen. Dabei hat sich der Raum, der für die Coding Classes eingerichtet wurde, wie auch ähnliche Räume in der Literatur, als hilfreich erwiesen. In klassisch eingerichteten Seminarräumen wäre ein häufig nicht erlaubtes, in Vorlesungsräumen sogar unmögliches Verschieben der Tische und Stühle notwendig, um eine angenehme Kombination aus den Möbeln für die Arbeit in Kleingruppen zu erreichen.

Beim Lehrpersonal auf Ebene der Übungsleitenden kann aus den Erfahrungen das Brooksche Gesetz [Bro19] zum Teil bestätigt werden. Dieses besagt, dass bei Software-Projekten, die nicht rechtzeitig fertiggestellt werden können, der Einsatz von zusätzlichen Personen negative Auswirkungen haben kann, da diese sich zunächst einarbeiten müssen [Bro19, S. 26]. Insbesondere bei spontan auftretenden Schwierigkeiten dauert es bei einer ausreichend großen Zahl an Übungsleitenden teilweise einige Zeit, bis alle über den gleichen Stand an Informationen verfügen. Allerdings sind für die Erstellung neuer Materialien, insbesondere auch Übungsaufgaben, personelle Kapazitäten notwendig. Grundsätzlich hat es sich als hilfreich erwiesen, wenn während der Vorlesungszeit bereits Aufgaben vorhanden waren, die nur nochmal überarbeitet und den aktuellen Inhalten zugeordnet werden mussten. Für die Erstellung vollständig neuer Aufgaben ist die verfügbare Zeit der bisher üblichen zwei Übungsleitenden während der Vorlesungszeit kaum ausreichend. Bedenken bezüglich der Wiederholung von Aufgaben haben sich als unproblematisch erwiesen, da trotz Wiederholung nur sehr wenige Studierende auf vorhandene Beispiellösungen aus vorherigen Jahren zugegriffen haben (siehe Kapitel 6). Dies geschah zum einen häufig aus der Not heraus keine eigene, als ausreichend gut bewertete Lösung zu finden und zum anderen war hier auch eine Kopie von Lösungen

anderer Gruppen grundsätzlich möglich. Dies ließe sich auch mit ständig neuen Aufgaben nicht vollständig verhindern.

Grundsätzlich kann ein großer Teil der entwickelten Materialien auf diversen Wegen weiter genutzt werden. Innerhalb der nachfolgenden Module ist dabei die Nutzung der interaktiven Materialien durch Auswahl und teilweise Anpassung denkbar. Die Materialien für die eher theoretischen Inhalte eignen sich dabei vor allem im Bereich der Algorithmik, während die Materialien zum Programmieren teilweise an die verwendete Programmiersprache angepasst werden müssen. Als Open Educational Resources und für andere Kontexte sind die Materialien in der aktuell vorliegenden Fassung verwendbar.

Mit Blick auf die Zukunft ist es außerdem denkbar, dass die Vorgaben durch Universität und Prüfungsordnung von den vorgesehenen festen Stunden für Vorlesungszeit und Übungszeit abweichen könnte und damit noch offenere, an die Bedürfnisse der Studierenden angepassten, Konzepte sowie die Verwendung von Lehrzeit für alternative Angebote ermöglicht werden.

In jedem Fall bleibt es wichtig die Studierenden gerade zum Beginn ihres Studiums auf die Art zu lernen einzustellen, die im jeweiligen Konzept erforderlich ist. Durch das Verständnis des Konzepts steigert sich dabei auch das Zugehörigkeitsgefühl, was wiederum zu positiven Auswirkungen auf das Lernen führen kann [FAJ07]. Beim Übergang aus der Schule sind hier besondere Schwerpunkte darauf zu legen wie Studierende mit der fehlenden Anwesenheitspflicht umgehen können und wie Selbstregulationsstrategien beim eigenständigen Lernen helfen können. Idealerweise würden in Zukunft im Rahmen eines Studium Generale einige Lernkompetenzen auch bewusst und vor oder parallel zu fachlich-inhaltlichen Module vermittelt werden.

9.2 Generalisierung

Neben dem konkreten Modulkonzept und den Ergebnissen der Fallstudien lassen sich aus den Forschungsarbeiten einige Erkenntnisse ziehen, die auch auf andere Lehrsituationen übertragbar sind. Im Folgenden werden diese näher erläutert.

Die Bestandteile des neuen Modulkonzepts haben sich durch die iterative Entwicklung über mehrere Jahre als überwiegend positiv bei der Bewertung durch die Studierenden erwiesen. Obwohl das genaue Konzept in der Gesamtheit sicherlich nicht ohne Anpassungen an ähnliche Module in anderen Szenarien übertragen werden kann, können einige Bestandteile als Inspiration dienen und bei Entscheidungen für Gestaltungsrahmen behilflich sein.

Wie bereits in bisherigen Publikationen hat sich der Flipped Classroom als grundsätzlich dazu geeignetes Konzept gezeigt einen höheren Anteil praktischer Mitarbeit durch Studierende auch in größeren Lehrveranstaltungen zu ermöglichen. Die Einführung des Konzepts an der Universität Osnabrück und die damit verbundenen Gestaltungsentscheidungen haben Details zum Vorschein gebracht, die in der Literatur bisher wenig berücksichtigt wurden. Die gesammelten Erfahrungen und Ergebnisse könnten dabei für andere Lehrende hilfreich sein, um die verwendeten Konzepte in ihrer Veranstaltung anzuwenden. Durch die pandemische Lage konnten zudem unterschiedliche Szenarien

erprobt werden, die sich insbesondere auf die Gestaltung der synchronen Veranstaltungen auswirkten.

Eine spezifische Frage zur Gestaltung der Präsenzzeit ergab sich dadurch, dass durch die vorgesehenen sechs Kontaktstunden vergleichsweise viel Zeit zur Anwendung der in eigenständiger Arbeit erlernten Inhalte bestehen. Die zwei Wochenstunden der Übung wurden bereits im Initialkonzept zu einer aktiven Coding Class, in der an die jeweilige kleinere Gruppe angepasst die praktische Anwendung der Inhalte erfolgt. Die Vorlesung blieb dagegen bis zur zweiten Iteration bestehen. Durch die Lessons ab der zweiten Iteration wurden aber die Inhalte vollständig im Selbststudium vermittelt, sodass die vier Wochenstunden der Vorlesung noch gestaltet werden mussten. Während der Durchführungen zur Pandemiezeit als Online-Format erschien aufgrund der zeitlichen Flexibilität eine Reduktion auf die Hälfte der Zeit sinnvoll. Diese wurde als Fragestunde gestaltet und vor allem mit Fragen der Studierenden zu den Inhalten der Lessons gefüllt. Durch die Ausweitung auf zwei Termine im Wintersemester 2022/23 blieb allerdings mehr Zeit. Diese wird mit weiteren ergänzenden Hintergründen gefüllt und bietet dadurch zum einen die Möglichkeit Fragen zu den Lessons zu stellen, zum anderen erhalten interessierte Studierende weitere Einsichten und andere Betrachtungsweisen, die beim tieferen Verständnis hilfreich sein können. Es existiert allerdings in Inhalten und Gestaltung eine klare Abgrenzung zu den Coding Classes und es werden keine neuen prüfungsrelevanten Inhalte vorgestellt. Dieser Aspekt ist besonders wichtig, da die Veranstaltungen als Angebote aufgefasst werden sollen und der Besuch nicht verpflichtend ist, um den zeitlichen Aufwand nicht zu intensivieren.

Im Kontext von Hochschullehre muss die grundsätzliche Freiwilligkeit und Selbstverantwortung beim Lernen der Studierenden besonders betrachtet werden. Durch diese ergeben sich Schwankungen bei der Vorbereitung und Anwesenheit bei synchronen Veranstaltungen, die nicht immer planbar und nur teilweise beeinflussbar sind. Innerhalb des Flipped Classroom-Konzepts lautet die Empfehlung aus bisheriger Forschung, dass zur Stärkung der Vorbereitung in den Kontaktphasen keine Inhalte wiederholt werden sollen [BS12, KNBH16]. Aus den Erfahrungen und der Evaluation des Modulkonzepts im ersten Jahr wurde ersichtlich, dass ein striktes Verbot Inhalte zu wiederholen ebenfalls negative Auswirkungen haben kann. So kann ein Aufgreifen von eigentlich vorausgesetzten Inhalten zwar die Teilnahme mit geringer Vorbereitung fördern, der empfohlene Umgang jedoch die Teilnahme grundsätzlich unattraktiver gestalten. Einige Studierende haben sich in der ersten Coding Class überfordert davon gefühlt, dass ohne angemessene Vorbereitung keine Mitarbeit möglich ist und haben in Folge dessen nahezu vollständig auf die weitere Teilnahme verzichtet. Die Teilnahme an einer Veranstaltung muss einen tatsächlichen Mehrwert für die Studierenden bieten [HP16]. Daher ist eine individuelle Abwägung dazu, wie viele Inhalte doch wiederholt und welcher Umfang an Vorbereitung zwingend vorausgesetzt wird, unerlässlich. Grundsätzlich kann dann auch zusätzliche aufgewendete Zeit als sinnvoll empfunden werden [ZLM⁺09].

Die Qualität der Vorbereitung der Studierenden hat sich vor allem durch die weitere Konzeptentwicklung verbessert, in der das Material zum Selbstlernen als Basis für die Inhalte aufbereitet wurde. Dadurch hat die Vorbereitung den Stellenwert bekommen, den zuvor der Besuch der Vorlesung gehabt hat und der lediglich ergänzende Charakter wurde abgelöst. Aktive Konzepte erfordern aber grundsätzlich aktive Mitarbeit, zu der

Studierende motiviert werden müssen. Neben den fachlichen Kompetenzen kann dies für den Aufbau von Selbstregulierungskompetenzen förderlich sein [Mor19], die im Idealfall auf diverse Arbeits- und Organisationsprozesse übertragbar sind.

Grundsätzlich bleibt Zeit ein wichtiger Faktor im Studium und Studierende wägen den Einsatz dieser meist sorgfältig ab. Durch die Bologna-Reform hat sich der Druck das Studium in der vorgesehenen Zeit abzuschließen erhöht, sodass Studierende häufig viele Veranstaltungen parallel besuchen müssen und neben dem Studium häufig noch weitere Verpflichtungen wie z. B. eine nebenberufliche Tätigkeit ausführen [Mey10, GBHW12, SM11]. Dadurch entsteht ebenfalls ein Qualitätsdruck für Lehrangebote. Werden diese nicht als hinreichend nützlich wahrgenommen, besuchen entweder weniger Studierende die Veranstaltung oder sind nur noch physisch anwesend. Das letztere Verhalten kann in klassischen Frontalvorlesungen oft beobachtet werden.

Insgesamt ist Lehrenden daher zu raten vielfältige und attraktive Lehrangebote zu bieten, wobei auch die Unterschiede klar erläutert werden sollten. Wenn die Erwartungen der Studierenden angemessen erfüllt werden, ist es entsprechend des Modells aus Kapitel 4 wahrscheinlich, dass das Angebot akzeptiert und mit einem positiven Eindruck angenommen wird.

Dennoch sollten sich Lehrende bewusst darüber sein, dass es als illusorisch zu betrachten ist alle Studierenden gleichermaßen zu erreichen. Einige benötigen z. B. aufgrund von Vorkenntnissen keine speziellen Lehrangebote. Andere versuchen dagegen einen Teil der Ressource Zeit zu sparen und verpassen dabei den Moment, an dem sie Hilfe beim Verständnis der Inhalte benötigen. An dieser Stelle helfen semesterbegleitende Leistungen wie die Abgaben und Testate im Modulkonzept (vgl. Kapitel 5). Durch eine Individualisierung, die durch das Eingehen auf die konkreten Schwierigkeiten der anwesenden Studierenden erfolgt, können die verfügbaren Angebote erheblich optimiert werden und zu einer größeren Zufriedenheit unter den Studierenden führen.

Das vorgestellte Konzept ist insbesondere durch die Verwendung von Jupyter Notebooks und der dadurch entstehenden interaktiven Möglichkeiten Programme ändern und ausführen zu können stark an das Fach Informatik angepasst. Dennoch können Bestandteile des Konzepts auch auf andere Fächer übertragen werden. Für andere Module innerhalb der Informatik oder für informatiknahe Fächer ist dies besonders einfach möglich. In anderen Fachkontexten müssen Überlegungen dazu stattfinden wie in diesen interaktive Materialien gestaltet werden können, die idealerweise auch eine Rückmeldung zu den Eingaben der Studierenden beinhalten. Hier können eventuell Quizfragen oder Reflexionen mit gegenseitigem Feedback von Studierenden sinnvoll sein. Die Konzepte des Flipped Classrooms, offener projektartiger Aufgaben sowie der Gamification lassen sich grundsätzlich in allen Fächern einsetzen, erfordern aber teilweise, dass sich Lehrende und Studierende bewusst darauf einlassen, da in einigen Fachkulturen noch deutlich stärker klassische Lehrformate erwartet werden als in der Informatik (siehe auch [OBK21]).

Für den Einsatz des Konzepts in fortgeschritteneren Modulen sollten allerdings einige Überlegungen zur Anpassung erfolgen. Der überwiegende Teil der Forschung zum Flipped Classroom bezieht sich jedoch eher auf Studierende in den ersten Studienjahren. Grundsätzlich sei das Konzept aber auch für Kurse in höheren Semestern geeignet [OP15]. Die Lehrenden sollten sich bewusst darüber sein, dass Studierende in höheren Semestern

schon mehr klassische Lehre kennengelernt haben und ihnen die Anpassung an das neue Szenario gegebenenfalls schwer fallen kann. Jedoch können Erfolge und Lernerfahrungen in traditionellen Lernsettings dazu beitragen bereits einige Lernkompetenzen aufgebaut zu haben, die das eigenverantwortliche Lernen erleichtern. Es sollte darauf geachtet werden, dass die Studierenden ausführlich über das Lehrkonzept informiert werden und ihnen beim Einstieg dabei geholfen wird. Insbesondere der Wechsel der Rolle des Dozenten von einem Wissensvermittler zu einem Lernbegleiter kann dabei für einige Studierende herausfordernd sein. Außerdem sollte je nach Veranstaltungsgröße geprüft werden, welche Bestandteile des Konzepts sinnvoll einsetzbar sind und wodurch diese ergänzt werden können. Grundsätzlich sind insbesondere die Aktivitäten in Präsenz mit kleineren Gruppen besser durchführbar. Eine umfassende eigenständige Erstellung von Lernmaterialien lohnt sich dagegen eher für große Lerngruppen. Für kleine Lerngruppen oder nicht wiederholt stattfindende Module sollten daher auch bereits existente, externe Materialien gesichtet und auf ihre Eignung geprüft werden.

Als mögliche Alternative zu JupyterNotebooks für interaktive Lernmaterialien könnten auch *zyBooks* [BVL21] in Betracht gezogen werden. Diese ermöglichen ebenfalls die Einbettung von ausführbarem Code und anderen interaktiven Elementen. Erste Anwendungen weisen auf positive Ergebnisse für die Bearbeitung durch Studierende hin [EVL⁺15].

Im Gegensatz zu den von der Literatur leicht abweichenden Ergebnissen zur Wiederholung von Inhalten in synchronen Veranstaltungen hat sich bestätigt, dass das Flipped Classroom-Konzept einen hohen Aufwand für die Vorbereitung der Materialien erfordert. Dieser relativiert sich allerdings durch die mehrfache Nutzung der Materialien in verschiedenen Jahren und liegt dann, unter Einbezug der dennoch zu füllenden Präsenztermine, lediglich etwas höher als die vergleichbare Vorbereitung klassischer Lehrkonzepte. Außerdem verringert sich der Produktionsaufwand durch die bessere Verfügbarkeit von Technik und einfacher zu bedienenden Programmen zunehmend und ermöglicht deutlich mehr Gestaltungsvarianten als noch vor einigen Jahren [LP00, Cop07].

Durch die aktuellen Regelungen zu Lehrverpflichtungen ist es trotz Auslagerung der Vorlesungsinhalte auf Lernmaterialien abgesehen von pandemiebedingten Sonderregelungen nötig die exakten Stunden als synchrone Veranstaltung durchzuführen. Möglicherweise werden sich hier in Zukunft Änderungen ergeben, wenn vermehrt alternative Lehrkonzepte umgesetzt werden. Dadurch könnten möglicherweise mehr Lehrende ausreichend Zeit finden, um Lernmaterialien zu entwickeln und einen verbleibenden Teil synchroner Lehre aktiver umzusetzen.

Aus der Studie zu digitalen Lernmaterialien [OB22] hat sich im Kontext bisheriger Literatur bestätigt, dass Perfektion für Lernmaterialien nur von nachrangiger Bedeutung ist. Im Gegenteil können Fehler sogar Lerngelegenheiten bieten. Die Produktion kann dadurch insgesamt zeitlich mit geringerem Aufwand und auch mit grundlegender technischer Ausstattung erfolgen. Professionelle Aufnahmestudios werden für Lernmaterial nicht zwingend benötigt. Bei aufgenommenen umfangreicheren Fehlern sollte darauf geachtet werden, ob diese im Lernprozess hilfreich sein können und im weiteren Verlauf des Materials erläutert und gelöst werden. Dann könnten diese als Lerngelegenheit explizit thematisiert werden. Fehler von sehr geringem Umfang, wie z. B. Tippfehler oder kurze, korrigierte Syntaxfehler, können insbesondere den Programmierprozess authentisch

darstellen. Je nach konkretem Inhalt ist das Auftreten problematischer Fehler grundsätzlich weniger wahrscheinlich. In stärker theoretischen Inhalten sowie in Inhalten einiger anderer Fächer handelt es sich bei Unzufriedenheiten Lehrender mit ihren Aufnahmen häufiger um Formulierungen oder bei Aufnahme der sprechenden Person um Gestik sowie Mimik. Sofern die Formulierungen keine Fehlvorstellungen erzeugen könnten, sollten Lehrende sich dabei dem Vergleich zu klassischen Vorlesungen bewusst werden, in denen sie ebenfalls nicht perfekt agieren können und trotzdem den Lernstoff vermitteln. Gute digitale Lernmaterialien sind dabei möglichst abwechslungsreich im Format, sprechen Lernende interaktiv an und berücksichtigen unterschiedliche Erfahrungsstufen innerhalb des Zielkontexts.

Insgesamt zeigen die Ergebnisse, dass Lehrende sich trauen sollten Veränderungen in der Lehre auszuprobieren und dabei zu erwarten ist, dass die Studierenden die Bemühungen anerkennen werden. Fehler sollten dabei für Studierende explizit erlaubt sein [KNH⁺15b], da dies sowohl die Akzeptanz für Fehler Lehrender erhöhen kann als auch Lerngelegenheiten bietet. Die konkret beschriebene Gestaltung des Moduls kann dabei helfen die eigenen Anforderungen abzuwägen und das eigene Konzept entsprechend erfolgsversprechend zu gestalten. Im Rahmen weiterer Lehr-Lern-Forschung sollten auch einige betrachtete Aspekte weiter analysiert werden. So besteht insbesondere bei psychologischen Zusammenhängen mit Frustration beim Programmieren und auf Einstellungen und Erfolge wirkenden anderen Faktoren, aber auch bei der konkreten Gestaltung von Lernmaterialien zum Programmieren weiterer Forschungsbedarf.

10 Zusammenfassung

Die vorliegende Arbeit stellt den Manteltext für die kumulative Dissertation zum Thema digital unterstützter universitärer Anfangslehre in der Informatik unter Berücksichtigung heterogener Gruppen dar. Aufgrund der Heterogenität der Studierenden und der besonderen Schwierigkeit des Programmierenlernens besteht ein Optimierungsbedarf für die Lehre in Einführungsveranstaltungen der Informatik, wie in Kapitel 1 motiviert wird. Innerhalb dieses Kapitels wird auch der Aufbau der Arbeit näher beschrieben.

Zunächst wird dafür die Literatur zu den relevanten Konzepten erläutert (siehe Kapitel 2). Neben Metaanalysen der Forschung zu Einführungsmodulen in der Informatik werden dabei insbesondere die Konzepte des Flipped Classrooms, der Projektarbeit und Gamification betrachtet. Beim Flipped Classroom wird die Vermittlung der Inhalte in die Selbstlernphase außerhalb der Lehrveranstaltungen verlagert, während die Präsenzphasen für die begleitete Anwendung und Übung genutzt werden. Unter Projektarbeit können unterschiedliche Ansätze von stärker praxisorientierten, offenen und umfangreicheren Aufgaben, bei denen ein Produkt entwickelt wird, verstanden werden. Gamification kann dabei motivierend wirken, wenn Punktesysteme oder thematisch ansprechende Gestaltung eingesetzt und Konkurrenz oder Kooperation gefördert werden.

In Kapitel 3 wird die Methodik der Arbeit erläutert. Aufgrund der Breite der involvierten Aspekte wurden diverse Methoden eingesetzt, sodass sowohl quantitative als auch qualitative Ergebnisse entstanden sind. Das Kapitel beschreibt auch das Modul *Einführung in Algorithmen und Datenstrukturen* als Kontext der Untersuchungen und gibt eine Übersicht über die Veröffentlichungen, die zur kumulativen Dissertation gehören.

Eine nicht veröffentlichte vollständige Studie ist in Kapitel 4 dargestellt. Dabei wurden ausgewählte Studierende zu Beginn des Moduls interviewt, um qualitativ-explorativ ihre Einstellungen und Erwartungen an das Modul einordnen zu können. Dabei zeigte sich, dass bereits in den ersten Wochen und im Vorkurs viele Informationen aufgenommen werden, die dabei helfen können, dass die Studierenden sich auf das Lernen einstellen können. Obwohl nur wenige sachlich falsche Aussagen geäußert wurden, weisen die Vorstellungen von Informatik insbesondere von Studierenden wenig informatikbezogener Fächer darauf hin, dass diese Informatik häufig auf den Umgang mit Computern reduzieren.

Die Kapitel 5, 6 und 7 beschäftigen sich stärker mit der praktischen Gestaltung von Einführungsmodulen in der Informatik im Kontext der Heterogenität der Studierenden und der Möglichkeiten, die die Digitalisierung eröffnet. Dabei beschleunigte die Corona-Pandemie und die dadurch zunächst erzwungene digitale Lehre innerhalb eines Projekts zur Umgestaltung der Lehre mit Flipped Classroom-Elementen die Entwicklung, sodass im zweiten Jahr der Umgestaltung bereits ein vollständiger Flipped Classroom entstanden ist. Kapitel 5 beschreibt dabei die grundlegende Entwicklung und zeigt die Veränderungen bis zum vierten, zum Zeitpunkt des Verfassens dieser Arbeit, laufenden

Jahr. Als zentrale Elemente sind dabei die interaktiven Lernmaterialien namens *Lessons* auf der Plattform JupyterNotebooks, die aktive Übung namens Coding Class sowie zugehörige Aufgaben für Coding Class und Abgabe entstanden. Die Aufgaben ermöglichen dabei durch dynamisch wachsende Bearbeitungszeiträume sowohl die Kontrolle der regelmäßigen Mitarbeit zu Beginn des Kurses als auch den Einsatz von Kreativität in einem umfangreicheren Projekt. Ein kooperatives Gamificationkonzept, bei dem die Studierenden gemeinsam durch Aktivitäten und kooperatives Verhalten Belohnungen unterhaltsamer und organisatorischer Art freischalten, ergänzt das Lehrkonzept.

Kapitel 6 betrachtet insbesondere das dritte Jahr. Dieses verlief zwar weitgehend mit einem ähnlichen Konzept wie das zweite Jahr, konnte in der Publikation zum Modulkonzept und dessen Evaluation jedoch noch nicht berücksichtigt werden. Neben einer Bestätigung der bisher bereits überwiegend positiven Ergebnisse konnten durch die Anpassungen im Zuge der Lockerungen von pandemiebedingten Einschränkungen weitere Erkenntnisse gewonnen werden. Insbesondere der Vergleich zwischen Coding Classes in Präsenz und in digitaler Form zeigt wie wichtig der persönliche Kontakt für das erfolgreiche Erlernen des Programmierens ist. In den Testaten, die zur Besprechung der Abgaben dienen, fallen dagegen nur geringe Unterschiede zwischen dem Präsenzformat und der digitalen Version auf. Insgesamt äußern sich viele Studierende positiv über die Breite des Angebots, sodass im Rahmen der Möglichkeiten diverse Veranstaltungsformate angeboten werden sollten. Darüber hinaus zeigt sich der hohe Stellenwert klarer Informationen für die Studierenden über die Veranstaltungsbestandteile sowie der erwarteten Lernweise entsprechend bisheriger Literaturempfehlungen.

Kapitel 7 beschäftigt sich mit den weiteren Publikationen zu der praktischen Gestaltung der Lehre und geht dabei zum einen auf die Gestaltung von Lernmaterialien ein, die im Rahmen des Flipped Classroom und der digitalen Möglichkeiten vor allem multimedial umgesetzt werden. Außerdem werden Überlegungen zur Verbindung des fachlichen Kontexts und digitaler Lehre erläutert sowie Strategien angeführt, um kompetenzorientiert Open-Books-Klausuren umzusetzen. Dabei werden sowohl zuvor beschriebene Empfehlungen aus der Literatur zu multimedialen Lehrinhalten aufgegriffen, aber auch die Erkenntnisse aus der Fallstudie einbezogen. Im Kontext des bisherigen Forschungsstands weisen diese darauf hin, dass authentisch und enthusiastisch gestaltete Lernmaterialien nicht zwingend Perfektion und professioneller Aufnahmetechnik bedürfen, um beim erfolgreichen Lernen zu unterstützen.

Im letzten Teil der Arbeit werden die Ergebnisse zunächst insgesamt im Hinblick auf die Limitierungen und ihre Auswirkungen diskutiert (siehe Kapitel 8). Obwohl die Untersuchungen lediglich im Rahmen eines Moduls durchgeführt wurden, können einige Erkenntnisse auch unter Betrachtung der Studienlage zu den inkludierten Aspekten zur weiter erfolgreichen Gestaltung der Lehre in ähnlichen Kontexten beitragen. Besonders die mehrmalige Durchführung und Anpassung des Konzepts sorgte dafür, dass auch Details betrachtet werden konnten, sodass ein Gesamtkonzept für die gegebene Lage entstanden ist. Auch wenn dieses aufgrund von Änderungen in der Studienstruktur nicht genauso weiter geführt werden kann, können Bestandteile davon in Folgemodulen übernommen werden. Kapitel 9 beschreibt dabei welche Überlegungen zur Anpassung notwendig sind. Eine ebenfalls in dem Kapitel vorhandene Generalisierung erläutert die

Übertragbarkeit der Ergebnisse auch auf andere fachliche Kontexte und fortgeschrittene Kurse des Fachs, womit das Kapitel insgesamt einen Ausblick darstellt.

Die Arbeit zeigt, dass bei der Gestaltung der Lehre im Fach Informatik diverse Voraussetzungen zu beachten sind, das Fach dabei allerdings einen guten Rahmen bietet, um neue Lehrkonzepte auszuprobieren. Ein konkretes Konzept für ein Einführungsmodul mit dem Fokus auf der aktiven Anwendung der Inhalte und einer motivierenden und kooperativen Lehrform wurde im Rahmen dieser Arbeit vorgestellt. Obwohl die hochschuldidaktische Forschung in der Informatikdidaktik bisher einen eher geringen Stellenwert einnimmt, liegen bereits einige Fallstudien und erste Metastudien vor, zu denen die Ergebnisse dieser Arbeit ebenfalls beitragen sollen. Dadurch sollen mehr Lehrende ermutigt werden zumindest einzelne Bestandteile der erläuterten Konzepte in der Lehre auszuprobieren und die Programmierausbildung Studierender dabei nachhaltig zu verbessern.

Literaturverzeichnis und Anhang

Literaturverzeichnis

- [AB18] AUSTERSCHMIDT, Kim; BEBERMEIER, Sarah: Studienanfänger/innen in Fächern mit mathematischen Lehrinhalten. Mathematikbezogene Einstellungen und Kompetenzen, Nutzung & Nutzen von Vorkursen. In: STEIN, Martin (Hrsg.): *Beiträge zum Mathematikunterricht 2018. Vorträge zur Mathematikdidaktik und zur Schnittstelle Mathematik/Mathematikdidaktik auf der gemeinsamen Jahrestagung GDM und DMV 2018*, WTM, 2018, S. 169–172
- [AD14] ABEYSEKERA, Lakmal; DAWSON, Phillip: Motivation and cognitive load in the flipped classroom: definition, rationale and a call for research. In: *Higher Education Research & Development* 34 (2014), S. 1–14. <http://dx.doi.org/10.1080/07294360.2014.934336>
- [AIA15] AZMI, Shahdatunnaim; IAHAD, Noorminshah; AHMAD, Norasnita: Gamification in online collaborative learning for programming courses: A literature review. In: *ARPJ Journal of Engineering and Applied Sciences* 10 (2015), Nr. 23, S. 18087–18094. – ISSN 1819–6608
- [AKFY19] ALTINPULLUK, Hakan; KILINC, Hakan; FIRAT, Mehmet; YUMURTACI, Onur: The influence of segmented and complete educational videos on the cognitive load, satisfaction, engagement, and academic achievement levels of learners. In: *Journal of Computers in Education* 7 (2019). <http://dx.doi.org/10.1007/s40692-019-00151-7>
- [AKK⁺10] AVOURIS, Nikolaos; KAXIRAS, Stefanos; KOUFOPAVLOU, Odysseas; SGARBAS, Kyriakos; STATHOPOULOU, Polyxeni: Teaching Introduction to Computing Through a Project-Based Collaborative Learning Approach. In: *2010 14th Panhellenic Conference on Informatics*, 2010. <http://dx.doi.org/10.1109/PCI.2010.13>. S. 237–241
- [BDM01] BRUCE, Kim B.; DANYLUK, Andrea; MURTAGH, Thomas: A Library to Support a Graphics-Based Object-First Approach to CS 1. In: *SIGCSE Bull.* 33 (2001), Nr. 1. <http://dx.doi.org/10.1145/366413.364527>. – ISSN 0097–8418
- [BEF⁺73] BLOOM, Benjamin S.; ENGELHART, Max D.; FÜRST, Edward J.; HILL, Walker H.; KRATHWOHL, David R.; FÜNER, Eugen (Hrsg.); HÖRN, Ralf (Hrsg.): *Taxonomie von Lernzielen im kognitiven Bereich*. Weinheim und Basel: Beltz, 1973

- [BGJG13] BARATA, Gabriel; GAMA, Sandra; JORGE, Joaquim; GONÇALVES, Daniel: Engaging Engineering Students with Gamification. In: *2013 5th International Conference on Games and Virtual Worlds for Serious Applications, VS-GAMES 2013* (2013), S. 24–31. <http://dx.doi.org/10.1109/VS-GAMES.2013.6624228>
- [BHF11] BIEHLER, Rolf; HOCHMUTH, Reinhard; FISCHER, Pascal R.; WASSONG, Thomas: Transition von Schule zu Hochschule in der Mathematik: Probleme und Lösungsansätze. Version:2011. http://www.mathematik.tu-dortmund.de/ieem/bzmu2011/_BzMU11_2_Einzelbeitraege/BzMU11_BIEHLER_Rolf_Transition.pdf. In: *Beiträge zum Mathematikunterricht 2011*. 2011
- [Big96] BIGGS, John: Enhancing Teaching Through Constructive Alignment. In: *Higher Education* 32 (1996), S. 347–364. <http://dx.doi.org/10.1007/BF00138871>
- [BKSZ93] BOULDING, William; KALRA, Ajay; STAELIN, Richard; ZEITHAML, Valarie A.: A Dynamic Process Model of Service Quality: From Expectations to Behavioral Intentions. In: *Journal of Marketing Research* 30 (1993), Nr. 1, 7-27. <http://www.jstor.org/stable/3172510>
- [BM15] BENYON, David; MIVAL, Oli: Blended Spaces for Collaboration. In: *Computer Supported Cooperative Work (CSCW)* 24 (2015). <http://dx.doi.org/10.1007/s10606-015-9223-8>
- [BMNR14] BLACKWELL, Alan; MCLEAN, Alex; NOBLE, James; ROHRHUBER, Julian: Collaboration and learning through live coding (Dagstuhl Seminar 13382). In: *Dagstuhl Reports* Bd. 3 Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2014
- [BMRW05] BROWN, Margaret; MACRAE, Sheila; RODD, Melissa; WILIAM, Dylan: *Full report of research activities and results: Students' experiences of undergraduate mathematics*. King's College London Department of Education and Professional Studies, 2005
- [BQ19] BECKER, Brett A.; QUILLE, Keith: 50 Years of CS1 at SIGCSE: A Review of the Evolution of Introductory Programming Education Research. In: *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. New York, NY, USA: Association for Computing Machinery, 2019 (SIGCSE '19). – ISBN 9781450358903. <http://dx.doi.org/10.1145/3287324.3287432>
- [Bri21] BRINKMEIER, Michael: *AlgoViz*. <https://gitdid.informatik.uni-osnabrueck.de/algoviz/algoviz>, 2021. – zuletzt aufgerufen am 20.01.2023
- [Bro19] BROOKS, F.P.: *Vom Mythos des Mann-Monats: Essays über Software-Engineering*. mitp, 2019 (Mitp Business). – ISBN 9783958458772
- [BS03] BEICHNER, Robert; SAUL, Jeff: Introduction to the SCALE-UP (Student-Centered Activities for Large Enrollment Undergraduate Programs) Project. (2003)

- [BS08] BRATHWAITE, Brenda; SCHREIBER, Ian: *Challenges for Game Designers*. 1. USA: Charles River Media, Inc., 2008. – ISBN 158450580X
- [BS12] BERGMANN, J.; SAMS, A.: *Flip Your Classroom: Reach Every Student in Every Class Every Day*. International Society for Technology in Education, 2012 (Flipped Learning Series). – ISBN 9781564843159
- [BSB18] BATEMAN, John A.; SCHMIDT-BORCHERDING, Florian: The Communicative Effectiveness of Education Videos: Towards an Empirically-Motivated Multimodal Account. In: *Multimodal Technologies and Interaction 2* (2018), Nr. 3. <http://dx.doi.org/10.3390/mti2030059>. – ISSN 2414-4088
- [BT14] BOSSE, Elke; TRAUTWEIN, Caroline: Individuelle und institutionelle Herausforderungen der Studieneingangsphase. In: *Zeitschrift für Hochschulentwicklung 9* (2014). <http://dx.doi.org/10.3217/zfhe-9-05/03>
- [BV13] BISHOP, J.L.; VERLEGER, Matthew: The flipped classroom: A survey of the research. In: *ASEE Annual Conference and Exposition, Conference Proceedings* (2013)
- [BVL21] BAKSHI, Smita; VAHID, Frank; LYSECKY, Roman: *zyBooks - About Us*. <https://www.zybooks.com/about-us/>, 2021. – zuletzt aufgerufen am 03.01.2023
- [CHC16] CAMPBELL, Jennifer; HORTON, Diane; CRAIG, Michelle: Factors for Success in Online CS1. In: *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*. New York, NY, USA: Association for Computing Machinery, 2016 (ITiCSE '16). – ISBN 9781450342315. <http://dx.doi.org/10.1145/2899415.2899457>
- [CHCG14] CAMPBELL, Jennifer; HORTON, Diane; CRAIG, Michelle; GRIES, Paul: Evaluating an Inverted CS1. In: *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*. New York, NY, USA: Association for Computing Machinery, 2014 (SIGCSE '14). – ISBN 9781450326056. <http://dx.doi.org/10.1145/2538862.2538943>
- [CJ18] COLLIOT, Tiphaine; JAMET, Eric: Understanding the effects of a teacher video on learning from a multimedia document: an eye-tracking study. In: *Educational Technology Research and Development 66* (2018), S. 1415–1433
- [CK77] CLANCY, Michael J.; KNUTH, Donald E.: *A Programming and Problem-Solving Seminar*. Stanford, CA, USA: Stanford University, 1977. – Forschungsbericht
- [Cla83] CLARK, Richard E.: Reconsidering Research on Learning from Media. In: *Review of Educational Research 53* (1983), Nr. 4, 445–459. <http://www.jstor.org/stable/1170217>. – ISSN 00346543, 19351046
- [Cop07] COPLEY, Jonathan: Audio and video podcasts of lectures for campus-based students: production and evaluation of student use. In: *Innovations in Education and Teaching International 44* (2007), Nr. 4, 387–399. <http://dx.doi.org/10.1080/14703290701602805>

- [Dan04] DANAN, Martine: Captioning and Subtitling: Undervalued Language Learning Strategies. In: *Meta* 49 (2004), S. 67–77. <http://dx.doi.org/10.7202/009021ar>
- [Dav93] DAVIES, Simon P.: Models and theories of programming strategy. In: *International Journal of Man-Machine Studies* 39 (1993), Nr. 2, 237–267. <http://dx.doi.org/10.1006/imms.1993.1061>. – ISSN 0020–7373
- [DB16] DÖRING, Nicola; BORTZ, Jürgen: *Forschungsmethoden und Evaluation in den Sozial- und Humanwissenschaften*. 5. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016. http://dx.doi.org/10.1007/978-3-642-41089-5_10. – ISBN 978–3–642–41089–5
- [DDAA15] DICHEVA, Darina; DICHEV, Christo; AGRE, Gennady; ANGELOVA, Galia: Gamification in education: A systematic mapping study. In: *Journal of educational technology & society* 18 (2015), Nr. 3, S. 75–88
- [DDKN11] DETERDING, Sebastian; DIXON, Dan; KHALED, Rilla; NACKE, Lennart: From Game Design Elements to Gamefulness: Defining 'Gamification'. In: *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*. New York, NY, USA: Association for Computing Machinery, 2011 (MindTrek '11). <http://dx.doi.org/10.1145/2181037.2181040>. S. 9–15
- [DID19] DICHEVA, Darina; IRWIN, Keith; DICHEV, Christo: OneUp: Engaging Students in a Gamified Data Structures Course. In: *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. New York, NY, USA: Association for Computing Machinery, 2019 (SIGCSE '19). <http://dx.doi.org/10.1145/3287324.3287480>. S. 386–392
- [DKS35] DEWEY, J.; KILPATRICK, W.H.; SCHULZ, G.: *Der Projekt-Plan: Grundlegung und Praxis*. Böhlau, 1935 (Pädagogik des Auslands)
- [DSFD16] DAZO, Suzanne L.; STEPANEK, Nicholas R.; FULKERSON, Robert; DORN, Brian: An Empirical Analysis of Video Viewing Behaviors in Flipped CS1 Courses. In: *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*. New York, NY, USA: Association for Computing Machinery, 2016 (ITiCSE '16). – ISBN 9781450342315. <http://dx.doi.org/10.1145/2899415.2899468>
- [DTL+11] DETTMERS, Swantje; TRAUTWEIN, Ulrich; LÜDTKE, Oliver; GOETZ, Thomas; FRENZEL, Anne C.; PEKRUN, Reinhard: Students' emotions during homework in mathematics: Testing a theoretical model of antecedents and achievement outcomes. In: *Contemporary Educational Psychology* 36 (2011), Nr. 1, 25–35. <http://dx.doi.org/10.1016/j.cedpsych.2010.10.001>. – ISSN 0361–476X
- [dTRP09] DE KONING, Björn; TABBERS, Huib; RIKERS, Remy; PAAS, Fred: Towards a Framework for Attention Cueing in Instructional Animations: Guidelines for Research and Design. In: *Educational Psychology Review* 21 (2009), S. 113–140. <http://dx.doi.org/10.1007/s10648-009-9098-7>

- [EA12] ELNAGAR, Ashraf; ALI, Mahir: A Modified Team-Based Learning Methodology for Effective Delivery of an Introductory Programming Course. In: *Proceedings of the 13th Annual Conference on Information Technology Education*. New York, NY, USA: Association for Computing Machinery, 2012 (SIGITE '12). – ISBN 9781450314640. <http://dx.doi.org/10.1145/2380552.2380604>
- [EMV14] ENGELBERT, Benedikt; MORISSE, Karsten; VORNBERGER, Oliver: Zwischen Nutzung und Nutzen – Die Suche nach geeigneten Lernmaterialien und deren Mehrwerte im Kontext einer Informatikveranstaltung. In: *GMW 2014 - Lernräume gestalten - Bildungskontexte vielfältig denken*, 2014. <http://dx.doi.org/10.13140/RG.2.1.3218.4162>
- [ES17] ELMALEH, Joelle; SHANKARARAMAN, Venky: Improving student learning in an introductory programming course using flipped classroom and competency framework. In: *2017 IEEE Global Engineering Education Conference (EDUCON)*, 2017. <http://dx.doi.org/10.1109/EDUCON.2017.7942823>. S. 49–55
- [EVL⁺15] EDGCOMB, A.D.; VAHID, Frank; LYSECKY, R.; KNOESEN, A.; AMIRTHARAJAH, R.; DORF, M.L.: Student performance improvement using interactive textbooks: A three-university cross-semester analysis. In: *Proc. of ASEE Annual Conference*, 2015
- [Fac20] FACHBEREICH MATHEMATIK/INFORMATIK: *Modulbeschreibungen für die Lehrinheit 'Informatik'*. https://www.uni-osnabrueck.de/fileadmin/documents/public/ordnungen/Modulbeschreibungen_Informatik_2020-07.pdf, 2020. – zuletzt abgerufen am 14.12.2022
- [FAJ07] FREEMAN, Tierra M.; ANDERMAN, Lynley H.; JENSEN, Jane M.: Sense of Belonging in College Freshmen at the Classroom and Campus Levels. In: *The Journal of Experimental Education* 75 (2007), Nr. 3, S. 203–220. – ISSN 00220973, 19400683
- [Fea92] FEATHER, Norman T.: Values, Valences, Expectations, and Actions. In: *Journal of Social Issues* 48 (1992), Nr. 2, S. 109–124. <http://dx.doi.org/10.1111/j.1540-4560.1992.tb00887.x>
- [Fes62] FESTINGER, L.: *A Theory of Cognitive Dissonance*. Stanford University Press, 1962 (Mass communication series). – ISBN 9780804709118
- [FGN⁺11] FLATLA, David R.; GUTWIN, Carl; NACKE, Lennart E.; BATEMAN, Scott; MANDRYK, Regan L.: Calibration Games: Making Calibration Tasks Enjoyable by Adding Motivating Game Elements. In: *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA: Association for Computing Machinery, 2011 (UIST '11). – ISBN 9781450307161. <http://dx.doi.org/10.1145/2047196.2047248>
- [FMLR16] FOTARIS, Panagiotis; MASTORAS, Theodoros; LEINFELLNER, Richard; ROSUNALLY, Yasmine: Climbing up the leaderboard: An empirical study

- of applying gamification techniques to a computer programming class. In: *Electronic Journal of e-learning* 14 (2016), Nr. 2, S. 94–110
- [FPD18] FERRATT, Thomas; PRASAD, Jayesh; DUNNE, E.: Fast and Slow Processes Underlying Theories of Information Technology Use. In: *Journal of the Association for Information Systems* 19 (2018), S. 1–22. <http://dx.doi.org/10.17705/1jais.00477>
- [FR16] FALLMANN, Irmgard; REINTHALER, Pia: Bedeutung und Förderung von selbstreguliertem Lernen im Inverted Classroom. In: *Das Inverted Classroom Modell Begleitband zur 5. Konferenz 'Inverted Classroom and Beyond' 2016 an der FH St. Pölten am 23. und 24. Februar 2016*, Johann Haag, Christian F. Freisleben-Teutscher, 2016
- [Fri21] FRICKE, Johanna: *Lehre der Zukunft gestalten: Projekt UOS.DLL gestartet*. <https://digitale-lehre.virtuos.uni-osnabrueck.de/2021/09/20/1ehre-der-zukunft-gestalten-projekt-uos-dll-gestartet/>, 2021. – zuletzt aufgerufen am 09.01.2023
- [GB19] GROENING, Christopher; BINNEWIES, Carmen: 'Achievement unlocked!' - The impact of digital achievements as a gamification element on motivation and performance. In: *Computers in Human Behavior* 97 (2019), S. 151–166. <http://dx.doi.org/10.1016/j.chb.2019.02.026>. – ISSN 0747–5632
- [GBHW12] GROSS, Lena; BOGER, Mai-Anh; HAMANN, Silke; WEDJELEK, Marc: *ZEITLast Lehrzeit und Lernzeit: Studierbarkeit der BA-/BSc- und MA/MSc- Studiengänge als Adaption von Lehrorganisation und Zeitmanagement unter Berücksichtigung von Fächerkultur und neuen Technologien*. https://www.blogs.uni-mainz.de/medienpaedagogik/files/2014/03/Abschlussbericht_ZEITLast.pdf, 2012. – zuletzt abgerufen am 03.01.2023
- [GKC14] GIANNAKOS, Michail N.; KROGSTIE, John; CHRISOCHOIDES, Nikos: Reviewing the Flipped Classroom Research: Reflections for Computer Science Education. In: *Proceedings of the Computer Science Education Research Conference*. New York, NY, USA: Association for Computing Machinery, 2014 (CSERC '14). – ISBN 9781450333474. <http://dx.doi.org/10.1145/2691352.2691354>
- [GKR14] GUO, Philip J.; KIM, Juho; RUBIN, Rob: How Video Production Affects Student Engagement: An Empirical Study of MOOC Videos. In: *Proceedings of the First ACM Conference on Learning @ Scale Conference*. New York, NY, USA: Association for Computing Machinery, 2014 (L@S '14). – ISBN 9781450326698. <http://dx.doi.org/10.1145/2556325.2566239>
- [GMT09] GRIFFIN, Darren K.; MITCHELL, David; THOMPSON, Simon J.: Podcasting by synchronising PowerPoint and voice: What are the pedagogical benefits? In: *Computers & Education* 53 (2009), Nr. 2, 532–539. <http://dx.doi.org/10.1016/j.compedu.2009.03.011>. – ISSN 0360–1315

- [Göl19] GÖLLER, R.: *Selbstreguliertes Lernen im Mathematikstudium*. Springer Fachmedien Wiesbaden, 2019 (Studien zur Hochschuldidaktik und zum Lehren und Lernen mit digitalen Medien in der Mathematik und in der Statistik). – ISBN 9783658286811
- [GT05] GHINEA, G.; THOMAS, J.P.: Quality of perception: user quality of service in multimedia presentations. In: *IEEE Transactions on Multimedia* 7 (2005), Nr. 4, S. 786–789. <http://dx.doi.org/10.1109/TMM.2005.850960>
- [Gud14] GUDJONS, H.: *Handlungsorientiert lehren und lernen: Schüleraktivierung - Selbsttätigkeit - Projektarbeit*. Klinkhardt, 2014 (Erziehen und Unterrichten in der Schule). – ISBN 9783781520011
- [HAB11] HAKIMZADEH, Hossein; ADAIKKALAVAN, Raman; BATZINGER, Robert: Successful Implementation of an Active Learning Laboratory in Computer Science. In: *Proceedings of the 39th Annual ACM SIGUCCS Conference on User Services*. New York, NY, USA: Association for Computing Machinery, 2011 (SIGUCCS '11). – ISBN 9781450310239
- [Hak98] HAKE, Richard: Interactive-Engagement Versus Traditional Methods: A Six-Thousand-Student Survey of Mechanics Test Data for Introductory Physics Courses. In: *American Journal of Physics - AMER J PHYS* 66 (1998). <http://dx.doi.org/10.1119/1.18809>
- [Har05] HARRINGTON, Neil: The Frustration Discomfort Scale: development and psychometric properties. In: *Clinical Psychology & Psychotherapy* 12 (2005), S. 374 – 387. <http://dx.doi.org/10.1002/cpp.465>
- [HCC+14] HORTON, Diane; CRAIG, Michelle; CAMPBELL, Jennifer; GRIES, Paul; ZINGARO, Daniel: Comparing Outcomes in Inverted and Traditional CS1. In: *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education*. New York, NY, USA: Association for Computing Machinery, 2014 (ITiCSE '14). – ISBN 9781450328333. <http://dx.doi.org/10.1145/2591708.2591752>
- [HHS+10] HEUBLEIN, Ulrich; HUTZSCH, C; SCHREIBER, J; SOMMER, D; BESUCH, G: *Ursachen des Studienabbruchs in Bachelor- und in herkömmlichen Studiengängen: Ergebnisse einer bundesweiten Befragung von Exmatrikulierten des Studienjahres 2007/08*. Hochschul-Informationssystem GmbH, 2010 (HIS: Forum Hochschule). ISSN 1863–5563
- [HK13] HAMARI, Juho; KOIVISTO, Jonna: Social motivations to use gamification: An empirical study of gamifying exercise. In: *ECIS 2013 - Proceedings of the 21st European Conference on Information Systems* (2013). https://aisel.aisnet.org/ecis2013_cr/105
- [HKS14] HAMARI, Juho; KOIVISTO, Jonna; SARSA, Harri: Does Gamification Work? — A Literature Review of Empirical Studies on Gamification. In: *Proceedings of the Annual Hawaii International Conference on System Sciences* (2014). <http://dx.doi.org/10.1109/HICSS.2014.377>

- [HML⁺20] HODGES, Charles; MOORE, Stephanie; LOCKEE, Barbara; TRUST, Torrey; BOND, Mark: The Difference Between Emergency Remote Teaching and Online Learning. (2020). <https://er.educause.edu/articles/2020/3/the-difference-between-emergency-remote-teaching-and-online-learning>
- [Hod19] HODGES, Mark: Flipping One Day Each Week in a Smaller CS1 Course: An Experience Report. In: *J. Comput. Sci. Coll.* 34 (2019), Nr. 7. – ISSN 1937–4771
- [HP16] HSU, Wen C.; PLUNKETT, Scott W.: Attendance and Grades in Learning Programming Classes. In: *Proceedings of the Australasian Computer Science Week Multiconference*. New York, NY, USA: Association for Computing Machinery, 2016 (ACSW '16). – ISBN 9781450340427. <http://dx.doi.org/10.1145/2843043.2843061>
- [HSA13] HASENBERG, Svea; SCHMIDT-ATZERT, Lothar: Die Rolle von Erwartungen zu Studienbeginn: Wie bedeutsam sind realistische Erwartungen über Studieninhalte und Studienaufbau für die Studienzufriedenheit? In: *Zeitschrift für pädagogische Psychologie* 27 (2013), Nr. 1-2, 87-93. <https://doi.org/10.1024/1010-0652/a000091>
- [HSS15] HURTIENNE, Dominik; SCHROEDER, Ulrik; SPANNAGEL, Christian: IT EnGAGES!–Adaptierbare Gamification in einer Anfänger-Programmiervorlesung. In: *HDI 2014 : Gestalten von Übergängen 2015* (2015), Nr. 9, S. 27 – 43
- [Jen02] JENKINS, Tony: On the difficulty of learning to program. In: *3rd Annual Conference of the LTSN Centre for Information and Computer Sciences*, 2002
- [JJS86] JOHNSON, Roger T.; JOHNSON, David W.; STANNE, Mary B.: Comparison of Computer-Assisted Cooperative, Competitive, and Individualistic Learning. In: *American Educational Research Journal* 23 (1986), Nr. 3, 382–392. <https://doi.org/10.2307/1163055>. – ISSN 00028312, 19351011
- [Jup18] JUPYTER, Project: *Jupyter receives the ACM Software System Award*. <https://blog.jupyter.org/jupyter-receives-the-acm-software-system-award-d433b0dfe3a2>. Version: 2018. – zuletzt aufgerufen am 13.12.2022
- [KGM18] KAPOOR, Amanpreet; GARDNER-MCCUNE, Christina: Understanding Professional Identities and Goals of Computer Science Undergraduate Students. In: *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*. New York, NY, USA: Association for Computing Machinery, 2018 (SIGCSE '18). <http://dx.doi.org/10.1145/3159450.3159474>. S. 191–196
- [KGS⁺14] KIM, Juho; GUO, Philip J.; SEATON, Daniel T.; MITROS, Piotr; GAJOS, Krzysztof Z.; MILLER, Robert C.: Understanding In-Video Dropouts and Interaction Peaks Inonline Lecture Videos. In: *Proceedings of the First ACM*

- Conference on Learning @ Scale Conference*. New York, NY, USA: Association for Computing Machinery, 2014 (L@S '14). – ISBN 9781450326698. <http://dx.doi.org/10.1145/2556325.2566237>
- [KHBP14] KLOSTERMANN, Mareike; HÖFFLER, Tim; BERNHOLT, Andrea; PARCHMANN, Ilka: Erfassung und Charakterisierung kognitiver und affektiver Merkmale von Studienanfängern im Fach Chemie. In: *Zeitschrift für Didaktik der Naturwissenschaften* (2014), Nr. 20, 101–113. <https://doi.org/10.1007/s40573-014-0011-7>
- [KHGP20] KOHLS, Christian; HIMPSL-GUTERMANN, Klaus; POHL, Hans-Martin: Digitalisierung und die Gestaltung von Lehr-/Lernräumen. In: DEIMANN, Markus (Hrsg.); TREECK, Timo van (Hrsg.): *Digitalisierung der Hochschullehre Aspekte und Perspektiven der Transformation*, DUZ Open, 2020
- [KINP14] KNUTAS, Antti; IKONEN, Jouni; NIKULA, Uolevi; PORRAS, Jari: Increasing Collaborative Communications in a Programming Course with Gamification: A Case Study. In: *Proceedings of the 15th International Conference on Computer Systems and Technologies*. New York, NY, USA: Association for Computing Machinery, 2014 (CompSysTech '14). – ISBN 9781450327534. <http://dx.doi.org/10.1145/2659532.2659620>
- [KJCJ17] KARABULUT, Aliye; JARAMILLO CHERREZ, Nadia; JAHREN, Charles: A systematic review of research on the flipped learning method in engineering education: Flipped Learning in Engineering Education. In: *British Journal of Educational Technology* 49 (2017). <http://dx.doi.org/10.1111/bjet.12548>
- [KKJ+15] KOEDINGER, Kenneth R.; KIM, Jihee; JIA, Julianna Z.; MCLAUGHLIN, Elizabeth A.; BIER, Norman L.: Learning is Not a Spectator Sport: Doing is Better than Watching for Learning from a MOOC. In: *Proceedings of the Second (2015) ACM Conference on Learning @ Scale*. New York, NY, USA: Association for Computing Machinery, 2015 (L@S '15). – ISBN 9781450334112. <http://dx.doi.org/10.1145/2724660.2724681>
- [KL21] KRAPF, Regula; LIEBENDÖRFER, Michael: Was bewirkt die Pflichtabgabe von Übungsaufgaben in der Hochschulmathematik? Ein empirischer Vergleich. In: *mathematica didactica* 44 (2021), Nr. 2. <http://dx.doi.org/10.18716/ojs/md/2021.1215>
- [KM15] KEEN, Aaron; MAMMEN, Kurt: Program Decomposition and Complexity in CS1. In: *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*. New York, NY, USA: Association for Computing Machinery, 2015 (SIGCSE '15). – ISBN 9781450329668. <http://dx.doi.org/10.1145/2676723.2677219>
- [KNBH16] KÖPPE, Christian; NIELS, Ralph; BAKKER, René; HOPPENBROUWERS, Stijn: Flipped Classroom Patterns: Controlling the Pace. In: *Proceedings of the 10th Travelling Conference on Pattern Languages of Programs*. New York, NY, USA: Association for Computing Machinery, 2016 (VikingPLoP '16). – ISBN 9781450342001. <http://dx.doi.org/10.1145/3022636.3022637>

- [KNH⁺15a] KÖPPE, Christian; NIELS, Ralph; HOLWERDA, Robert; TIJMSMA, Lars; VAN DIEPEN, Niek; VAN TURNHOUT, Koen; BAKKER, René: Flipped Classroom Patterns: Designing Valuable in-Class Meetings. In: *Proceedings of the 20th European Conference on Pattern Languages of Programs*. New York, NY, USA: Association for Computing Machinery, 2015 (EuroPLoP '15). – ISBN 9781450338479. <http://dx.doi.org/10.1145/2855321.2855348>
- [KNH⁺15b] KÖPPE, Christian; NIELS, Ralph; HOLWERDA, Robert; TIJMSMA, Lars; VAN DIEPEN, Niek; VAN TURNHOUT, Koen; BAKKER, René: Flipped Classroom Patterns: Using Student Solutions. In: *Proceedings of the 22nd Conference on Pattern Languages of Programs*. USA: The Hillside Group, 2015 (PLoP '15). – ISBN 9781941652039
- [Kno97] KNOLL, Michael: The project method: Its vocational education origin and international development. In: *Journal of Industrial Teacher Education* 34 (1997), S. 59–80
- [Kol96] KOLMOS, Anette: Reflections on Project Work and Problem-based Learning. In: *European Journal of Engineering Education* 21 (1996), Nr. 2, S. 141–148. <http://dx.doi.org/10.1080/03043799608923397>
- [Koz94] KOZMA, Robert B.: Will Media Influence Learning? Reframing the Debate. In: *Educational Technology Research and Development* 42 (1994), Nr. 2. <http://www.jstor.org/stable/30218683>. – ISSN 10421629, 15566501
- [KP14] KÖPPE, Christian; PORTIER, Michel: Lecture Design Patterns: Improving the Beginning of a Lecture. In: *Proceedings of the 19th European Conference on Pattern Languages of Programs*. New York, NY, USA: Association for Computing Machinery, 2014 (EuroPLoP '14). – ISBN 9781450334167. <http://dx.doi.org/10.1145/2721956.2721957>
- [KSWB13] KERRES, M.; SCHMIDT, A.; WOLFF-BENDIK, K.: Didaktische Konzeption und Instruktionsdesign – der Vielfalt gerecht werden. In: KERRES, M. (Hrsg.); HANFT, A. (Hrsg.); WILKESMANN, U. (Hrsg.); WOLFF-BENDIK, K. (Hrsg.): *Studium 2020: Positionen und Perspektiven zum lebenslangen Lernen an Hochschulen*. Waxmann Verlag, 2013. – ISBN 9783830925798, Kapitel 2.4
- [Kus08] KUSSMAUL, Clifton L.: Scaffolding for Multiple Assignment Projects in CS1 and CS2. In: *Companion to the 23rd ACM SIGPLAN Conference on Object-Oriented Programming Systems Languages and Applications*. New York, NY, USA: Association for Computing Machinery, 2008 (OOPSLA Companion '08). – ISBN 9781605582207. <http://dx.doi.org/10.1145/1449814.1449890>
- [LA10] LUCERO, Andrés; ARRASVUORI, Juha: PLEX Cards: A Source of Inspiration When Designing for Playfulness. In: *Proceedings of the 3rd International Conference on Fun and Games*. New York, NY, USA: Association for Computing Machinery, 2010 (Fun and Games '10). – ISBN 9781605589077. <http://dx.doi.org/10.1145/1823818.1823821>

- [LADU19] LEE, Chris; ANG, Chin-Siang; DIPOLOG-UBANAN, Genevieve: 'My First Year in The University': Students' Expectations, Perceptions and Experiences. In: *JOURNAL OF SOCIAL SCIENCE RESEARCH* 14 (2019), S. 3134–3145. <http://dx.doi.org/10.24297/jssr.v14i0.8006>
- [Lar13] LARGENT, David L.: Flipping a Large CS0 Course: An Experience Report about Exploring the Use of Video, Clickers and Active Learning. In: *J. Comput. Sci. Coll.* 29 (2013), Nr. 1. – ISSN 1937–4771
- [LE13] LOCKWOOD, Kate; ESSELSTEIN, Rachel: The Inverted Classroom and the CS Curriculum. In: *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*. New York, NY, USA: Association for Computing Machinery, 2013 (SIGCSE '13). – ISBN 9781450318686. <http://dx.doi.org/10.1145/2445196.2445236>
- [LHC17] LO, Chung K.; HEW, Khe F.; CHEN, Gaowei: Toward a set of design principles for mathematics flipped classrooms: A synthesis of research in mathematics education. In: *Educational Research Review* 22 (2017), 50–73. <http://dx.doi.org/10.1016/j.edurev.2017.08.002>. – ISSN 1747–938X
- [Lim] LIMESURVEY GMBH: *LimeSurvey*. <https://www.limesurvey.org/de/>, . – zuletzt aufgerufen am 31.01.2023
- [LK77] LANDIS, J. R.; KOCH, Gary G.: The Measurement of Observer Agreement for Categorical Data. In: *Biometrics* 33 (1977), Nr. 1, 159–174. <http://www.jstor.org/stable/2529310>. – ISSN 0006341X, 15410420
- [LL15] LACHER, Lisa L.; LEWIS, Mark C.: The Effectiveness of Video Quizzes in a Flipped Class. In: *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*. New York, NY, USA: Association for Computing Machinery, 2015 (SIGCSE '15). – ISBN 9781450329668. <http://dx.doi.org/10.1145/2676723.2677302>
- [LLH17] LAPPALAINEN, Vesa; LAKANEN, A.-J; HÖGMANDER, Harri: Towards Computer-based Exams in CS1. In: *Proceedings of the 9th International Conference on Computer Supported Education - Volume 2: CSEDU*, 2017. <http://dx.doi.org/10.5220/0006323501250136>. S. 125–136
- [LLS15] LATULIPE, Celine; LONG, N. B.; SEMINARIO, Carlos E.: Structuring Flipped Classes with Lightweight Teams and Gamification. In: *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*. New York, NY, USA: Association for Computing Machinery, 2015 (SIGCSE '15). – ISBN 9781450329668. <http://dx.doi.org/10.1145/2676723.2677240>
- [LM99] LICATA, Jane W.; MAXHAM, James G.: Student Expectations of the University Experience: Levels and Antecedents for Pre-Entry Freshmen. In: *Journal of Marketing for Higher Education* 9 (1999), Nr. 1, S. 69–91. http://dx.doi.org/10.1300/J050v09n01_06

- [LMIS00] LEVIN, Daniel T.; MOMEN, Nausheen; IV, Sarah B. D.; SIMONS, Daniel J.: Change Blindness Blindness: The Metacognitive Error of Overestimating Change-detection Ability. In: *Visual Cognition* 7 (2000), Nr. 1-3, S. 397–412. <http://dx.doi.org/10.1080/135062800394865>
- [LP00] LAGE, Maureen; PLATT, Glenn: The Internet and the Inverted Classroom. In: *Journal of Economic Education* 31 (2000). <http://dx.doi.org/10.1080/00220480009596756>
- [LPT00] LAGE, Maureen; PLATT, Glenn; TREGLIA, Michael: Inverting the Classroom: A Gateway to Creating an Inclusive Learning Environment. In: *Journal of Economic Education* 31 (2000). <http://dx.doi.org/10.1080/0022048009596759>
- [LR12] LLOYD, Steven A.; ROBERTSON, Chuck L.: Screencast Tutorials Enhance Student Learning of Statistics. In: *Teaching of Psychology* 39 (2012), Nr. 1. <http://dx.doi.org/10.1177/0098628311430640>
- [LRSA⁺18] LUXTON-REILLY, Andrew; SIMON; ALBLUWI, Ibrahim; BECKER, Brett A.; GIANNAKOS, Michail; KUMAR, Amruth N.; OTT, Linda; PATERSON, James; SCOTT, Michael J.; SHEARD, Judy; SZABO, Claudia: Introductory Programming: A Systematic Literature Review. In: *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*. New York, NY, USA: Association for Computing Machinery, 2018 (ITiCSE 2018 Companion). – ISBN 9781450362238. <http://dx.doi.org/10.1145/3293881.3295779>
- [LS11] LASSERRE, Patricia; SZOSTAK, Carolyn: Effects of Team-Based Learning on a CS1 Course. In: *Proceedings of the 16th Annual Joint Conference on Innovation and Technology in Computer Science Education*. New York, NY, USA: Association for Computing Machinery, 2011 (ITiCSE '11). – ISBN 9781450306973. <http://dx.doi.org/10.1145/1999747.1999787>
- [LWC23] LIEBERMAN, Zach; WATSON, Theodore; CASTRO, Arturo: *OpenFrameworks*. <https://openframeworks.cc/>, 2023. – zuletzt aufgerufen am 20.01.2023
- [MA68] MOORE, Omar K.; ANDERSON, Alan R.: Some Principles for the Design of Clarifying Educational Environments. In: GOSLIN, David (Hrsg.): *Handbook of Socialization Theory and Research*. Chicago: Rand McNally, 1968
- [MA16] MATTHÍASDÓTTIR, Ásrún; ARNALDS, Hallgrímur: E-Assessment: Students' Point of View. In: *Proceedings of the 17th International Conference on Computer Systems and Technologies 2016*. New York, NY, USA: Association for Computing Machinery, 2016 (CompSysTech '16). – ISBN 9781450341820. <http://dx.doi.org/10.1145/2983468.2983497>. S. 369–374
- [Mal81] MALONE, Thomas W.: Toward a theory of intrinsically motivating instruction. In: *Cognitive Science* 5 (1981), Nr. 4, S. 333–369. [http://dx.doi.org/10.1016/S0364-0213\(81\)80017-1](http://dx.doi.org/10.1016/S0364-0213(81)80017-1). – ISSN 0364–0213

- [Mal10] MALAN, David J.: Reinventing CS50. In: *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*. New York, NY, USA: Association for Computing Machinery, 2010 (SIGCSE '10). – ISBN 9781450300063. <http://dx.doi.org/10.1145/1734263.1734316>. S. 152–156
- [May21] MAYER, Richard E.: Evidence-Based Principles for How to Design Effective Instructional Videos. In: *Journal of Applied Research in Memory and Cognition* 10 (2021), Nr. 2, S. 229–240. <http://dx.doi.org/10.1016/j.jarmac.2021.03.007>. – ISSN 2211–3681
- [MD06] MATZKO, Sarah; DAVIS, Timothy: Using Graphics Research to Teach Freshman Computer Science. In: *ACM SIGGRAPH 2006 Educators Program*. New York, NY, USA: Association for Computing Machinery, 2006 (SIGGRAPH '06). – ISBN 1595933646. <http://dx.doi.org/10.1145/1179295.1179305>
- [Mey10] MEYER AUF DER HEYDE, Achim: Macht der Bachelor krank? Studieren zwischen Druck, Kontrolle und Konkurrenz - Anforderungen an eine neue Studierendengeneration. Version:2010. <http://dx.doi.org/10.3278/6001591w>. In: *Endstation Bologna? Zehn Jahre europäischer Hochschulraum*. Bd. 116. 2010. – DOI 10.3278/6001591w. – ISBN 9783763943661, S. 69–73
- [Moh20] MOHAMED, Abdallah: Evaluating the Effectiveness of Flipped Teaching in a Mixed-Ability CS1 Course. In: *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education*. New York, NY, USA: Association for Computing Machinery, 2020 (ITiCSE '20). – ISBN 9781450368742. <http://dx.doi.org/10.1145/3341525.3387395>
- [Mor16] MORISSE, Karsten: Inverted Classroom in der Hochschullehre - Chancen, Hemmnisse und Erfolgsfaktoren. In: *Das Inverted Classroom Modell Begleitband zur 5. Konferenz 'Inverted Classroom and Beyond' 2016 an der FH St. Pölten am 23. und 24. Februar 2016*, Johann Haag, Christian F. Freisleben-Teutscher, 2016
- [Mor19] MORISSE, Karsten: Inverted Classroom in der Informatik: ein Ansatz zum Erwerb überfachlicher Kompetenzen. In: KAUFFELD, Simone (Hrsg.); OTHMER, Julius (Hrsg.): *Handbuch Innovative Lehre*. Wiesbaden: Springer Fachmedien Wiesbaden, 2019. – ISBN 978–3–658–22796–8, S. 99–113
- [MTOL13] MOCCOZET, Laurent; TARDY, Camille; OPPRECHT, Wanda; LÉONARD, Michel: Gamification-based assessment of group work. In: *2013 International Conference on Interactive Collaborative Learning (ICL)*, 2013. <http://dx.doi.org/10.1109/ICL.2013.6644565>. S. 171–179
- [Müh10] MÜHLMANN, T.: *Studien- und Berufserwartungen von Studienanfängern Sozialer Arbeit: Ergebnisse einer quantitativen Befragung von Studierenden Sozialer Arbeit zu ihren Merkmalen und Erfahrungen, Gründen der Studien- und Hochschulwahl sowie berufsbezogenen Interessen, Einstellungen und Zielen zu Beginn ihres Studiums*. Books on Demand, 2010 (Schriften und

- Werkstattspapier aus dem Institut für Soziale Arbeit und Sozialpolitik). – ISBN 9783839115909
- [NB19] NAINAN, Mariam; BALAKRISHNAN, Balamuralithara: Design and Evaluation of Worked Examples for Teaching and Learning Introductory Programming at Tertiary Level. In: *Malaysian Online Journal of Educational Technology* 7 (2019), S. 30–44. <http://dx.doi.org/10.17220/mojet.2019.04.003>
- [Neu18] NEUDEUTSCH 2.0: *Kollateralproduktivität*. <https://www.facebook.com/wortgeburten/photos/weil-wir-zum-einen-kein-wortspielportal-sind-und-zum-anderen-mal-wieder-ein-posi/1007069019456497/>, 2018
- [NGK11] NIKULA, Uolevi; GOTEL, Orlena; KASURINEN, Jussi: A Motivation Guided Holistic Rehabilitation of the First Programming Course. In: *ACM Trans. Comput. Educ.* 11 (2011), Nr. 4. <http://dx.doi.org/10.1145/2048931.2048935>
- [OB22] OSSOVSKI, E.; BRINKMEIER, M.: COMPARING VIDEO AND INTERACTIVE LEARNING MATERIAL STYLES FOR PROGRAMMING. In: *EDULEARN22 Proceedings*, IATED, 2022 (14th International Conference on Education and New Learning Technologies). – ISBN 978–84–09–42484–9. <http://dx.doi.org/10.21125/edulearn.2022.1273>. S. 5381–5390
- [OBB⁺22] OSSOVSKI, Elisaweta; BRINKMEIER, Michael; BECKER, Ann-Katrin; HEMBROCK, Laura; KALBREYER, Daniel; KLECKER, Sven: Heterogenitätsbewusste digitale Gestaltung eines Einführungsmoduls in der Informatik. In: *MedienPädagogik: Zeitschrift für Theorie und Praxis der Medienbildung* 48 (2022), Nr. Digitalisierung als Katalysator für Diversität an Hochschulen et vice versa, S. 173–193. <http://dx.doi.org/10.21240/mpaed/48/2022.06.14.X>
- [OBK21] OSSOVSKI, Elisaweta; BRINKMEIER, Michael; KALBREYER, Daniel: Warum Onlinelehre sowie Open-Books-Klausuren und das Fach Informatik zusammenpassen. In: KIENLE, Andrea (Hrsg.); HARRER, Andreas (Hrsg.); HAAKE, Joerg M. (Hrsg.); LINGNAU, Andreas (Hrsg.): *DELFI 2021*. Bonn: Gesellschaft für Informatik e.V., 2021. – ISBN 978–3–88579–710–4, S. 265–270
- [OHS99] OSER, Fritz; HASCHER, Tina; SPYCHIGER, Maria: Lernen aus Fehlern Zur Psychologie des 'negativen' Wissens. In: ALTHOF, Wolfgang (Hrsg.): *Fehlerwelten: Vom Fehlermachen und Lernen aus Fehlern. Beiträge und Nachträge zu einem interdisziplinären Symposium aus Anlaß des 60. Geburtstags von Fritz Oser*. Wiesbaden: VS Verlag für Sozialwissenschaften, 1999. – ISBN 978–3–8100–2343–8, S. 11–41
- [OKHB21] OSSOVSKI, Elisaweta; KALBREYER, Daniel; HEMBROCK, Laura; BRINKMEIER, Michael: Cooperative Gamification in a Computer Science Introductory Module. In: *Proceedings of the 10th Computer Science Education Research Conference*. New York, NY, USA: Association for Com-

- puting Machinery, 2021 (CSERC '21). – ISBN 9781450385763. <http://dx.doi.org/10.1145/3507923.3507953>. S. 70–78
- [OP15] O'FLAHERTY, Jacqueline; PHILLIPS, Craig: The use of flipped classrooms in higher education: A scoping review. In: *The Internet and Higher Education* 25 (2015), S. 85–95. <http://dx.doi.org/10.1016/j.iheduc.2015.02.002>. – ISSN 1096–7516
- [Oss23] OSSOVSKI, Elisaweta: *Elisaweta Ossovski Dissertation Dataverse*. <https://data.goettingen-research-online.de/dataverse/ossovskidissertation>, 2023
- [OYT13] OHNO, Asako; YAMASAKI, Takahiro; TOKIWA, Kin-Ichiroh: A discussion on introducing half-anonymity and gamification to improve students' motivation and engagement in classroom lectures. In: *2013 IEEE Region 10 Humanitarian Technology Conference*, 2013. <http://dx.doi.org/10.1109/R10-HTC.2013.6669044>. S. 215–220
- [Per19] PERSIKE, Malte: Videos in der Lehre: Wirkungen und Nebenwirkungen. In: NIEGEMANN, Helmut (Hrsg.); WEINBERGER, Armin (Hrsg.): *Lernen mit Bildungstechnologien: Praxisorientiertes Handbuch zum intelligenten Umgang mit digitalen Medien*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2019. – ISBN 978–3–662–54373–3, S. 1–31
- [PGBP15] PEDREIRA, Oscar; GARCÍA, Félix; BRISABOA, Nieves; PIATTINI, Mario: Gamification in software engineering – A systematic mapping. In: *Information and Software Technology* 57 (2015), S. 157–168. <http://dx.doi.org/10.1016/j.infsof.2014.08.007>. – ISSN 0950–5849
- [Pin03] PINTRICH, P.: A Motivational Science Perspective on the Role of Student Motivation in Learning and Teaching Contexts. In: *Journal of Educational Psychology* 95 (2003), S. 667–686
- [PLMD18] POQUET, Oleksandra; LIM, Lisa; MIRRIAH, Negin; DAWSON, Shane: Video and Learning: A Systematic Review (2007–2017). In: *Proceedings of the 8th International Conference on Learning Analytics and Knowledge*. New York, NY, USA: Association for Computing Machinery, 2018 (LAK '18). – ISBN 9781450364003. <http://dx.doi.org/10.1145/3170358.3170376>
- [PM86] PERKINS, D. N.; MARTIN, Fay: Fragile Knowledge and Neglected Strategies in Novice Programmers. In: *Papers Presented at the First Workshop on Empirical Studies of Programmers on Empirical Studies of Programmers*. USA: Ablex Publishing Corp., 1986. – ISBN 089391388X, S. 213–229
- [Pos] POSIT SOFTWARE, PBC: *RSTUDIO IDE - The most trusted IDE for open source data science*. <https://www.r-project.org/>, . – zuletzt aufgerufen am 31.01.2023
- [PRSG14] PIRKER, Johanna; RIFFNALLER-SCHIEFER, Maria; GÜTL, Christian: Motivational Active Learning: Engaging University Students in Computer Science Education. In: *Proceedings of the 2014 Conference on Innovation &*

- Technology in Computer Science Education*. New York, NY, USA: Association for Computing Machinery, 2014 (ITiCSE '14). – ISBN 9781450328333. <http://dx.doi.org/10.1145/2591708.2591750>
- [PZB88] PARASURAMAN, A P.; ZEITHAML, Valarie; BERRY, Leonard: SERVQUAL: A multiple- Item Scale for measuring consumer perceptions of service quality. In: *Journal of retailing* (1988)
- [Ram03] RAMSDEN, Paul: *Learning to Teach in Higher Education*. 2. Auflage. London: Routledge, 2003. – ISBN 978-0-415-30345-3
- [Raz13] RAZAK, Saquib: A Case for Course Capstone Projects in CS1. In: *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*. New York, NY, USA: Association for Computing Machinery, 2013 (SIGCSE '13). – ISBN 9781450318686. <http://dx.doi.org/10.1145/2445196.2445398>
- [Ren05] RENKL, Alexander: The Worked-Out Examples Principle in Multimedia Learning. Version: 2005. <http://dx.doi.org/10.1017/CB09780511816819.016>. In: MAYER, Richard Editor (Hrsg.): *The Cambridge Handbook of Multimedia Learning*. Cambridge: Cambridge University Press, 2005 (Cambridge Handbooks in Psychology). – DOI 10.1017/CBO9780511816819.016, S. 229–246
- [RH13] RACH, Stefanie; HEINZE, Aiso: Welche Studierenden sind im ersten Semester erfolgreich? In: *Journal für Mathematik-Didaktik* 34 (2013). <http://dx.doi.org/10.1007/s13138-012-0049-3>
- [RR19] ROSIENE, C. P.; ROSIENE, J. A.: To Flip or Not to Flip: Experiences with a Hybrid Approach. In: *2019 IEEE Frontiers in Education Conference (FIE)*, 2019
- [RRR03] ROBINS, Anthony; ROUNTREE, Janet; ROUNTREE, Nathan: Learning and Teaching Programming: A Review and Discussion. In: *Computer Science Education* 13 (2003), Nr. 2, 137-172. <http://dx.doi.org/10.1076/csed.13.2.137.14200>
- [RVEL19] RADFELDER, Oliver; VOSSEBERG, Karin; ERB, Ulrike; LIPSKOCH, Henrik: Informatik ist nicht nur Programmieren – aber ohne Programmieren ist nichts Informatik. In: THURNER, Veronika (Hrsg.); RADFELDER, Oliver (Hrsg.); VOSSEBERG, Karin (Hrsg.): *Tagungsband des 16. Workshops 'Software Engineering im Unterricht der Hochschulen'*, 2019 (SEUH 2019 - Software Engineering im Unterricht der Hochschulen). – ISSN 1613-0073
- [Sar14] SARAWAGI, Namita: A Flipped CS0 Classroom: Applying Bloom's Taxonomy to Algorithmic Thinking. In: *Journal of Computing Sciences in Colleges* 29 (2014), Nr. 6. – ISSN 1937-4771
- [SBD21] SCHMIDT-BORCHERDING, Florian; DRENDEL, Lara: Erklärvideos in der digitalen Hochschullehre: Welche Rolle spielen Sprecherpräsenz und Kohärenz für Lernerleben und Lernerfolg? (2021). <http://dx.doi.org/10.3278/HSL2108W>

- [Sch01] SCHULMEISTER, Rolf: *Virtuelle Universität Virtuelles Lernen*. München: Oldenbourg Wissenschaftsverlag, 2001. <http://dx.doi.org/10.1524/9783486598926>. – ISBN 9783486598926
- [SCZ15] SAHIN, Alpaslan; CAVLAZOGLU, Baki; ZEYTUNCU, Yunus: Flipping a College Calculus Course: A Case Study. In: *Educational Technology & Society* 18 (2015), S. 142–152
- [Sei14] SEIDEL, Susen: Defizitär oder Produktiv. Die Heterogenität der Studierenden. In: *Die Hochschule : Journal für Wissenschaft und Bildung* 23 (2014), Nr. 2, S. 6–21. <http://dx.doi.org/10.25656/01:16235>. – ISBN 978-3-937573-42-7
- [SF20] SPRINT, Gina; FOX, Erik: Improving Student Study Choices in CS1 with Gamification and Flipped Classrooms. In: *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. New York, NY, USA: Association for Computing Machinery, 2020 (SIGCSE '20). – ISBN 9781450367936
- [SFL20] STÜTZER, CM; FROHWIESER, D; LENZ, K: Selbstreguliertes Lernen hoch 2 – Neue Herausforderungen und mögliche Handlungsstrategien. Version: 2020. https://tu-dresden.de/zqa/ressourcen/dateien/publikationen/digitalisierung/2020_08_Gaaw_Wifek_Selbstreguliertes_Lernen_hoch_2.pdf?lang=de. In: *Potentiale und Herausforderungen digitaler Hochschulbildung*. Zentrum für Qualitätsanalyse (ZQA), TU Dresden, 2020
- [SHF21] SCHWARZ, R.; HELLMIG, L.; FRIEDRICH, S.: Informatikunterricht in Deutschland – eine Übersicht. In: *Informatik Spektrum* 44 (2021), S. 95 – 103. <http://dx.doi.org/10.1007/s00287-021-01349-9>
- [SHF22] SCHWARZ, Richard; HELLMIG, Lutz; FRIEDRICH, Steffen: INFORMATIK-MONITOR. (2022). https://informatik-monitor.de/fileadmin/GI/Projekte/Informatik-Monitor/Informatik-Monitor_2022/Informatik-Monitor_2022_2023_FINAL.pdf
- [SLR14] SKUDDER, Ben; LUXTON-REILLY, Andrew: Worked Examples in Computer Science. In: *Proceedings of the Sixteenth Australasian Computing Education Conference - Volume 148*. AUS: Australian Computer Society, Inc., 2014 (ACE '14). – ISBN 9781921770319, S. 59–64
- [SLS15] SETTLE, Amber; LALOR, John; STEINBACH, Theresa: Reconsidering the Impact of CS1 on Novice Attitudes. In: *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*. New York, NY, USA: Association for Computing Machinery, 2015 (SIGCSE '15). – ISBN 9781450329668. <http://dx.doi.org/10.1145/2676723.2677235>. 229–234
- [SM11] SCHULMEISTER, Rolf; METZGER, Christiane; SCHULMEISTER, Rolf (Hrsg.); METZGER, Christiane (Hrsg.): *Der Workload im Bachelor. Zeitbudget und Studierverhalten. Eine empirische Studie*. Münster: Waxmann, 2011

- [SMRS16] SHETH, Swapneel; MURPHY, Christian; ROSS, Kenneth A.; SHASHA, Dennis: A Course on Programming and Problem Solving. In: *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*. New York, NY, USA: Association for Computing Machinery, 2016 (SIGCSE '16). – ISBN 9781450336857. <http://dx.doi.org/10.1145/2839509.2844594>
- [SPV20] STEINHORST, Phil; PETERSEN, Andrew; VAHRENHOLD, Jan: Revisiting Self-Efficacy in Introductory Programming. In: *Proceedings of the 2020 ACM Conference on International Computing Education Research*. New York, NY, USA: Association for Computing Machinery, 2020 (ICER '20). <http://dx.doi.org/10.1145/3372782.3406281>. S. 158–169
- [SSB+17] SHEARD, Judy; SIMON; BUTLER, Matthew; FALKNER, Katrina; MORGAN, Michael; WEERASINGHE, Amali: Strategies for Maintaining Academic Integrity in First-Year Computing Courses. In: *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education*. New York, NY, USA: Association for Computing Machinery, 2017 (ITiCSE '17). – ISBN 9781450347044. <http://dx.doi.org/10.1145/3059009.3059064>. S. 244–249
- [SVMP98] SWELLER, John; VAN MERRIENBOER, Jeroen J. G.; PAAS, Fred: Cognitive Architecture and Instructional Design. In: *Educational Psychology Review* 10 (1998), S. 251–296. <http://dx.doi.org/10.1023/a:1022193728205>
- [SW06] STEVENSON, Daniel E.; WAGNER, Paul J.: Developing Real-World Programming Assignments for CS1. In: *SIGCSE Bull.* 38 (2006), Nr. 3, S. 158–162. <http://dx.doi.org/10.1145/1140123.1140167>. – ISSN 0097–8418
- [Swe06] SWELLER, John: The worked example effect and human cognition. In: *Learning and Instruction* 16 (2006), Nr. 2, S. 165–169. <http://dx.doi.org/10.1016/j.learninstruc.2006.02.005>. – ISSN 0959–4752
- [TDH16] TARIMO, William T.; DEEB, Fatima A.; HICKEY, Timothy J.: A Flipped Classroom with and Without Computers. In: ZVACEK, Susan (Hrsg.); RESTIVO, Maria T. (Hrsg.); UHOMOIBHI, James (Hrsg.); HELFERT, Markus (Hrsg.): *Computer Supported Education*. Cham: Springer International Publishing, 2016. – ISBN 978–3–319–29585–5
- [The] THE R FOUNDATION: *The R Project for Statistical Computing*. <https://www.r-project.org/>, . – zuletzt aufgerufen am 31.01.2023
- [The21] THE NBGITPULLER TEAM: *nbgitpuller*. <https://jupyterhub.github.io/nbgitpuller/>, 2021. – zuletzt aufgerufen am 20.01.2023
- [Uni] UNIVERSITÄT OSNABRÜCK: *Geförderte Lehrprojekte*. <https://www.uni-osnabrueck.de/studium/lehrkolleg/preisgekroente-projekte-studium-und-lehre/#c246178>, . – zuletzt aufgerufen am 26.09.2022
- [Van15] VANDEGRIFT, Tammy: Supporting Creativity and User Interaction in CS 1 Homework Assignments. In: *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*. New York, NY, USA: Association

- for Computing Machinery, 2015 (SIGCSE '15). – ISBN 9781450329668. <http://dx.doi.org/10.1145/2676723.2677250>
- [VCEV15] VOSSEBERG, Karin; CZERNIK, Sofie; ERB, Ulrike; VIELHABER, Michael: Projektorientierte Studieneingangsphase - Das Berufsbild der Informatik und Wirtschaftsinformatik schärfen. In: *HDI 2014 : Gestalten von Übergängen 2015* (2015), Nr. 9, S. 169 – 177
- [VF18] VOGEL, Dita; FUNCK, Barbara J.: Immer nur die zweitbeste Lösung? Protokolle als Dokumentationsmethode für qualitative Interviews. In: *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research* 19 (2018), Nr. 1, S. 29. <http://dx.doi.org/10.17169/fqs-19.1.2716>. – ISSN 1438–5627
- [VHI14] VIHAVAINEN, Arto; HELMINEN, Juha; IHANTOLA, Petri: How Novices Tackle Their First Lines of Code in an IDE: Analysis of Programming Session Traces. In: *Proceedings of the 14th Koli Calling International Conference on Computing Education Research*. New York, NY, USA: Association for Computing Machinery, 2014 (Koli Calling '14). – ISBN 9781450330657. <http://dx.doi.org/10.1145/2674683.2674692>
- [vir] VIRTUOS SUPPORT-TEAM, UNIVERSITÄT OSNABRÜCK: *StudIP*. <https://studip.uni-osnabrueck.de/dispatch.php/start>, . – zuletzt aufgerufen am 31.01.2023
- [WA17] WANG, Jiahui; ANTONENKO, Pavlo D.: Instructor presence in instructional video: Effects on visual attention, recall, and perceived learning. In: *Computers in Human Behavior* 71 (2017), 79-89. <http://dx.doi.org/10.1016/j.chb.2017.01.049>. – ISSN 0747–5632
- [Wal14] WALGENBACH, Katharina: *Heterogenität - Intersektionalität - Diversity in der Erziehungswissenschaft*. 1. Auflage. Stuttgart, Deutschland: Verlag Barbara Budrich, 2014. <http://dx.doi.org/10.36198/9783838585468>. – ISBN 9783825285463
- [WB19] WOITKOWSKI, David; BREITKOPF, Sandra: Fähigkeitsselbstkonzept und Lernerfolg im ersten Fachsemester Physik. In: *die hochschullehre* 5 (2019), S. 855–876
- [Web10] WEBLER, Wolff-Dietrich: Eingangsphase zu welchem Ausgang? – Studienziele und deren anteilige Einlösung in der Studieneingangsphase. In: *HSW - Hochschulwesen* 4-5 (2010), S. 121–133
- [Wei99] WEINERT, Franz Z.: Aus Fehlern lernen und Fehler vermeiden lernen. In: ALTHOF, Wolfgang (Hrsg.): *Fehlerwelten: Vom Fehlermachen und Lernen aus Fehlern. Beiträge und Nachträge zu einem interdisziplinären Symposium aus Anlaß des 60. Geburtstags von Fritz Oser*. Wiesbaden: VS Verlag für Sozialwissenschaften, 1999. – ISBN 978–3–663–07878–4, S. 101–109

- [Wey18] WEYERS, Stefan: Inverted Classroom meets Gamification - Einsatz eines Bonuspunktesystems in einer Grundlagenveranstaltung. In: *Inverted Classroom Vielfältiges Lernen Begleitband zur 7. Konferenz Inverted Classroom and Beyond 2018*, 2018, S. 245–249
- [Win96] WINSLOW, Leon E.: Programming Pedagogy—a Psychological Overview. In: *SIGCSE Bull.* 28 (1996), Nr. 3, 17–22. <http://dx.doi.org/10.1145/234867.234872>. – ISSN 0097–8418
- [WLR97] WILSON, Keithia L.; LIZZIO, Alf; RAMSDEN, Paul: The development, validation and application of the Course Experience Questionnaire. In: *Studies in Higher Education* 22 (1997), Nr. 1, 33–53. <http://dx.doi.org/10.1080/03075079712331381121>
- [WN05] WICENTOWSKI, Richard; NEWHALL, Tia: Using Image Processing Projects to Teach CS1 Topics. In: *SIGCSE Bull.* 37 (2005), Nr. 1. <http://dx.doi.org/10.1145/1047124.1047445>. – ISSN 0097–8418
- [WS14] WEIDLICH, Joshua; SPANNAGEL, Christian: Die Vorbereitungsphase im Flipped Classroom - Vorlesungsvideos versus Aufgaben. In: RUMMLER, Klaus (Hrsg.): *Lernräume gestalten - Bildungskontexte vielfältig denken*, Waxmann Münster, 2014, S. 237–248
- [Zim66] ZIMBARDO, Philip G.: DIVISION OF PSYCHOLOGY: THE COGNITIVE CONTROL OF MOTIVATION. In: *Transactions of the New York Academy of Sciences* 28 (1966), Nr. 7 Series II, S. 902–922. <http://dx.doi.org/10.1111/j.2164-0947.1966.tb02395.x>
- [ZLM⁺09] ZAPPE, Sarah; LEICHT, Robert; MESSNER, John; LITZGINER, Thomas; LEE, Hyeon W.: 'Flipping' the classroom to explore active learning in a large undergraduate course. In: *Proceedings of the 2009 American Society for Engineering Education Annual Conference and Exhibition* 92 (2009)
- [ZPM⁺19] ZHI, Rui; PRICE, Thomas W.; MARWAN, Samiha; MILLIKEN, Alexandra; BARNES, Tiffany; CHI, Min: Exploring the Impact of Worked Examples in a Novice Programming Environment. In: *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. New York, NY, USA: Association for Computing Machinery, 2019 (SIGCSE '19). – ISBN 9781450358903. <http://dx.doi.org/10.1145/3287324.3287385>

Abbildungsverzeichnis

4.1	Ausschnitt einer Stufe aus Modell nach Boulding et al. [BKSZ93], eigene Darstellung	41
4.2	Aus den Interviewantworten abgeleitete Kategorien	46
6.1	Durchschnittliche Anzahl an Teilnehmenden pro Coding Class in den Präsenz- und Online-Coding Classes im Wintersemester 2021/22 sowie Kapazitätsgrenze für Coding Classes aufgrund pandemiebedingter Maßnahmen	63

A Veröffentlichungen

Im Folgenden befindet sich ein Abdruck der Publikationen, die in Kapitel 3.2 näher erläutert wurden. Layout und Zitierweise entsprechen den Vorgaben der jeweiligen Publikationsstelle und können daher von der Gestaltung der Dissertation selbst abweichen.

A.1 Beitrag 1

Übersicht zu Beitrag 1

Titel:	Heterogenitätsbewusste digitale Gestaltung eines Einführungsmoduls in der Informatik
--------	--

Autoren:	Elisaweta Ossovski Michael Brinkmeier Ann-Katrin Becker Laura Hembrock Daniel Kalbreyer Sven Klecker
----------	---

Jahr:	2022
-------	------

Status:	veröffentlicht
---------	----------------

Bibliographische Informationen:	Ossovski, Elisaweta; Brinkmeier, Michael; Becker, Ann-Katrin; Hembrock, Laura; Kalbreyer, Daniel; Klecker, Sven: Heterogenitätsbewusste digitale Gestaltung eines Einführungsmoduls in der Informatik. In: MedienPädagogik: Zeitschrift für Theorie und Praxis der Medienbildung Nr. 48 (2022), Digitalisierung als Katalysator für Diversität an Hochschulen et vice versa, S. 173–193. http://dx.doi.org/10.21240/mpaed/48/2022.06.14.X
---------------------------------	--

Link zur Veröffentlichung:	https://www.medienpaed.com/article/view/1280
----------------------------	---

MedienPädagogik

Zeitschrift für Theorie und Praxis der Medienbildung
www.medienpaed.com

ISSN 1424-3636

Themenheft Nr. 48: Digitalisierung als Katalysator für Diversität an Hochschulen et vice versa. Herausgegeben von Natalia Reich-Stiebert, Jennifer Raimann, Carsten Thorbrügge und Len Ole Schäfer

Heterogenitätsbewusste digitale Gestaltung eines Einführungsmoduls in der Informatik

Elisaweta Ossovski¹ , Michael Brinkmeier¹, Ann-Katrin Becker¹, Laura Hembrock¹, Daniel Kalbreyer¹  und Sven Klecker¹

¹ Universität Osnabrück

Zusammenfassung

In Einführungsmodulen, die sowohl als Fachmodul als auch als Importmodul für verschiedene Studiengänge angeboten werden, unterscheiden sich die Vorkenntnisse, Ziele und Motivationen der Teilnehmenden erheblich. Im Zuge eines Projekts zur Digitalisierung der Lehre in der Informatik wurde daher über zwei Jahre ein Konzept für ein solches Einführungsmodul der Informatik entwickelt, das individuellere Lernangebote bereithält und synchrone mit asynchronen Lernphasen verbindet. Dabei wurden die Konzepte des Flipped Classroom und der Projektarbeit als didaktische Basis verwendet. Als Programmierumgebung und zur Bereitstellung der Lernmaterialien wurden Jupyter Notebooks genutzt. Durch verschiedene Evaluationsmethoden sowohl mit den Studierenden als auch mit den studentischen Tutor:innen, die diese betreuen, konnten nach dem ersten Jahr Aspekte identifiziert werden, die im zweiten Jahr weiter verbessert wurden. Trotz der Belastungen durch das digitale Coronasemester ist ergänzt um einen Gamification-Ansatz ein Konzept entstanden, das Studierenden individuelle Lernmöglichkeiten bietet und zu einer hohen Zufriedenheit führt. Insbesondere scheint es durch asynchrone Lernmaterialien gelungen zu sein, die bei klassischen Vorlesungen häufig fehlende Vor- und Nachbereitung der Studierenden zu fördern.

Heterogeneity-Aware Digital Design of an Introductory Module in Computer Science

Abstract

In introductory modules offered for different degree programmes with different impact on the study plan, the previous knowledge, goals and motivations of the participants differ considerably. As a part of a project on the digitalisation of teaching in computer science, a concept for such an introductory computer science module was therefore developed over two years. It provides individualised learning opportunities and combines synchronous with asynchronous learning phases. The concepts of flipped classroom and project work were used as a didactical basis, while Jupyter Notebook was used as a programming

environment and for providing learning material. Through various evaluation methods involving both the students and the student tutors, who supervise them, it was possible to identify aspects that were improved further in the second year after the first year. Despite the burdens of the digital semester caused by the COVID-pandemic, a concept has been developed that offers more individual learning opportunities to the students and leads to a high level of satisfaction. A gamification concept was also included. In particular, asynchronous learning materials seem to have succeeded in encouraging students to prepare and follow up on their learning, which is often lacking in traditional lectures.

1. Einleitung

Studiengangübergreifende Einführungsmodule stellen für die Lehre eine besondere Herausforderung dar. Besonders in der Informatik, die als Schulfach aktuell noch überwiegend den Status eines Wahl- oder Wahlpflichtfaches hat (vgl. Schwarz, Hellmig, und Friedrich 2021), sind Lehrende mit einer starken Heterogenität beim Vorwissen der Studierenden konfrontiert. Auch Freizeitaktivitäten tragen dazu bei, dass einige Studienanfänger:innen bereits über umfassende Kenntnisse verfügen. In studiengangübergreifenden Modulen unterscheiden sich aber auch die Ziele und dementsprechend die Motivationen der Studierenden. Während das Modul für Informatikstudierende die Basis für ihr weiteres Studium bildet, zweifeln einige Studierende, die das Modul innerhalb ihres Fachs als Ergänzung belegen müssen, am Nutzen dieser Veranstaltungen für sie.

An der Universität Osnabrück wird aktuell nur ein solches Einführungsmodul angeboten, bei dem Studierende diverser Studiengänge sowohl eine Einführung in die Programmierung als auch in die Algorithmik erhalten. Als aktive Tätigkeit erfordert insbesondere das Programmieren dabei eine kontinuierliche Auseinandersetzung mit den entsprechenden Kompetenzen. Diese Aspekte haben dazu geführt die Lehre im Modul entsprechend den Erkenntnissen der Lehr-Lern-Forschung umzugestalten. Diversität und Digitalisierung wirkten dabei gegenseitig als Katalysatoren, da die Digitalisierung zum einen Chancen bietet der Heterogenität besser gerecht zu werden, zum anderen eröffnen Aspekte der Digitalisierung neue Möglichkeiten in der Lehre.

Zentrale Fragen der begleitenden Forschung zur Umgestaltung des Moduls waren dabei, wie die Lehre vor dem Hintergrund der Heterogenität unter Berücksichtigung der personellen Kapazitäten verbessert werden kann und welche Massnahmen die bessere Nutzung praktischer Lerngelegenheiten zum Programmieren ermöglichen.

2. Theoretischer Hintergrund

Das in diesem Beitrag vorgestellte Konzept ist auf der Basis bisheriger Forschung zu ähnlichen Modulen und allgemeiner didaktischer Konzepte entstanden. Dabei sind besonders die Ansätze des Flipped Classroom sowie die bereits seit Längerem eingesetzte, aber oft unterschiedlich definierte Projektarbeit von Bedeutung.

2.1 Flipped Classroom

Im Flipped-Classroom-Konzept wird die Wissensvermittlung, die für die Lernenden typischerweise passiv erfolgt, in die Selbstlernphase verlagert, während die Präsenzzeit als Übungsphase genutzt wird (vgl. Bergmann und Sams 2012; Lage, Platt, und Treglia 2000). Dies hat den Vorteil, dass in der oft als schwieriger empfundenen Anwendungsphase eine direkte Reaktion auf Probleme und Schwierigkeiten möglich ist (vgl. Bergmann und Sams 2012). Der erste «Flipped Classroom» unter diesem Namen wurde von Lage, Platt und Treglia in einem Wirtschaftskurs realisiert (2000). Die Studierenden erhielten Vorlesungsfolien mit einer Tonspur sowie weitere Materialien. Einige Jahre später wandten Bergmann und Sams (2012) dieses Konzept auf ihren Schulunterricht an, nachdem sie festgestellt hatten, dass Selbstlernvideos bei Schüler:innen, die den Unterricht verpasst hatten, erfolgreich waren. Seitdem wurde das Konzept des Flipped Classroom in verschiedenen Szenarien in Bildungskontexten eingesetzt und überwiegend erfolgreich evaluiert.

2.2 Projektarbeit

Nach Knoll (1997) hat das Konzept des projektorientierten Lernens eine lange Geschichte erfahren, wobei es keine einheitliche Definition, sondern unterschiedliche Ansätze gibt (vgl. Kolmos 1996). Das Konzept des Projektunterrichts, das oft auf den Pädagogen und Philosophen John Dewey zurückgeführt wird, entstand im Grunde als Gegenbewegung zum Frontalunterricht, wurde aber auch im universitären Kontext eingesetzt, wenn es darum ging, konkrete Probleme zu bearbeiten und praktisch zu handeln (vgl. Knoll 1997; Kolmos 1996). Im Rahmen eines «Learning-by-Doing»-Ansatzes beinhaltet die Projektarbeit darüber hinaus eine soziale Dimension. Auch durch den engen Bezug zum problemorientierten Lernen sollen verschiedene theoretische Inhalte gezielt und mit sozialer Interaktion angewendet werden. Gudjons (2001) stellt im Rahmen der Handlungsorientierung individuelle Handlungsmöglichkeiten für tatsächliche Fragestellungen in Bezug auf Zeit und Ziele als wesentliche Merkmale für Projektarbeit fest.

Kolmos (1996) definiert verschiedene Projekttypen, von denen einer als «aufgabenbasiertes Projekt» bezeichnet wird und vorsieht, dass die Lehrenden das Projekt auswählen und planen, indem sie den Lernenden Aufgaben stellen. Nicht nur das

Projekt selbst, sondern auch das Thema und die Methoden werden dabei von den Betreuenden gewählt. Diese Art von Projekten sei besonders im ersten Studienjahr geeignet, da für offenere Projekte ein gewisser Wissensstand erforderlich ist (vgl. ebd.).

2.3 Forschungsstand in Bezug auf Informatikeinführungsmodule

Zum Einsatz von Flipped-Classroom-Konzepten in Einführungsmodulen der Informatik gibt es bereits einige Studien, die sich vor allem auf den Prüfungserfolg und die Zufriedenheit der Studierenden konzentrieren.

Horton et al. (2014) und Campbell et al. (2014) verglichen einen videobasierten Flipped Classroom eines CS1-Kurses (Computer Science 1) mit der Programmiersprache Python und einem Objects-First-Ansatz mit bisheriger klassischer Lehre. Dabei verbesserten sich die Noten der Studierenden im Flipped Classroom-Setting und die Studierenden bevorzugten dieses gegenüber der klassischen Lehre. Festgestellte geringere Anwesenheitsquoten konnten durch die fehlende Notwendigkeit des Besuchs bei gutem Verständnis der Materialien erklärt werden. Ähnliche Effekte wurden auch von Lockwood und Esselstein (2013) und Sarawagi (2014) festgestellt.

Tarimo, Deeb und Hickey (2016) sowie Largent (2013) konnten feststellen, dass Studierende in Einführungskursen computerbasierte Übungen bevorzugen und dass Ansätze zum aktiven Lernen generell positiv bewertet werden. Die Zufriedenheit mit dem Konzept wurde auch von den Studierenden in den Untersuchungen von Rosiene und Rosiene (2019) und Hodges (2019) bestätigt, die jeweils einen Blended-Learning-Ansatz testeten, bei dem nur ein Teil der Lehreinheiten im Flipped Classroom durchgeführt wurde.

Mohamed (2020) erhielt gutes Feedback von Studierenden, die an einem teilweise geflippten Lehransatz mit der Programmiersprache Processing und Pair-Programming teilnahmen, konnte aber keine signifikanten Unterschiede zwischen Studierenden mit unterschiedlichem Vorwissen oder aus unterschiedlichen Fächern feststellen. Latulipe et al. (vgl. Latulipe, Long, und Seminario 2015; Latulipe, Rorrer, und Long 2018) untersuchten diverse Auswirkungen eines Flipped Classroom in einem CS1-Modul. Das überfachlich-soziale Lernen wurde durch die Gruppenarbeit gestärkt, ohne dass dies Auswirkungen auf die Endnote hatte.

In Koeppel et al. (2015) und Koeppel et al. (2016) stellen die Autoren Strategien zur Umsetzung verschiedener Komponenten des Flipped Classroom vor. Sie empfehlen unter anderem, den Studierenden zu signalisieren, dass jede Mitarbeit anerkannt wird, ihnen individuelle Hilfe anzubieten und Lösungen der Studierenden in Besprechungen zu nutzen. Hakimzadeh, Adaikkalavan und Batzinger (2011) haben gute Erfahrungen mit einer speziellen Raumeinrichtung namens *SCALE-UP* für aktives Lernen gemacht, wobei diese dem Feedback der beteiligten Personen nach aktive Lerntätigkeiten gefördert hat. Der Raum war mit Gruppentischen entlang der Wände

sowie für alle Gruppenteilnehmenden sichtbaren Bildschirmen ausgestattet. Ergänzt wurden diese durch einige Computerarbeitsplätze sowie zwei Grossbildschirme und einen Arbeitsplatz für die Lehrperson.

Auch aus Sicht der Studierenden haben sich Flipped Classroom-Konzepte erfolgreich bewährt und seitens der Lehr-Lern-Forschung gibt es bereits einige Empfehlungen für die konkrete Gestaltung. Auch Projektarbeit wurde bereits in unterschiedlichen Ausprägungen in der Informatiklehre eingesetzt und evaluiert. Dadurch konnte eine Basis für die Gestaltung der neuen Abgabearbeiten entwickelt werden.

Wicentowski und Newhall (2005) setzten Bildverarbeitungsprojekte in einem CS1-Modul mit 42 Studierenden ein. Diese gaben an, Freude an den Aufgaben gehabt zu haben, wobei sie diese meist zu zweit bearbeiteten und als hilfreich für das Verständnis empfanden. In ähnlicher Weise verfolgten Bruce, Danyluk und Murtagh (2001) einen grafikbasierten Objects-First-Ansatz, da sie bereits aus früheren Kursen die Erfahrung gemacht hatten, dass visuelles Feedback zu weniger Programmierfehlern führt.

Kussmaul (2008) beschreibt Projektaufgaben mit konkreten Anforderungen an die verwendeten Konzepte und seine Erfahrungen aus dem Einsatz dieser in mehreren CS1- und CS2-Kursen. Er empfiehlt die Erwartungen zu Beginn des Kurses detaillierter zu beschreiben und zum Ende hin offener zu werden. Studierende seien zufriedener, wenn sie die Möglichkeit haben eigene Entscheidungen zu treffen. Daher sollten Projekte mit unabhängigen Aufgaben kombiniert werden. In ähnlicher Weise beschreiben Keen und Mammen (2015) ein grösseres Semesterprojekt mit Meilensteinen und ebenfalls grafischen Themen, wobei die Anforderungen mit der Zeit offener werden. Die Autoren stellten fest, dass sich die Dekompositionsfähigkeiten der Studierenden verbessert haben. Weitere Erfahrungen mit grafischen Projekten wurden von Razak (2013) sowie Matzko und Davis (2006) gemacht.

Auf dieser bisherigen Forschung zum Flipped Classroom und zu grafikbasierter Projektarbeit aufbauend wurde ein Kurskonzept für ein Einführungsmodul in der Informatik erstellt, das im Laufe von zwei Jahren weiter verbessert und angepasst wurde. Die grosse Anzahl an Studierenden und die Heterogenität dieser stellten dabei eine besondere Herausforderung dar.

3. Konzept

Basierend auf dem Forschungsstand in ähnlichen Modulen und den Erkenntnissen der Lehr-Lern-Forschung wurde ein Modulkonzept entwickelt. Dessen Komponenten wurden auch der Digitalisierung und Diversität angepasst, wobei diese Aspekte sich wechselseitig beeinflusst haben.

3.1 Ausgangssituation

Bei dem beschriebenen Modul handelt es sich um die *Einführung in Algorithmen und Datenstrukturen* an der Universität Osnabrück. Das Modul wird jährlich jeweils im Wintersemester angeboten und von etwa 550 Studierenden belegt. Bei diesen handelt es sich zum überwiegenden Teil um Erstsemesterstudierende aus Studiengängen der Informatik (Bachelor of Science, 2-Fächer-Bachelor mit Lehramtsoption, B. Sc. Eingebettete Softwaresysteme), Cognitive Science, (Umwelt-) Systemwissenschaften sowie Geo- und Wirtschaftsinformatik. Zusätzlich müssen Studierende der Mathematik im Bachelor of Science oder 2-Fächer-Bachelor auch beim Profil Lehramt an Gymnasien das Modul belegen, jedoch im dritten oder fünften Semester. Ausserdem belegen einige weitere Studierende das Modul als Wahlfach, z. B. im Bachelor of Science Physik, oder im Rahmen von Angleichungsstudien auch im Masterstudium z. B. im Studiengang Geoinformatik nach einem Bachelor in Geografie.

Vor der Umstrukturierung des Moduls wurden die 9 ECTS durch zwei klassische Vorlesungen je 90 Minuten (2x2 SWS) sowie eine klassische und für die Studierenden überwiegend passive Übung (2 SWS), die zu vier Uhrzeiten an einem Tag angeboten wurde, abgedeckt. Für die Zulassung zur Klausur mussten etwa 13 Aufgabenblätter im wöchentlichen Rhythmus in Teams von jeweils zwei Studierenden bearbeitet und in einem 30-minütigen Testat einer studentischen Tutorin oder einem Tutor präsentiert werden. Dabei durften nur in einem Aufgabenblatt weniger als 50% der möglichen Punkte erreicht werden.

Im ersten Jahr erfolgte die Umstrukturierung vor allem mit dem Ziel, die als passiv empfundene Übung anzupassen und die Abgabearbeiten offener zu gestalten, um mehr Kreativität zu ermöglichen. Die Übung hatte an Attraktivität für die Studierenden verloren, weil zunehmend mehr handlungsorientierte Beispiele bereits in der Vorlesung thematisiert wurden und damit die Kerninhalte der Übung zunehmend geringer wurden. Auch die regelmässig präsentierten Lösungen vergangener Aufgabenblätter schienen für viele Studierende uninteressant. Für den Fall, dass sie diese bereits selbst erfolgreich gelöst hatten, stellte die Präsentation lediglich eine Wiederholung dar. Hatten sie jedoch Schwierigkeiten mit den Aufgaben, lagen die Probleme meist eher im Prozess der Lösung, sodass die Präsentation der fertigen Lösung keinen echten Mehrwert darstellte. Dies führte immer häufiger zum Wunsch der Studierenden die Lösungen erst am Ende der Übung zu besprechen, um diese vorzeitig verlassen zu können. Die Besuchsintention lag daher stärker darauf, keine Hinweise zu den Aufgaben zu verpassen als darauf, den Stoff tatsächlich zu vertiefen. Ausserdem empfanden die Lehrenden insbesondere die Nachbereitung der Vorlesung durch die Studierenden als zu gering und wollten hier Anreize schaffen, um den Studierenden Materialien zur Verfügung zu stellen, mithilfe derer eine Vor- und Nachbereitung gelingen kann. Die These, dass die Selbstlernzeiten grundsätzlich in für die zu erwerbenden ECTS zu geringem Umfang stattfinden, wird auch durch die

ZEITLast-Studie von Schulmeister und Metzger (2011) bestätigt. Diese untersuchten die tatsächlichen Lernzeiten der Studierenden sowohl in Lehrveranstaltungen als auch ausserhalb dieser in Abgrenzung zu den häufig überschätzten Zeitangaben in Evaluationen.

Im zweiten Jahr erfolgte eine Anpassung des entwickelten Konzepts, um die Erkenntnisse aus der Evaluation des ersten Jahres einfließen zu lassen. Dabei gewann die Digitalisierung als Katalysator eine weitere Bedeutung: Durch die Coronapandemie musste auf Präsenzlehre verzichtet werden. Im Zuge der Entwicklung eines vollständig digitalen Konzepts wurde dann nicht nur die Übung, sondern die komplette Veranstaltung zu einem Flipped Classroom umgewandelt und statt der Vorlesung wurde asynchrones Lernmaterial über Jupyter Notebooks zur Verfügung gestellt. Abbildung 1 zeigt die zeitliche Abfolge und gibt einen Überblick über die Inhalte der folgenden Abschnitte.



Abb. 1: Zeitleiste der Anpassungen sowie Zuordnung der Abschnitte.

3.2 Von der Vorlesung zu asynchronen Lernmaterialien

Im ersten Jahr der Umstrukturierung blieb die Vorlesung weitestgehend klassisch erhalten. Im zweiten Jahr wurde die Lehre jedoch von der Notwendigkeit der Umstellung durch die Coronapandemie konfrontiert. Für die Lehrenden bestand die Wahl zwischen hybrider Lehre, bei der aufgrund der Abstandsregeln ca. 1/7 der Studierenden bei der Vorlesung hätte anwesend sein können oder rein digitalen Alternativen. Da auch die klassische Videoaufzeichnung aus Sicht der Lehrenden die Möglichkeiten digitaler Lernmaterialien nicht ausreichend berücksichtigt, wurden unter Berücksichtigung diverser innerhalb des Projekts entstandener Ideen wie individuellen Lernpfaden, Materialien zur Vor- und Nachbereitung sowie aktivem Trial-and-Error asynchron nutzbare Lernmaterialien namens *Lessons* erstellt. Diese enthalten Verlinkungen zu thematischen Voraussetzungen und möglichen anschliessenden Themen. Sie bestehen grundsätzlich aus vertonten Folien sowie Markdown- und Codezellen in Jupyter Notebooks. Die vertonten Folien lassen sich wie ein Video ansehen, jedoch sind auch ein angepasstes Tempo und Pausen möglich. Einen grossen Vorteil bieten die vertonten Folien gegenüber Videos in ihrer einfachen Bearbeitung. So muss bei einer kleinen Änderung nicht das ganze Video erneut aufgenommen oder geschnitten

werden, sondern die Aufnahme der Audiospur für eine bestimmte Folie ist ausreichend. In den Markdown- und Codezellen folgen dann konkrete Anwendungen der in den Folien behandelten Inhalte. Dabei können Codeabschnitte entsprechend der Anweisungen in den Markdownzellen durch die Lernenden verändert oder ergänzt werden, um ein bestimmtes Ziel zu erreichen. Innerhalb des Jupyter Notebooks kann dann direkt eine Ausführung erfolgen, um die Lösung zu prüfen. Um den Lessons einen Rahmen zu geben, wurden neben einer thematischen Ordnerstruktur Wochenpläne veröffentlicht, in denen jeweils für eine Woche die Grundlagen, die Kerninhalte sowie freiwillige Zusätze verlinkt waren und die einen Überblick geben. Die asynchronen Lernmaterialien wurden um ein wöchentliches Livestreaming ergänzt, bei dem im Chat gestellte Fragen zu den Inhalten der Lessons diskutiert wurden.

3.3 Von der Übung zur Coding Class

Mit der Umgestaltung der Übung sollte vor allem aus der vorher passiven Übung eine aktive Übung werden, bei der die Studierenden mithilfe von Tutor:innen die konkrete Anwendung des Gelernten, vor allem das Programmieren, üben sollten. Zur Verdeutlichung der neuen Ausrichtung wurde der Name *Coding Class* gewählt. Ausserdem sollten sowohl das soziale Lernen zwischen den Studierenden als auch die Vor- und Nachbereitung gestärkt werden. Dazu wurden zusätzliche Aufgaben erstellt, deren Bearbeitung auf die folgenden Abgabenaufgaben vorbereiten soll. Im ersten Jahr wurden dazu ein gesondertes Aufgabenblatt sowie über eine Courseware im Lern-Management-System zusätzliche Materialien zur Verfügung gestellt, die vor dem Besuch der Coding Class bearbeitet werden sollten. Die meisten Coding Classes im ersten Jahr fanden in einem speziell dafür eingerichteten Raum statt, bei dem in der Mitte Gruppentische und entlang der Wände aufgestellte Tische mit Computern zu finden waren. Ein Smartboard für die Lehrenden ergänzte die technische Ausstattung.

Durch die zusätzlichen Materialien und die Erwartung an eine Vorbereitung auf die Coding Class neben dem Besuch der Vorlesung berichteten viele Studierende von einer zeitlichen Überforderung. Als Folge wurden die Coding Classes im zweiten Jahr weiterentwickelt. Die asynchronen Lernmaterialien deckten durch die Möglichkeiten kurze Codebeispiele zu testen und die theoretischen Inhalte direkt beim Lernen praktisch auszuprobieren, bereits die Vor- und Nachbereitung der theoretischen Inhalte, die zuvor in der Vorlesung vermittelt wurden, ab. Zusätzlich wurden die Aufgaben für die Coding Classes in das Aufgabenblatt mit den Abgabenaufgaben integriert und stärker auf diese abgestimmt. Häufig stellen die Abgabenaufgaben dabei eine direkte Erweiterung der Coding Class Aufgaben dar. Ausserdem wurde für jede Woche ein gemeinsamer Einstieg («Intro») in die Arbeitsphase der Coding Class erarbeitet, bei dem die Tutor:innen gemeinsam mit den Studierenden den Anfang einer Coding Class-Aufgabe oder ein ähnliches Beispiel bearbeiten und dabei auch bereits in den

Lessons thematisierte Fragen beantworten können. Zum Ende der Coding Class fragten die Tutor:innen die Studierenden bei fehlenden Lösungen aus der Arbeitsphase, ob die Präsentation einer Lösung durch sie gewünscht sei.

3.4 Von Aufgabenblättern über Meilensteine zu dynamischen Abgabezeiträumen

Im ersten Jahr der Umgestaltung wurde ein Meilensteinansatz mit offenen Aufgaben verfolgt, bei dem nur alle drei Wochen eine Abgabe erfolgte. Innerhalb dieser Meilensteine wurden für jede Aufgabe jeweils Grundanforderungen definiert, die in jedem Fall von der Lösung abgedeckt sein sollten, sowie Ideen zur Erweiterung zur Verfügung gestellt. Durch die erfolgreiche Erweiterung der Grundanforderungen konnten so Bonuspunkte erworben werden, die zusammen mit den Klausurpunkten die Endnote bestimmten. Es waren maximal zehn Bonuspunkte, entsprechend etwa zwei Teilnoten über die vier Meilensteine zu erwerben. Für die Erfüllung der Grundanforderungen wurden keine Punkte mehr vergeben, sondern mit mehreren Anpassungen während des Semesters inhaltliche Kriterien festgestellt, die zum Bestehen führten.

Im zweiten Jahr führten die Erfahrungen zu einem kombinierten Ansatz. Während in den ersten Wochen eine enge Taktung von einer Woche zwischen den Abgaben die kontinuierliche Mitarbeit gewährleisten sollte und damit Schwierigkeiten schon früh erkannt wurden, bot die Ausweitung der Bearbeitungszeiten zu zweiwöchentlichen Abgaben bis hin zu einem dreiwöchigen offenen Projekt die Gelegenheit, komplexere Programme zu entwerfen. Durch die höhere Anzahl an Aufgabenblättern und Testaten wurde den Studierenden wieder erlaubt in einem Testat weniger als 50 % der erreichbaren und wieder vergebenen Punkte zu erreichen.

Für das Projekt, bei dem in jedem Fall mindestens die Hälfte der möglichen Punkte erreicht werden mussten, wurden fünf Spielprinzipien vom statischen Brettspiel bis zum Dungeon Crawler, einem komplexeren Spiel mit aktiver Spielerfigur und aktiven Gegnern, zur Wahl gestellt, die beliebig implementiert werden durften. Die Projekte wurden dabei nach vermuteter Schwierigkeit für die Umsetzung geordnet, mit einem konkreten Beispiel und wie zuvor bei den Meilensteinen mit Grundanforderungen und Erweiterungsideen ausgegeben. Dadurch konnten bis zu 5 Bonuspunkte entsprechend einer Teilnotenstufe für die Abschlussnote erworben werden.

3.5 Ergänzung durch Gamification-Konzept

Ergänzend wurde im zweiten Jahr ein kooperatives Gamification-Konzept umgesetzt, um die aktive Mitarbeit der Studierenden zu fördern und kooperatives Verhalten zu belohnen. Dabei können die Studierenden durch Aktivität in synchronen Veranstaltungen wie Coding Classes oder dem Livestreaming, aber auch durch Forenbeiträge, kreative Zusatzabgaben oder Teilnahme an Forschungsprojekten zur dauerhaften

Verbesserung der Lehre gemeinsam Punkte sammeln. Beim Erreichen bestimmter Punktestände werden Belohnungen für alle Studierenden freigeschaltet, die beispielsweise eine Verlängerung der Klausurzeit zum Ausgleich technischer Schwierigkeiten bei Studierenden oder die Möglichkeit Bonuspunkte für die Abschlussnote zu erwerben, bedeuten. Das konkrete Konzept sowie dessen Ergebnisse werden in einem anderen Beitrag beschrieben (vgl. Ossovski et al. 2021).

4. Evaluationsmethodik

Die Umgestaltung des Moduls wurde auf unterschiedliche Arten evaluiert. Dabei fanden sowohl während als auch zum Ende der jeweiligen Semester Befragungen der beteiligten Personen mit unterschiedlichen Arten von Online-Fragebögen und teilweise als Gruppeninterview statt. Zusätzlich zu den Evaluationen wurden auch Leistungen der Studierenden bei den Abgabepflichten im Modul ausgewertet. Dabei wurde der Fokus vor allem auf die Bestehensquoten der Testate und Meilensteine gelegt, da die Klausuren während der Umstellungsphase aufgrund von Corona unter deutlich veränderten Bedingungen stattfanden und der Vergleich daher nur in begrenztem Masse auf die Umgestaltung des Moduls zurückzuführen wäre.

4.1 Wöchentliche Fragebögen in Coding Classes

Im ersten Jahr wurde zur Einführung der Coding Classes eine wöchentliche Evaluation dieser durchgeführt. Dazu wurde jeweils fünf Minuten vor Ende jeder Coding Class ein QR-Code zur Verfügung gestellt, der zu einem kurzem Online-Fragebogen mit 12 geschlossenen Items, die in jedem Fragebogen vorhanden waren, zwei themenbezogenen geschlossenen Items sowie zwei offenen Fragen führte. Neben demografischen Angaben wurden Items zur Vorbereitung, Themenauswahl (z. B. «Zu welchen Thematiken hätten Sie sich mehr (Anwendungs-)Aufgaben gewünscht?»), Einschätzung zur Menge des Gelernten, Gründe für den Besuch sowie die Zufriedenheit jeweils auf Likert-Skalen erhoben. Dadurch konnten pseudonymisiert auch Verläufe einzelner Studierender z. B. bezüglich der Zufriedenheit, nachvollzogen werden. Neben den geschlossenen Fragen boten offene Fragen (z. B. «Gibt es weitere Gründe, die den heutigen Besuch der Coding Class unterstützt haben?») die Gelegenheit auch qualitative Rückmeldungen zu geben. Zudem gaben sie den Lehrenden ein Feedback zur thematischen Aufbereitung einzelner Inhalte. Es konnten 390 unterschiedliche Pseudonyme während der 12 Wochen registriert werden, wobei die tatsächlichen Antwortzahlen pro Wochenfragebogen zwischen 31 in der zehnten Woche und 267 in der ersten Woche lagen.

4.2 Gesamtevaluation

Zum Ende der Vorlesungszeit in jedem Semester wurde eine umfassende Gesamtevaluation der Veranstaltung durchgeführt. Dabei wurden alle Teile und Aspekte der Veranstaltung sowohl mit geschlossenen als auch mit offenen Fragen zur Bewertung gestellt, um sowohl quantitative als auch qualitative Aussagen erfassen zu können. Themen waren dabei u. a. die Teilnahme an den Veranstaltungsteilen (Intervallskalen), der subjektive Lernerfolg (z. B. «Wie viel haben Sie wo gelernt?» mit Unteritems für die Veranstaltungsteile und Likert-Skala), die Verständlichkeit der Materialien und Veranstaltungen, einzelne eingesetzte didaktische Konzepte, Einschätzungen zur Schwierigkeit sowie die Bewertung des digitalen Konzepts im zweiten Jahr der Umstellung. Im Jahr vor der Umstellung hatten 104 Studierende die zu diesem Zeitpunkt kürzere Gesamtevaluation mit acht offenen sowie 61 geschlossenen Items vollständig ausgefüllt. Im ersten Jahr der Umgestaltung haben insgesamt 200 Studierende daran teilgenommen, wobei 124 den Fragebogen mit neun offenen und 84 geschlossenen Items komplett beantwortet haben. Im zweiten Jahr steigerte sich diese Zahl weiter auf 259 Antworten, von denen 204 vollständig waren. Weitere 55 Studierende füllten den Fragebogen nur teilweise aus. Dieser Fragebogen enthielt 13 offene und 120 geschlossene Items, wobei sich nicht alle Items explizit auf die Umgestaltung der Veranstaltung beziehen. Einige Formulierungen wurden aus Gründen der Vergleichbarkeit von der bisher existenten Evaluation aus dem Wintersemester vor den ersten Änderungen (Wintersemester 2018/19) übernommen. Ein Vergleich zu Durchgängen vor diesem ist jedoch aufgrund eines Wechsels in der betreuenden Professur nicht als sinnvoll zu erachten.

4.3 Befragung von Tutor:innen

Zusätzlich zur Betrachtung des Konzepts aus Studierendensicht und zur Analyse der Ergebnisse wurden auch systematisch die jährlich etwa 20 studentischen Tutor:innen befragt. Neben dem informellen wöchentlichen Austausch mit der Erfassung qualitativer Ereignisse und der Betrachtung der quantitativen Ergebnisse der Abgaben der Studierenden wurden dazu themengeleitete, teilstandardisierte schriftliche Gruppenbefragungen durchgeführt. Dazu wurden Themen vorgegeben, zu denen die Tutor:innen frei schreiben sollten. Weitere Tutor:innen konnten eine Aussage dann unterstützen. Nach dem ersten Jahr wurden ausserdem ausgewählte Tutor:innen verschiedener Erfahrungsstufen zu einem Gruppeninterview eingeladen, in dem die Aussagen weiter diskutiert wurden, um zur Weiterentwicklung des Konzepts beizutragen.

5. Ergebnisse

Aus den Ergebnissen der verschiedenen Evaluationsmethoden konnten verschiedene Erkenntnisse gewonnen werden, wobei die Erkenntnisse des ersten Jahres bereits in die Entwicklung des zweiten Durchgangs einfließen. Aus Gründen des Umfangs der Daten werden nur umgestaltungsbezogene Ergebnisse, insbesondere die Entwicklungen zwischen den unterschiedlichen Jahren dargestellt. Auch wird auf eine ausführliche Darstellung der Ergebnisse aus der Befragung der Tutor:innen im ersten Jahr verzichtet, da diese im Wesentlichen die Anpassungen in Coding Classes und Abgabebefragungen beinhalten. Die Endnoten der Studierenden werden ebenfalls nicht genauer betrachtet, diese wiesen über die drei betrachteten Durchgänge inklusive des Jahres vor der Umstrukturierung jedoch keine signifikanten Änderungen auf.

5.1 Ergebnisse zu Coding Classes

Aus der wöchentlichen Evaluation mit den Studierenden in den Coding Classes kann vor allem ein Verlauf bezüglich der Zufriedenheit und der subjektiven Einschätzung des Lernerfolgs betrachtet werden (siehe Abbildung 2). Dabei ist erkennbar, dass beide Itemwerte einen Anstieg aufweisen. Jedoch hatten die Coding Classes im ersten Jahr mit 52,94 % eine vergleichsweise hohe Rate an Studierenden, die im Mittel ab der gerundet 4. Coding Class (Mittelwert 3,56) nicht mehr teilgenommen haben. Die Auswertung der offenen Antworten zeigte Gründe sowohl im wahrgenommenen Zeitmangel als auch im vermuteten geringen Lerneffekt. Kritisiert wurde hier vor allem der Einstieg, bei dem zu Beginn strikt keine ausführlichen Wiederholungen zugelassen wurden, um die Vorbereitung nicht überflüssig erscheinen zu lassen. Die Befragung der Tutor:innen sowie die Selbsteinschätzungen in der wöchentlichen Evaluation zeigten jedoch, dass viele Studierende nicht ausreichend vorbereitet waren. Dies erschwerte den geplanten Ablauf der Coding Classes und bedingte aus Sicht der Tutor:innen auch die mangelnde Zufriedenheit einiger Studierender. Unter den Studierenden, die die Coding Class nicht oder kaum besucht haben, verschoben sich zum zweiten Jahr hin auch die Hauptgründe dafür. Im ersten Jahr gaben 44,15 % als Hauptgrund an, keinen Lerneffekt vermutet zu haben, und 15,58 % gaben an, grundsätzlich zu wenig Zeit zu haben. Im zweiten Jahr gaben 30,23 % derer, die nicht oder kaum an der Coding Class teilgenommen haben, an aufgrund der Vermutung eines ausbleibenden Lerneffekts so gehandelt zu haben. 27,78 % gaben an, grundsätzlich zu wenig Zeit gehabt zu haben.

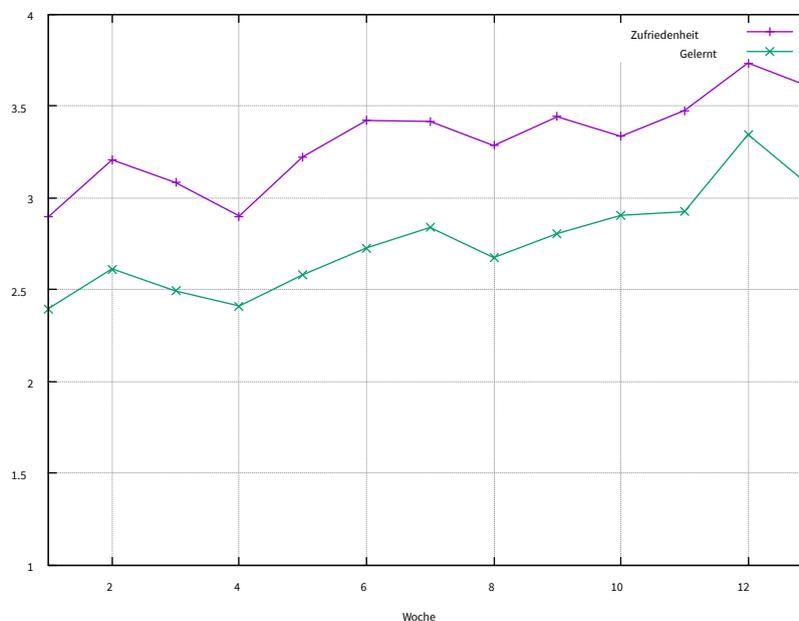


Abb. 2: Verlauf der Mittelwerte der Items «Zufriedenheit» und «Gelernt» in der wöchentlichen Coding-Class-Evaluation im Wintersemester 2019/20 (erstes Jahr der Umstellung), Skala von 1 für «gar nichts [gelernt]» bis 4 für «viel [gelernt]» bzw. 1 für «gar nicht [zufrieden]» bis 4 für «voll und ganz [zufrieden]», $n = 31$ (Woche 10) bis $n = 267$ (Woche 1).

Im zweiten angepassten Durchgang sank die Zahl derer, die ab einem bestimmten Termin auf den Besuch verzichteten, trotz digitaler Durchführung auf 44,77 % der Studierenden. Diese gaben an im Mittel ab der Coding Class 4,1 nicht mehr teilgenommen zu haben. Auch in den anderen Fragen der Evaluation konnte eine positive Entwicklung festgestellt werden. Bei der Frage, für wie sinnvoll die Studierenden das allgemeine Konzept der Coding Classes halten, sank der Mittelwert auf einer Skala von 1 für «sehr sinnvoll» bis 5 für «gar nicht sinnvoll» von 2,37 im ersten Jahr auf 1,77 im zweiten Jahr. Die Zustimmung zur Aussage «Die Anwesenheit in der Coding Class habe ich als vorteilhaft empfunden» weist einen Mittelwert von 3,01 auf, was auf der Skala von 1 für «trifft wenig zu» bis 4 für «trifft völlig zu» etwa der Antwort «trifft ziemlich zu» entspricht. Im Vorjahr lag dieser Wert bei 2,06 entsprechend etwa der Antwort «teils-teils».

5.2 Subjektiver Lernerfolg

Der subjektive Lernerfolg für die einzelnen Veranstaltungsteile wurde ebenfalls in der jährlichen Gesamtevaluation abgefragt. Die für die Umstellung relevanten Itemthemenbereiche sind in Tabelle 1 dargestellt. Dabei ist erkennbar, dass die Mittelwerte bereits im ersten Jahr leicht anstiegen, wobei nur bei den Coding Classes und Abgabeaufgaben tatsächliche Änderungen durchgeführt wurden. Der subjektive Lernerfolg bei den Coding Classes änderte sich zwischen den beiden Durchgängen nicht signifikant. Beim Vergleich zwischen Vorlesung und Lessons wiesen die Lessons einen stark gestiegenen Mittelwert auf, während der subjektive Lernerfolg des Livestreamings sich im Bereich der Mittelwerte von Vorlesung und Vorlesungsaufzeichnung einordnet. Auch bei den Abgabeaufgaben liegt ein starker signifikanter Anstieg vor.

Items Gelernt	Mittelwert vor Umstellung	Mittelwert 1. Jahr	Mittelwert 2. Jahr
Vorlesung	2,3	2,54	-
Vorlesungsaufzeichnung	2,38	2,65	-
Lessons	-	-	4,46
Livestreaming	-	-	2,76
Übung / Coding Class	2,82	3,31	3,27
Bearbeitung der Abgabeaufgaben	1,85	2,33	4,29 / 4,01 (Projekt)

Tab. 1: Mittelwerte der Items zur eigenen Lerneinschätzung in den Gesamtevaluationen zur Frage «Wie viel haben Sie bei folgenden Veranstaltungsteilen gelernt?» Skala von 1 – «gar nichts» bis 5 – «sehr viel», n = 104 / 124 / 204.

5.3 Ergebnisse zu Abgabeaufgaben und Prüfungszulassung

Bei den Abgabeaufgaben sind neben den Evaluationsergebnissen vor allem die tatsächlich erfolgten Zulassungen zur Klausur relevant, die durch erfolgreiches Absolvieren erlangt wurden. Dabei konnte bei den Meilensteinen im ersten Durchführungsjahr festgestellt werden, dass vor allem im zweiten Meilenstein mit 56 Studierenden ungewöhnlich viele Studierende keine oder keine ausreichende Lösung einreichten, sodass insgesamt 415 von 532 Studierenden, entsprechend 78,01 %, die Zulassung erlangten. Im Vorjahr mit klassischem Vorlesungs- und Übungskonzept lag die Zahl der Studierenden mit Zulassung bei 447 von 518, entsprechend 86,29 %. In der zweiten Durchführung mit dynamischen Zeiträumen konnten 459 von 510 Studierenden, entsprechend 90 %, die Zulassung erreichen. Bezüglich der dynamischen Zeiträume und der Ausweitung der Gruppengröße auf drei Studierende äusserten sich die Tutor:innen besonders zufrieden und positiv. Die Aufteilung der Aufgaben auf drei Personen sei dabei einfach möglich und das Monitoring durch die wachsenden Zeiträume passend gewesen. Die zwei- und dreiwöchentlichen Abgaberrhythmen wurden

mit Mittelwerten von 2,85 bzw. 2,94 auf einer Skala von 1 für «sehr stressig» bis 5 für «sehr entspannt» als angemessener empfunden als im wöchentlichen Rhythmus (Mittelwert 2,07). Ein Kritikpunkt der grundsätzlich von den Studierenden positiv empfundenen Bonuspunkte im Projekt war jedoch, dass diese nur subjektiv verteilt werden konnten, da die Abwägung zwischen Aufwand und Ergebnis nicht eindeutig war. Grundsätzlich wäre das Projekt gut für das Aufbauen eines Gesamtkonstrukts beim Programmieren im Sinne einer Verknüpfung der bislang erlernten Themen gewesen, jedoch hätten die Programme wenig neues Wissen über Programmier-elemente aufgewiesen. Auch blieben so die algorithmischen Themen der letzten Wochen nicht kontrolliert.

5.4 Weitere Ergebnisse

Die Lessons wurden sowohl von Studierenden als auch von Tutor:innen als gewinnbringend empfunden. Durch die Verlinkung optionaler weiterer Notebooks, die beliebig intensiv nutzbaren Codebeispiele sowie die grundsätzlichen Möglichkeiten beim Lerntempo und Lernzeitpunkt bieten die Lessons im Vergleich zur Vorlesung eine hohe Adaptivität. Allerdings wurde auch von teilweise demotivierender Wirkung bei auftretenden Performanceproblemen der Jupyter Notebooks berichtet. Auch wurde der Zeitaufwand für die Lessons kritisiert. Dabei gaben 19,12 % der Studierenden an durchschnittlich über 180 Minuten pro Woche für die Bearbeitung der Lessons aufgewendet zu haben.

Situationsbedingt lagen die Materialien für die letzten zwei Wochen der Vorlesungszeit als Vorlesungsaufzeichnung vor, sodass die Studierenden auch einen direkten Vergleich zwischen den Lessons und Vorlesungsaufzeichnungen ziehen konnten. Dabei gaben sie in der Evaluation mit einem Mittelwert von 1,97 auf einer Skala von 1 für «Lessons viel besser» bis 5 für «Vorlesungsaufzeichnungen viel besser» an, dass der allgemeine Eindruck bei den Lessons «etwas besser» war als bei den Vorlesungsaufzeichnungen. Bezüglich der anderen abgefragten Eigenschaften werden die Lessons tendenziell besser eingeschätzt, wobei Mittelwerte zwischen 2,11 für die Motivation und 2,18 für den Lernerfolg bis 2,64 bzw. 2,73 für den zeitlichen bzw. inhaltlichen Anspruch auftraten. Tabelle 2 zeigt eine Übersicht der Mittelwerte für die Vergleichskategorien von Lessons und Vorlesungsaufzeichnungen.

Allgemeiner Eindruck	1,97 (2 = «Lessons etwas besser»)
Motivation	2,11 (2 = «Lessons etwas besser»)
Lernerfolg	2,18 (2 = «Lessons etwas besser»)
Durchhaltevermögen	2,28 (2 = «Lessons etwas besser»)
Verständlichkeit	2,51 (3 = «beide gleich gut»)
Zeitlicher Anspruch	2,64 (3 = «beide gleich gut»)
Inhaltlicher Anspruch	2,73 (3 = «beide gleich gut»)

Tab. 2: Mittelwerte der Items zum Vergleich von Lessons und Vorlesungsaufzeichnung in der Gesamtevaluation 2020/21. Skala von 1 – «Lessons viel besser» bis 5 – «Vorlesungsaufzeichnungen viel besser», n = 204.

6. Diskussion

Zunächst kann festgestellt werden, dass die grosse Anzahl an Veränderungen insgesamt zu einem für die aktuellen Bedingungen besseren System geführt hat, dessen Auswirkungen aufgrund ihrer Komplexität und der veränderten Ausgangsbedingungen allerdings noch nicht vollständig bewertbar sind. Aufgrund der veränderten Prüfungsmodalitäten wegen der Coronapandemie und der grundsätzlich erschwerten Lehre während dieser Zeit können die im Mittel etwa gleichbleibenden Prüfungsergebnisse der Studierenden in der Prüfung als Erfolg gedeutet werden. Die digitalen Open-Books-Klausuren enthielten nahezu keine Reproduktionsaufgaben mehr und sind dadurch vom Niveau tendenziell als schwieriger einzustufen.

Aus dem Anstieg der Zufriedenheit und des subjektiven Lernerfolgs während des ersten Jahres kann vermutet werden, dass Lehrende wie auch Studierende sich mit der Zeit stärker auf das Konzept eingelassen haben. Allerdings haben tendenziell unzufriedene Studierende auf den weiteren Besuch von Coding Classes verzichtet und damit auch nicht mehr an den Befragungen teilgenommen, was einen möglichen Bias zur Folge hat. Durch die Anpassungen beim Lernmaterial und die Integration der wesentlichen Vorbereitung auf die Coding Classes in das zentrale Lernmaterial als Umsetzung eines vollständigen Flipped Classrooms wurde die als problematisch empfundene zusätzliche Vorbereitung jedoch deutlich abgemildert. Dies wurde mutmasslich auch durch die bessere Anpassung der Coding Class Aufgaben an die Abgabearbeiten erreicht. Der für einige Studierende als erhöht empfundene Zeitaufwand für die Bearbeitung der Lessons beruht zum Teil vermutlich auf der Vorstellung, dass die Lessons exakt 90 Minuten Anwesenheit in der Vorlesung ersetzen sollen. Obwohl bei der konkreten Frage nach der aufgewendeten Zeit nur etwa ein Fünftel der Studierenden angab, mehr Zeit aufgewendet zu haben, findet den Beobachtungen der Lehrenden nach eine bessere Nachbereitung der theoretischen Inhalte statt. Um das Gefühl, dass zu viel Zeit investiert werden muss, abzumildern, könnte den

Studierenden konkreter erläutert werden, wie viel Lernzeit in den Leistungspunkten eingeplant ist und dass die reine Anwesenheit in einer Vorlesung die Stoffvermittlung nicht vollständig abdecken kann.

Die deutlich geringere Quote an Studierenden, die die Prüfungszulassung durch die Meilensteine erhalten haben, ist zwar auch stark auf einen bestimmten Meilenstein zurückzuführen, zeigt jedoch, dass schwierige Aufgaben bei längeren Abgabezeiträumen zu Beginn schnell überfordernd wirken können. Aus diesem Grund wurden im zweiten Jahr die wachsenden Abgabezeiträume als besseres Konzept erprobt. Die Gruppengröße von drei Studierenden für die Abgabeaufgaben hat sich als tragfähig erwiesen und so eine Reduktion des Personalaufwands ohne Verzicht auf engmaschige Betreuung ermöglicht. Eine weitere Erhöhung der Gruppengröße ist unter Beibehaltung der Art der Betreuung jedoch vermutlich nicht sinnvoll. Die dynamischen und generell erhöhten Zeiträume für die Abgaben sorgen für Entlastung auf beiden Seiten – sowohl bei den studentischen Tutor:innen aufgrund der geringeren Anzahl an Testaten als auch bei den Studierenden, die über einen längeren Zeitraum differenziertere Lösungen entwickeln können.

Durch eine Steigerung der subjektiven Lernraten und eine Konzentration des Lernens auf die Lessons als Stoffvermittlung sowie die Aufgabenbearbeitung als Anwendung stehen nun stärker die dafür vorgesehenen Teile der Veranstaltung im Vordergrund.

Aufgrund der gewachsenen Komplexität des aktuellen Konzepts bleibt insbesondere im Hinblick auf die Beobachtung, dass der Zweck der Beratungszeit häufig missverstanden wurde, zu überdenken, ob weitere Darstellungen der Kursbestandteile wie beispielsweise ein Glossar sinnvoll sein könnten. Möglicherweise könnte den Studierenden ein Glossar über die verwendeten Begriffe oder eine Sammlung häufiger Fragen zur Verfügung gestellt werden, um die Suche nach relevanten Informationen zu vereinfachen.

In Bezug auf die Berücksichtigung der Diversität hat die Digitalisierung insbesondere durch die asynchronen Lernmaterialien erhebliche Fortschritte gemacht. Eine flexible Zeiteinteilung, das Lernen im individuellen Tempo und optionale Inhalte zur Ergänzung ermöglichten es Lernenden mit diversen Vorkenntnisstufen und unterschiedlichen Lebenssituationen die Inhalte passend zu ihrem Zeitplan zu erarbeiten. Die technisch einfach mögliche Einblendung von Untertiteln in den vertonten Folien bietet ausserdem eine Hilfestellung bei sprachlichen Schwierigkeiten und für hörgeschädigte Studierende.

7. Zusammenfassung und Ausblick

Im Rahmen der Umgestaltung des Einführungsmoduls der Informatik hat sich gezeigt, wie Diversität und Digitalisierung als gegenseitige Katalysatoren wirken. Geleitet durch den Anspruch der Heterogenität der Studierenden möglichst gut gerecht zu werden und für diverse Voraussetzungen passende Lerngelegenheiten zu schaffen, wurden verstärkt digitale Werkzeuge genutzt, deren Adaptivität sich als vorteilhaft erwiesen hat. Umgekehrt wurden durch die Pandemie stärker digitale Konzepte und Fernlehre erzwungen, was die weitere Entwicklung zusätzlich fördert. Dabei wurden auch die verbliebenen klassischen Teile des Moduls in ein digitales Konzept eingebettet und boten so weitere Gelegenheiten für adaptives Lernen.

Insbesondere konnte die bislang häufig fehlende Vor- und Nachbereitung der Vorlesungsinhalte durch die Aufbereitung in den Lessons als Jupyter Notebooks einfacher stattfinden. Dies löste zum Teil die Problematik der geringen Vorbereitung auf die Coding Classes, wobei dies auch durch einen ausgedehnteren Einstieg im zweiten Jahr abgemildert werden konnte. Dynamische Zeiträume für Abgabenaufgaben ermöglichten eine engmaschige Kontrolle des Lernerfolgs zu Beginn, während ein Projekt am Ende realitätsnäher ganzheitliche Programmierkompetenzen förderte und motivationsfördernd einen kleinen Teil der Endnote bestimmte. Zentrale Ergebnisse der Begleitforschung sind daher zum einen die Notwendigkeit Materialien nicht speziell für den Kontaktteil eines Flipped Classrooms bereit zu stellen, sondern diese im Gesamtkontext als zentrales Lernmaterial zu integrieren. Zum anderen haben sich offene Aufgaben zwar grundsätzlich wie auch in anderen Projekten als motivationsfördernd gezeigt, jedoch ist insbesondere bei Programmieranfänger:innen eine kontinuierliche Kontrolle der Mitarbeit sinnvoll, um eine Überforderung durch im Verlauf ansteigende Anforderungen zu vermeiden. Eindeutige, auf Punkten basierte Bewertungsschemata geben Studierenden sowie Tutor:innen dabei Orientierung die Leistungen einzuschätzen.

Insgesamt ist damit ein Konzept gelungen, das im digitalen Semester von den Studierenden positiv bewertet wurde. Bei einer Übertragung in zukünftige Semester wird weiterhin ein starker Fokus auf den digitalen Konzepten liegen, wobei insbesondere die Coding Classes auch von einer Durchführung in Präsenz weiter profitieren könnten.

Literatur

- Bergmann, Jonathan, und Aaron Sams. 2012. *Flip Your Classroom: Reach Every Student in Every Class Every Day*. International Society for Technology in Education.
- Bruce, Kim B., Andrea Danyluk, und Thomas Murtagh. 2001. «A Library to Support a Graphics-Based Object-First Approach to CS 1». *SIGCSE Bull* 33 (1). <https://doi.org/10.1145/366413.364527>.

- Campbell, Jennifer, Diane Horton, Michelle Craig, und Paul Gries. 2014. «Evaluating an Inverted CS1». In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education (Atlanta, Georgia, USA) (SIGCSE '14)*. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/2538862.2538943>.
- Gudjons, Herbert. 2001. *Handlungsorientiert lehren und lernen*. Bad Heilbrunn: Klinkhardt.
- Hakimzadeh, Hossein, Raman Adaikkalavan, und Robert P. Batzinger. 2011. «Successful Implementation of an Active Learning Laboratory in Computer Science». In *Proceedings of the 39th Annual ACM SIGUCCS Conference on User Services (San Diego, California, USA) (SIGUCCS '11)*. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/2070364.2070386>.
- Hodges, Mark 2019. «Flipping One Day Each Week in a Smaller CS1 Course: An Experience Report». *Journal of Computing Sciences in Colleges* 34 (7): 20–27.
- Horton, Diane, Michelle Craig, Jennifer Campbell, Paul. Gries, und D. Zingaro. 2014. «Comparing Outcomes in Inverted and Traditional CS1». In *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education (Uppsala, Sweden) (ITiCSE '14)*. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/2591708.2591752>.
- Keen, Aaron, und Kurt Mammen. 2015. «Program Decomposition and Complexity in CS1». In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education (Kansas City, Missouri, USA) (SIGCSE '15)*. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/2676723.2677219>.
- Knoll, Michael. 1997. «The project method: Its vocational education origin and international development». *Journal of Industrial Teacher Education* 34: 59–80.
- Kolmos, Anette. 1996. «Reflections on Project Work and Problem-based Learning». *European Journal of Engineering Education* 21(2), 141–148. <https://doi.org/10.1080/03043799608923397>.
- Köppe, Christian, Ralph Niels, Rene Bakker, und Stijn Hoppenbrouwers. 2016. «Flipped Classroom Patterns: Controlling the Pace». In *Proceedings of the 10th Travelling Conference on Pattern Languages of Programs (Leerdam, AA, Netherlands) (VikingPLoP '16)*. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/3022636.3022637>.
- Köppe, Christian, Ralph Niels, Robert Holwerda, Lars Tijsma, Niek Van Diepen, Koen Van Turnhout, und Rene Bakker. 2015. «Flipped Classroom Patterns: Designing Valuable in-Class Meetings». In *Proceedings of the 20th European Conference on Pattern Languages of Programs (Kaufbeuren, Germany) (Euro-PLoP '15)*. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/2855321.2855348>.
- Kussmaul, Clif. 2008. «Scaffolding for Multiple Assignment Projects in CS1 and CS2». In *Companion to the 23rd ACM SIGPLAN Conference on Object-Oriented Programming Systems Languages and Applications (Nashville, TN, USA) (OOPSLA Companion '08)*. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/1449814.1449890>.
- Lage, Maureen, Glenn Platt, und Michael Treglia. 2000. «Inverting the Classroom: A Gateway to Creating an Inclusive Learning Environment». *The Journal of Economic Education* 31 (1), 30–43. <https://doi.org/10.1080/00220480009596759>.

- Largent, David L. 2013. «Flipping a Large CS0 Course: An Experience Report about Exploring the Use of Video, Clickers and Active Learning». *J. Comput. Sci. Coll.* 29 (1).
- Latulipe, Celine, N. Bruce Long, und Carlos E. Seminario. 2015. «Structuring Flipped Classes with Lightweight Teams and Gamification». In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education (Kansas City, Missouri, USA) (SIGCSE '15)*. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/2676723.2677240>.
- Latulipe, Celine, Audrey Rorrer, und Bruce Long. 2018. «Longitudinal Data on Flipped Class Effects on Performance in CS1 and Retention after CS1». In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education (Baltimore, Maryland, USA) (SIGCSE '18)*. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/3159450.3159518>.
- Lockwood, Kate, und Rachel Esselstein. 2013. «The Inverted Classroom and the CS Curriculum». In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education (Denver, Colorado, USA) (SIGCSE '13)*. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/2445196.2445236>.
- Matzko, Sarah, und Timothy Davis. 2006. «Using Graphics Research to Teach Freshman Computer Science». In *ACM SIGGRAPH 2006 Educators Program (Boston, Massachusetts) (SIGGRAPH '06)*. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/1179295.1179305>.
- Mohamed, Abdallah. 2020. «Evaluating the Effectiveness of Flipped Teaching in a Mixed-Ability CS1 Course». In *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education (Trondheim, Norway) (ITICSE '20)*. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/3341525.3387395>.
- Ossovski, Elisaweta, Daniel Kalbreyer, Laura Hembrock, und Michael Brinkmeier. 2021. «Co-operative Gamification in a Computer Science Introductory Module». In *CSERC '21: Proceedings of the 10th Computer Science Education Research Conference (Virtual Event, Netherlands)*. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/3507923.3507953>.
- Razak, Saquib. 2013. «A Case for Course Capstone Projects in CS1». In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education (Denver, Colorado, USA) (SIGCSE '13)*. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/2445196.2445398>.
- Rosiene, Carolyn Pe, und Joel A. Rosiene. 2019. «To Flip or Not to Flip: Experiences with a Hybrid Approach». In 2019 IEEE Frontiers in Education Conference (FIE), Covington, KY, USA. <https://doi.org/10.1109/FIE43999.2019.9028540>.
- Sarawagi, Namita. 2014. «A Flipped CS0 Classroom: Applying Bloom's Taxonomy to Algorithmic Thinking». *Journal of Computing Sciences in Colleges* 29 (6).
- Schulmeister, Rolf, und Christiane Metzger. 2011. *Der Workload im Bachelor. Zeitbudget und Studierverhalten. Eine empirische Studie*. Münster: Waxmann.

-
- Schwarz, Richard, Lutz Hellmig, und Steffen Friedrich. 2021. «Informatikunterricht in Deutschland – eine Übersicht». *Informatik Spektrum* 44: 95–103. <https://doi.org/10.1007/s00287-021-01349-9>.
- Tarimo, William T., Fatima Abu Deeb, und Timothy J. Hickey. 2016. «A Flipped Classroom with and Without Computers». In *Computer Supported Education*, herausgegeben von S. Zvacek, Maria Teresa Restivo, James Uhomoibi, und Markus Helfert, 333-347. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-29585-5_19.
- Wicentowski, Richard, und Tia Newhall. 2005. «Using Image Processing Projects to Teach CS1 Topics». *SIGCSE Bull* 37 (1): 287–291. <https://doi.org/10.1145/1047124.1047445>.

Förderhinweis

Die Autorin *Elisaweta Ossovski* ist Stipendiatin des Ernst Ludwig Ehrlich Studienwerks.

A.2 Beitrag 2

Übersicht zu Beitrag 2

Titel: Cooperative Gamification in a Computer Science Introductory Module

Autoren: Elisaweta Ossovski
Daniel Kalbreyer
Laura Hembrock
Michael Brinkmeier

Jahr: 2021

Status: veröffentlicht

Bibliographische Informationen: Ossovski, Elisaweta; Kalbreyer, Daniel; Hembrock, Laura; Brinkmeier, Michael: Cooperative Gamification in a Computer Science Introductory Module. In: Proceedings of the 10th Computer Science Education Research Conference. New York, NY, USA: Association for Computing Machinery, 2021 (CSERC '21). – ISBN 9781450385763. <http://dx.doi.org/10.1145/3507923.3507953>. S. 70–78

Link zur Veröffentlichung: <https://dl.acm.org/doi/10.1145/3507923.3507953>

Cooperative Gamification in a Computer Science Introductory Module

Elisaweta Ossovski*
 eossovski@uos.de
 Institute for Computer Science
 Universität Osnabrück, Germany

Laura Hembrock
 lhembrock@uos.de
 Institute for Computer Science
 Universität Osnabrück, Germany

Daniel Kalbreyer
 dakalbreyer@uos.de
 Institute for Computer Science
 Universität Osnabrück, Germany

Michael Brinkmeier
 mbrinkmeier@uos.de
 Institute for Computer Science
 Universität Osnabrück, Germany

ABSTRACT

Especially in introductory modules, that are predominantly attended by first-year students, it is of particular importance to set incentives for active participation, to develop a community feeling and to promote cooperative behaviour by the students. The online distance learning necessitated by the Covid19-pandemic has further intensified this challenge. In order to encourage the desired activity on the part of the students and to build up a sense of belonging, a gamification approach was tried out in a large computer science introductory module with about 550 participants. This involved collecting so-called "Algos" together by all course participants through various activities, with advantages granted to all students for reaching a certain amount of Algos together. In combination with humorous elements and contexts (e.g. "Easter eggs" and word problems), the gamification system led to a high level of satisfaction among the students. Although the majority of the students did not perceive the Algos as the main motivation for most of the concrete activities, the lecturers were also satisfied with the actual participation in the digital semester.

CCS CONCEPTS

• **Social and professional topics** → **Computer science education; CS1**; • **Applied computing** → **Distance learning**; Collaborative learning.

KEYWORDS

Gamification, CS1, Cooperative, Easter eggs, Word problem

ACM Reference Format:

Elisaweta Ossovski, Daniel Kalbreyer, Laura Hembrock, and Michael Brinkmeier. 2021. Cooperative Gamification in a Computer Science Introductory Module. In *The 10th Computer Science Education Research Conference (CSERC '21)*.

*The author is an ELES Research Fellow.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CSERC '21, November 22–23, 2021, Virtual Event, Netherlands

© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-8576-3/21/11... \$15.00

<https://doi.org/10.1145/3507923.3507953>

November 22–23, 2021, Virtual Event, Netherlands. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3507923.3507953>

1 INTRODUCTION

When introductory modules are taken by students from different study programmes, this often leads to different goals and correspondingly different motivation among the students. In computer science, the challenge of dealing with this heterogeneity is further intensified by different levels of prior knowledge, as computer science can currently only be taken as an elective subject at German schools. Besides this, some students also already have programming knowledge from an vocational training or their hobbies. The Covid19-pandemic and the related often completely digital teaching increases difficulties in motivation and active cooperation between students. As part of the restructuring of the module, a revised flipped classroom concept with interactive learning materials and online conferencing for questions and exercises were used in a large introductory computer science module (CS1). In order to promote active participation, a gamification approach was additionally implemented. Students were supposed to collectively unlock rewards by getting "Algos" through desired behaviour, from which all students could in turn benefit, when all participants reached a certain score together. At the same time, however, as in the previous concept, personal preliminary work for the final grade should remain possible to a small extent. The evaluation of the concept focused on the following research questions:

- (Q1) How do the implemented, cooperative gamification elements influence the satisfaction and motivation of the students in the scenario with online distance learning?
- (Q2) How can a realistic distribution of scores be designed in a course-wide cooperative gamification concept?

2 THEORETICAL BACKGROUND

Deterding et. al. [2] define gamification as "the use [...] of design [...] elements [...] characteristic for games [...] in non-game contexts" ([2, p.13]). In contrast, games with an intended learning effect, so-called "serious games", are complete systems. In the other dimension, the target nature is of relevance, in order to differentiate "playing" from "gaming". In the context of motivational psychology, there are also findings that provide a theoretical basis for the use and success of gamification. Thus, appropriate expectations and personal interest encourage the motivation of students [6]. Popular elements

CSERC '21, November 22–23, 2021, Virtual Event, Netherlands

Ossovski, Kalbreyer, Hembrock and Brinkmeier

of gamification are scores awarded for certain behaviour that lead to rewards. It is therefore important to choose the required scores for rewards as achievable as possible, while they should still be perceived as rewards for a certain amount of work. In addition, personal interest can possibly also be generated through the design of the learning material, if the interest in the content itself appears to be limited.

2.1 Previous research

Elements of gamification have already been used in some higher education scenarios and have also been tested in undergraduate computer science courses with mostly positive results.

Dicheva, Irwin and Dichev [3] describe a gamification concept for self-learning activities in a small data structure course compared to a previous course. On a learning platform, points could be collected individually for the successful completion of learning activities outside of compulsory assignments and used for self-selected rewards in the form of assessment or time advantages. This concept led to lower failure rates and good acceptance within the few students.

In a larger CS1 course, Latulipe, Long and Seminario [5] combined collaboration in fixed groups in the classroom part of a flipped classroom with awards for successful participation, with 10 awards each improving the final grade by 1 %. In addition to general satisfaction, this led to better networking among students. Building on this research, Sprint and Fox [7] developed a CS1 course that was tested in different student groups, each with 21 to 44 participants and different combinations for the use of fixed groups as well as gamification elements. In the process, especially the timing behaviour of the students was rewarded by assigning scores for submitting programming solutions early, but also for questions and answers in the forum. As a reward, the scores of selected questions in the final exam were multiplied for the best teams.

In a related subject, Weyers [8] applied a gamification approach in a basic statistics course with a flipped classroom concept, which allowed both individual students and groups to score points that resulted in a direct increase in the final grade as well as advantages for the exam design as collectively collected points. This made it possible to achieve a good level of participation and a high level of interest on the part of the students in the attendance events. However, the collective collection of points was rated better than the individual competition.

To take a broader perspective, Hamari and Koivisto [4] examined various social factors as well as recognition value as influences on the intention to use gamification on the basis of an app whose users set themselves sporting goals and are rewarded for this with badges. The authors conclude that social factors also have an important role in the use of gamification, so that mutual reinforcement takes place in a person's communication circle. Even though this study was conducted outside of higher education, this finding can be helpful for a cooperative model.

In addition to the fully collaborative nature of the gamification concept presented here, the overall context also differs from previous studies on this topic. Although some experience has been gained in flipped classroom scenarios with learning opportunities at home and exercises in physical presence, the module described in this paper took place entirely digitally with asynchronous learning

opportunities and synchronous exercises. These conditions, as well as the fact that the module is predominantly taken by first-year students, made group formation and communication difficult.

3 CONDITIONS AND METHODOLOGY

The gamification concept was implemented in an introductory computer science module, which corresponds to a rather extensive CS1 in terms of content and also addresses algorithms and data structures in addition to an introduction to programming. About 550 students from different study programmes such as Computer Science, Cognitive Science, Mathematics (including teacher training), Business- and Geo-Informatics as well as (Environmental) System Sciences take part in this annual module. In the described semester 412 students took the first exam and 166 took the second with part of the students in the second exam repeating it.

The digital implementation of the module during the Corona pandemic was part of a project to generally restructure the teaching in this module. In the previous year, "coding classes" were introduced as an active form of exercise, which replaced a formerly rather passive and lecture-like exercise. Special tasks were provided to prepare for the submission tasks¹ and to promote the cooperative work of the students with simultaneous supervision by lecturers and student tutors.

While the lecture was still largely classical at that time, the digital learning scenario led to a complete changeover to the flipped classroom concept. The learning material was made available on a Jupyter notebook server, where slideshows with explanatory texts and executable code cells were linked to visual elements for testing and practice. Both the learning material itself and the tasks for coding classes and submissions were made available and processed via the JupyterHub. The asynchronous learning material was supplemented by weekly interactive livestreaming to clarify questions. In addition, the submission tasks were worked on weekly at the beginning as assignment sheets and three-weekly at the end as a programming project in groups of up to three students and discussed online with student tutors, with each student having to explain a previously unknown assigned part.

The gamification concept, like the other organisational aspects of the event, was explained on the second day of the course, after a thematic introduction on the first day, in an online conference. However, the list of scores to be earned and the achievable goals (see tables 1 and 2, the latter without the scores) were published in the learning management system after about two weeks.

To answer the first research question, in the final evaluation of the course various questions were asked, as experience has shown that this reaches the largest possible number of students best. The evaluation started in the last week of the lecture period, was open for 10 days and was advertised by email and in the learning management system. Within the gamification concept, participation itself was rewarded. 204 students completed the evaluation in full. This is a relatively high participation rate compared to previous final evaluations and especially to other questionnaires during the module.

¹In order to achieve the exam admission, it was necessary to regularly work on and submit tasks.

The questions related to the gamification concept included both closed questions with Likert scale response options and open questions where students could describe their impressions on specific aspects or in general. For the general questions, a five-point Likert scale was used in order to also be able to represent a reasonable neutral attitude towards the concepts. In order to map the perceived motivation by the Algos to behave in a certain way, a three-point Likert scale was chosen, as motivation is only assumed in one direction and thus the negative part of a possible five-point scale is omitted. For the second research question, the development of the points awarded, the proportions by category as well as the goals achieved and their scores were analysed.

4 GAMIFICATION CONCEPT

For the introductory computer science module described above, a gamification approach was developed in which students collectively earn scores, so-called "Algos", for desired behaviour and also collectively receive rewards for reaching certain Algo numbers together. In addition, humorous insertions, so-called "eastereggs", as well as work problems, were increasingly used in previous implementations to loosen up and ensure authenticity on the part of the lecturers. Within the JupyterHub used for the materials and as a development environment, a counter was set up for the scores in the gamification concept as well as a message display through which entries could be communicated and explained (see figure 1).

4.1 Possibilities to gain Algos

The Algos were intended to reward active participation and cooperative behaviour also towards other students. Initially, it was decided to only award Algos in large amounts, with the smallest unit awarded being 100 Algos. For each category, a minimum number of Algos was defined to be granted. Beyond that, however, the lecturer or tutor who made the entry was allowed to decide on the exact number of Algos as well as the concrete interpretation of the criteria. Both the responsible teachers, specifically the professor and the research assistants, and the student tutors were encouraged to assign Algos. An overview of the possibilities for gaining Algos is shown in table 1.

These minimum numbers of Algos for each behaviour were chosen based on sample calculations of differently active weeks, also taking into account the corresponding efforts. During the entry process, the lecturers could still decide individually for each entry to make some adjustments in order to suit the situation as best as possible. However, due to the minimum number of Algos assigned for a category, the students had a guarantee that they would receive a certain number of Algos for a certain activity in any case.

High activity in the synchronous online conferences (mainly Coding Classes) or in the chat complementing the Livestreaming were rated as the most important behaviours to be awarded with Algos. After some consideration based on the frequency of the events per week, a ratio of 1 to 5 was determined for the Algos for the basic activity in Coding Classes and Livestreaming. Thus, in 10 Coding Classes per week with 2,000 Algos each, theoretically twice as many Algos could be achieved for the activity alone as through Livestreaming with 10,000 Algos. By also assigning a high score to the presentation (and uploading) of solutions or solution

approaches, even more Algos could be achieved through the Coding Classes.

This proved to be useful in the further course, since a high level of activity in the Coding Classes is fundamentally responsible for the learning effect and the willingness to ask questions was fundamentally higher in Livestreaming. For all synchronous dates, it was possible to receive 1.5 times the amount of Algos for a very high level of activity. Forum contributions represent a further component of the students' activity. In order to avoid a disproportionate valuation due to the possible high number, rather low numbers of Algos were assigned here for helpful and meaningful contributions. Good answers could achieve at least 500 Algos, five times the number of Algos compared to good questions with at least 100 Algos.

Apart from these activities, efforts and products that go beyond the requirements also should be rewarded. If students shared additional material of good quality, tutorials for further functions of the programming tool or other interesting contributions, this was rewarded in a similar way as an uploaded solution to the coding class.

In addition, it was valued if students did not write emails to the personal email addresses of the lecturers, but used a module-related email address that was provided and constantly communicated. The background to this is that often basic questions were sent to several lecturers, which considerably increased the organisational effort due to coordination and consultation. All enquiries concerning the module that were not particularly specialised should therefore be sent to the module-related e-mail address to which the professor and the research assistants had access. The students were not prevented from asking questions in principle, but were merely requested to use this form for simplified organisation.

Another possibility to get Algos was participation in studies on aspects of teaching as well as evaluations.

There were no individual Algo scores and the allocation of individual students or behaviours to acquired Algos was only possible via the messages that Algos were awarded for a particular Coding Class or a completed questionnaire. This limited the possible competitive character of a gamification concept, but in principle there also exist cooperative games. The concept aimed at supporting the general activity as well as cooperative behaviour between students as all students could benefit of shared ideas concerning questions and solutions and could get advices in the forum.

4.2 Rewards for collected Algos

The rewards for collected Algos were called "stretch goals" in the style of crowdfunding projects. At the beginning, the exact order and the required Algos were left open. It was explained that in principle up to 5 bonus points can be unlocked, but these still have to be achieved individually through creative extensions as part of the final submission task in small groups like in previous semesters. The individually achieved bonus points were added to the exam points to form a final grade, with 5 bonus points representing approximately one partial grade (e.g. the improvement from 2.7 to 2.3 corresponding to C+ to B-).

In addition, there were two targets concerning the submission tasks and their corresponding audits, which allow a single fail



Figure 1: Counter and test message in the Jupyter Notebooks Hub

Behaviour	(minimum) amount of Algos
Good question/answer in forum	100 / 500
Upload of a solution (attempt) for a Coding Class task	1,000
Explanation of a solution (attempt) in a Coding Class	5,000
Additional material/tutorial of at least good quality	1,000 / 2,000
Submission of assignment proposals (e.g. for the exam)	200
(Very) High activity in Coding Classes	2,000 (3,000)
(Very) High activity in Livestreaming	10,000 (15,000)
Fully completed questionnaire/participation in interview	200 / 2,000
No mails to personal addresses (only module address) for one week	10,000

Table 1: Amount of Algos rewarded for desired behaviours

except for the last submission (only $n-1$ of the task sheets before the project had to be passed), as well as a choice of the explaining person for one task each in the test. This was realized by the possibility to swap the explaining person for one task. An entertaining Christmas lecture was unlocked as a non-module-relevant stretch goal. The remaining stretch goals related to the written exam. Here, three times 10 minutes additional exam time could be unlocked, whereby these could be justified with increased technical requirements in the digital exam within the framework of the examination regulations. In addition, simulations of such exams were made possible and further sample exams were planned.

The scores needed for reaching a stretch goal were developed during the lecture period due to the lack of experience so far. At first the initial calculation was kept that with 75,000 Algos per stretch goal, on average a little more than one stretch goal would be unlocked per week. Towards the middle of the semester, it became clear that the initially high participation could not be reached constantly, so that the number was lowered to 50,000 Algos for each next stretch goal, especially for the bonus points to match the time of the last submission. An overview of the final order and the Algos required for each stretch goal is shown in table 2.

4.3 Eastereggs and word problem tasks

Eastereggs are humorous inserts that are sometimes hidden in learning materials and other components of the course and often link to memes, videos, etc. This is intended to increase motivation due to the presentation, as described in section 2. In the described module, on exercise sheets, a link was regularly included in a line that was existing anyway due to the layout. In the learning material, where the classical lecture (recording) was replaced by Jupyter

notebook lessons consisting of slides with audio and markdown and code cells for practising, the Eastereggs were harder to find. An example of an Easteregg from a learning material is shown in Figure 2. Often the links led to music to create short pauses in learning. In a learning material where a visualized raindrop list was to be programmed, each occurrence of "rain" and "drops" was linked to a thematically appropriate song. Tasks within the learning materials were mostly set in such a way that solutions and other outputs unlocked Eastereggs. In addition, an attempt was made, especially in the programming tasks in the Coding Classes and for the submissions, to use word problems with reference to pop culture and narrative contexts from the youth and childhood of the target group. An example is shown here in Figure 3. In this case, the implementation of an internally chained list of circles was set in the context of the cartoon series "The Secret World of Santa Claus" in order to achieve more motivation to match the Christmas holidays.

5 EVALUATION

The effects of the Algos were monitored in particular as a new concept and also specifically evaluated in the final evaluation of the event. The overall opinion was predominantly positive.

5.1 Development of the Algo score

A total of 133 Algo entries were counted in the database, some of which combined several awarding procedures. 108 or 81.2 % of these entries were made by the four responsible lecturers (one professor and three research assistants) and 25 or 18.8 % by eight of the 19 student tutors. Most of the entries related to the categories activity in live streaming, activity and presentation of solutions in Coding

Stretch goal	Algo score needed
only n-1 sheets must be passed	75,000
1. Bonus point*	150,000
1. Online exam test	225,000
Joker for selecting the explaining person in the audit (one swap is allowed)	300,000
2. Bonus point*	375,000
Christmas lecture	450,000
Examination time extension by 10 minutes	500,000
3.-5. Bonus point*	550,000 / 600,000 / 650,000
2x Examination time extension by 10 minutes	725,000 / 800,000
2. Online exam test	not reached
Submission of assignment proposals (e.g. for the exam) is allowed	not reached
Additional exam example / Upload of exam example solution	not reached

Table 2: Rewards for reaching certain Algo scores (* Bonus points have to be acquired individually within the final submission task)



Figure 2: Section from the learning material on Strings with Easteregg: Clicking on the apple leads to one version of the ‘Pen-Pineapple-Apple-Pen’ song, clicking on the pineapple to another

Classes as well as participation in questionnaires and interviews. The exact distribution of Algos by entries and Algo numbers is shown in table 3. In the category "Other", Algos were awarded, for example, for a particularly engaged student or as an apology for technical difficulties or the cancellation of an appointment. In the last entry Algos were awarded for various aspects in order to reach the final realistic stretch goal at 800,000 Algos. The distribution during the semester was mostly proportional (see figure 4), although there were occasional late additions and no Algos were awarded during the Christmas holidays.

5.2 Results of the evaluation

204 students participated fully in the evaluation. The Algo concept in general was rated as "very good" by 44.12 % of the students, as "rather good" by 35.78 %, as "neither good nor bad" by 8.82 % and as "rather bad" or "very bad" by 4.9 % and 2.45 % respectively. The remaining 3.92% gave no answer. As shown in figure

5, the majority of students thus rate the Algos rather positively. The opinion regarding Eastereggs is similar. 41.18 % perceive the presence of Eastereggs as "very positive/motivating", 34.13 % as "rather positive/motivating", 16.67 % as "neutral" and 0.49 % (one person) as "rather negative/disturbing", as figure 6 shows. 10.78% of the students stated that they had not noticed the Eastereggs in the evaluation.

Among the comments on Algos and Eastereggs, 15 students praise the humour in general, 13 other students even make concrete references to the content itself or inserted links they had chosen themselves. A further 12 respectively 4 students state that the Eastereggs respectively Algos were particularly beneficial for their learning motivation, one of them would even have competed with a fellow student to see who could find an Easteregg first during the learning. Seven students emphasise the authenticity of the course and the lecturers regarding the references in the material

Testaufgabe: Elfen helfen! (30 Punkte)

Wie schon in den Coding Classes erfahren, hat sich folgende Geschichte zugetragen: Eigentlich hatten die Elfen des Weihnachtsmanns für heute Schnee bestellt! Doch Gruntelbart hat ihren Plan gestört und da es in Osnabrück wie so häufig eigentlich zu warm für Schnee ist und die Seen nicht zufrieren, wurde aus dem bestellten Schnee Hagel, der in den See fällt und für Kreise sorgt.

Doch die Elfen wären nicht die Elfen, wenn sie nichts gegen Gruntelbarts Gemeinheiten unternehmen könnten: sie haben einen Zauberspruch gefunden, der die Hagelkreise zumindest in schöne bunte Kreise verwandeln kann, sodass sie das entstandene Muster für Geschenkpapier nutzen können.

Erstellen Sie entsprechend den Anforderungen und dem Video eine Animation, die zeigt, wie der Hagel in den See fällt und sich dort in bunte Kreise verwandelt.



Bildquelle Bildquelle

Anforderungen (für alle, die keine Lust auf die Geschichte dazu haben)

- Es sollen zwei Klassen sowie ein "Main" entwickelt werden, die eine einfach verkettete Liste, deren Verkettung über ein "next"-Attribut in der Klasse Drop funktioniert, implementiert werden.
- Erstellen Sie die Klasse Drop mit folgenden Attributen:

Figure 3: Word problem about lists with reference to "The Secret World of Santa Claus" [1] ²

Category	Number of entries	Sum of Algos	Percentage of total Algos
Forum: Questions and answers	14	9,400	1.18 %
Activity in Livestreaming	14	200,000	25 %
Additional programs, tutorials, etc.	13	13,200	1.65 %
Participation in questionnaires and in- terviews	10	220,100	27.51 %
Coding Classes: Activity, explanation of solutions	73	287,500	35.94 %
No mails to personal addresses	3	29,000	3.63 %
Other	6	40,800	5.1 %
Sum	133	800,000	100 %

Table 3: Distribution of Algos by category

and believe that they have a lasting memory as a result. Three students criticise the fact that in previous editions of the module, some of the stretch goals were already given without the prior collection of Algos.

Other comments reflect on Eastereggs as an opportunity to take a break, on the Christmas lecture and on the lack of transparency in the Algo system. There is also criticism that Algos were awarded for participation in the evaluation, although no direct learning success is associated with it. When asked how motivating the Algos were for

²Translation of the word problem before the requirements are explained: As we already learned in the Coding Classes, the following story happened: Santa's elves had actually ordered snow for today! But Gruntelbart disturbed their plan and since it is actually too warm for snow in Osnabrück, as it is often the case, and the lakes don't freeze, the ordered snow turned into hail that falls into the lake and causes circles. But the elves wouldn't be the elves if they couldn't do anything about Gruntelbart's nastiness: they found a spell that can at least turn the hail circles into nice colourful circles, so they can use the resulting pattern for wrapping paper. According to the requirements and the video, create an animation that shows how the hail falls into the lake and turns into colourful circles there.

certain activities, there is no particularly high direct motivation for activity in events and in the forum, with mean values between 1.34 and 1.7 on a three-point Likert scale from "not at all motivating (1)" to "strongly motivating (3)". The perception of being more creative in the submission tasks is slightly higher (1.85) and the highest motivation is not to write mails to personal mail addresses (2.01) as well as to participate in questionnaires (2.59). When rating the importance of achieving certain goals, the exam time extensions are assessed with a mean value of 2.75 and the bonus points with 2.68 on a scale from "not at all important (1)" to "very important (3)". The least important aspects were the non-released submission of exam assignment suggestions (1.54) and the Christmas lecture (1.74).

Looking at the evaluation results as a whole, it can be noticed that the overall satisfaction with the digital realisation is also very high compared to previous years. For example, regular participation in Coding Classes has increased significantly and the participants'

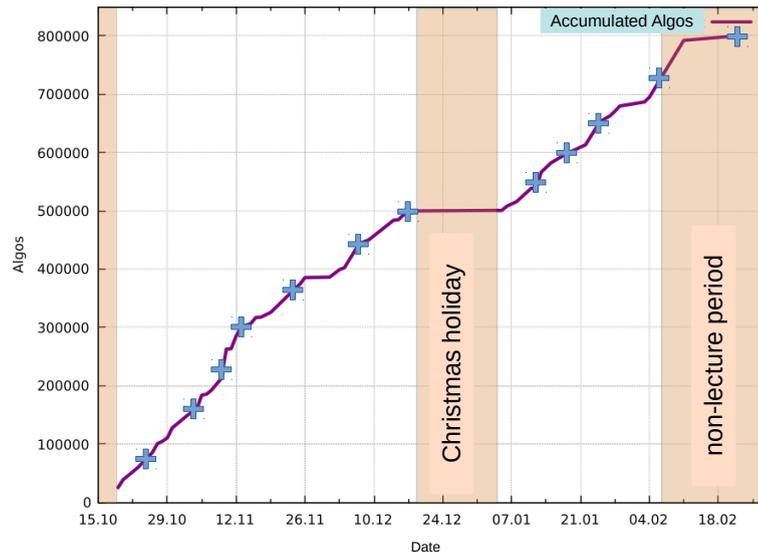


Figure 4: Development of the Algo score during the semester with marked stretch goals

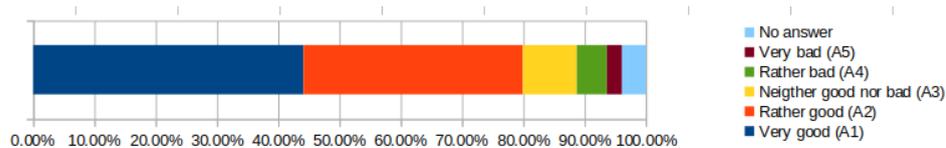


Figure 5: Assessment of the Algo concept in the evaluation

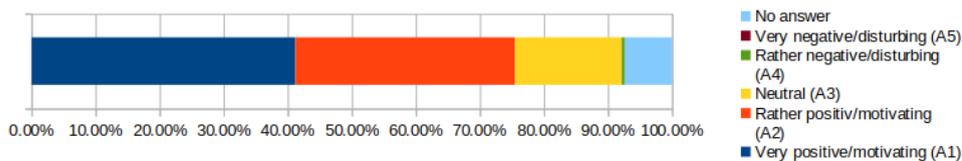


Figure 6: Assessment of the Easteregg in the evaluation

self-assessment of their learning success is also much better. When comparing the examination results with previous years, no significant differences can be found.

6 DISCUSSION AND CONCLUSIONS

Due to the assignment of Algos for the evaluation questionnaire, there is possibly a self-selection bias that should be taken into account. However, considering the high number of participants in the evaluation, also in comparison to previous years and in comparison to other, often much shorter questionnaires with significantly lower participation, it can be assumed that the effect is actually

limited. The frequent comments on content also show differentiated criticisms and not exclusively positive aspects.

So it appears that those who are more critical towards the gamification concept were also reached at least partially. In addition, the evaluation also offers students who are fundamentally more dissatisfied the opportunity to express their criticism through the questions regarding other aspects of the module.

In general, the development of the Algos during the semester was more linear than expected, which indicates that the goal of achieving Stretch-Goals at regular and realistic intervals has basically been achieved with the numbers of Algos constructed at the beginning of the semester. Only the order of the stretch goals had

to be changed occasionally in order to unlock the stretch goals in line with the events in which they are useful.

When looking at the proportions of Algos awarded for specific categories, activity in synchronous sessions makes up the largest proportion overall, with substantially more Algos awarded for activity in Coding Classes than in the Live Streaming session. However, this is appropriate given the number of Coding Classes, especially since there were some Coding Classes without good participation, while the Live Streaming sessions were almost consistently rated as very active. The proportion of Algos for questionnaires and interviews turned out to be relatively high and the assignment can only be explained in terms of improving future teaching. The Algos awarded here should therefore be slightly lower.

The criticism that students had to unlock some otherwise existing regulations in the semester with gamification is partly justified. In the previous year, the bonus points were realised for the first time and did not have to be activated, but the lecturers noticed that the number of 10 was too high in order to create fair conditions for the final grade within the group performance. The possibility of not having to pass a submission was available two years before the described semester, but for consistently weekly and accordingly more submissions overall.

The Eastereggs were generally described as well done, but sometimes a specific taste in music was mentioned. At this point, the Eastereggs are less diverse, but provide an authentic impression of the lecturers. In particular, the Eastereggs adapted in the evaluation as well as Memes about the Christmas lecture that were already circulating at the beginning of the semester indicate a high level of acceptance.

From the questions on direct motivation, it can be seen that only in the case of directly visible conversion of the activity into Algos, such as in the questionnaires or emails, an immediate motivation for such behaviour is recognisable. However, compared to the previous semester, the activity increased despite the digital scenario and was on an overall satisfactory level. Even though this was certainly additionally influenced by other factors and the Coding Classes in particular developed a schedule- and group-based dynamic after a few weeks, the Algos could have contributed to the activity here as well. For forum posts, however, simpler handling would be beneficial in order to reduce a switch to other messenger services.

The overall evaluation of the module as well as the general activity and participation were significantly better in the semester described than in the previous year. However, it should be taken into consideration that the digital implementation differed from the previous year due to other aspects apart from gamification.

No differences could be found in the examination results, but these are only comparable to a limited extent due to the digital open-books format introduced in the evaluated round. These can be rated as tending to be more difficult, as easy contents requiring a low level of competence have been significantly shortened due to the possibility of easy internet research.

Based on the development of the Algos during the lecture period and the overall positive evaluation results, it can be concluded that the gamification concept presented was fundamentally successful and could be implemented well with relatively little effort. Especially in the digital semester, the game-based elements and opportunities for pausing to learn led to a lot of praise from the

students. They often mentioned how authentically the teaching was designed aiming at Easter eggs and word problems with pop-cultural contexts and appreciated the effort that had gone into it.

The main criticism was the lack of transparency regarding the Algos, especially at the beginning, which resulted from the fact that the lecturers were still uncertain about the distribution and realistic target values. For future runs, it should therefore be clear at the beginning of the semester how many Algos have to be collected for which target. In addition, each entry should be linked to a message for the reason for the award in order to create a direct character as with the questionnaire Algos and thus also increase the visible motivation.

The effort for the Algo concept was limited to a few minutes per day for entering and possible discussions between lecturers, in addition to the basic design and implementation of the scoreboard at the beginning of the semester. From about the middle of the semester, tutors, especially from Coding Classes, had to be reminded of possible entries. It was sometimes unclear to the research assistants whether an entry had been forgotten or whether the Coding Class was simply not considered active enough for the assignment of Algos. Here, a detailed overview of the entries and activity ratings could also be useful for the lecturers.

The humorous and contextualized design of the materials in the course was sometimes slightly more time-consuming, but the research of humorous insertions and contexts also offered a pleasant change for the lecturers and was not particularly noticeable in comparison to the basic creation of the course materials.

For the construction of a scoring system for similar gamification concepts, the authors recommend calculating how many points students would get in a very active week and in a very inactive week. For this purpose, considerations similar to those described in section 4.1 can be made on how points for the intended behaviours should be weighted in relation to each other. For frequently and easily gained aspects, the score should be correspondingly low, while rare, elaborate or particularly important events should be rewarded at a higher rate.

For the first attempts, mean values of expected very active and less active weeks can then be given as targets, while small variations, especially in the direction of a weakening of the required points, do not cause great difficulties. However, a basic transparency about the considerations is important for the trust among the students.

It is planned to follow up the concept and to evaluate it in presence whenever it will be possible, in order to contribute to an even stronger cooperation between the students.

REFERENCES

- [1] [n.d.]. *Images of Gilfi and Trixi*. last accessed on 29.04.2021.
- [2] Sebastian Deterding, Dan Dixon, Rilla Khaled, and Lennart Nacke. 2011. From Game Design Elements to Gamefulness: Defining "Gamification" (*MindTrek '11*). Association for Computing Machinery, New York, NY, USA, 9–15. <https://doi.org/10.1145/2181037.2181040>
- [3] Darina Dicheva, Keith Irwin, and Christo Dichev. 2019. OneUp: Engaging Students in a Gamified Data Structures Course. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (Minneapolis, MN, USA) (*SIGCSE '19*). Association for Computing Machinery, New York, NY, USA, 386–392. <https://doi.org/10.1145/3287324.3287480>
- [4] Juho Hamari and Jonna Koivisto. 2013. Social motivations to use gamification: An empirical study of gamifying exercise. *ECIS 2013 - Proceedings of the 21st European Conference on Information Systems*.

Cooperative Gamification in a Computer Science Introductory Module

CSERC '21, November 22–23, 2021, Virtual Event, Netherlands

- [5] Celine Latulipe, N. Bruce Long, and Carlos E. Seminario. 2015. Structuring Flipped Classes with Lightweight Teams and Gamification. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education* (Kansas City, Missouri, USA) (SIGCSE '15). Association for Computing Machinery, New York, NY, USA, 392–397. <https://doi.org/10.1145/2676723.2677240>
- [6] P. Pintrich. 2003. A Motivational Science Perspective on the Role of Student Motivation in Learning and Teaching Contexts. *Journal of Educational Psychology* 95 (2003), 667–686.
- [7] Gina Sprint and Erik Fox. 2020. Improving Student Study Choices in CS1 with Gamification and Flipped Classrooms (SIGCSE '20). Association for Computing Machinery, New York, NY, USA, 773–779. <https://doi.org/10.1145/3328778.3366888>
- [8] Stefan Weyers. 2018. Inverted Classroom meets Gamification - Einsatz eines Bonuspunktesystems in einer Grundlagenveranstaltung. In *Inverted Classroom Vielfältiges Lernen Begleitband zur 7. Konferenz Inverted Classroom and Beyond 2018* (St. Pölten). 245–249.

A.3 Beitrag 3

Übersicht zu Beitrag 3

Titel:	Warum Onlinelehre sowie Open-Books-Klausuren und das Fach Informatik zusammenpassen
Autoren:	Elisaweta Ossovski Michael Brinkmeier Daniel Kalbreyer
Jahr:	2021
Status:	veröffentlicht
Bibliographische Informationen:	Ossovski, Elisaweta; Brinkmeier, Michael; Kalbreyer, Daniel: Warum Onlinelehre sowie Open-Books-Klausuren und das Fach Informatik zusammenpassen. In: Kienle, Andrea (Hrsg.) ; Harrer, Andreas (Hrsg.); Haake, Joerg M. (Hrsg.); Lingnau, Andreas (Hrsg.): DELFI 2021. Bonn: Gesellschaft für Informatik e.V., 2021. – ISBN 978-3-88579-710-4, S. 265-270
Link zur Veröffentlichung:	https://dl.gi.de/handle/20.500.12116/37020

Warum Onlinelehre sowie Open-Books-Klausuren und das Fach Informatik zusammenpassen

Elisaweta Ossovski¹, Michael Brinkmeier¹ und Daniel Kalbreyer¹

Abstract: Durch die Covid19-Pandemie waren Lehrende an vielen Hochschulen gezwungen innerhalb kurzer Zeit Konzepte für digitale Lehre zu entwickeln. Obwohl in Ausnahmefällen zumindest Prüfungen in Präsenz ermöglicht wurden, fand eine hohe Zahl an Klausuren ebenfalls vollständig digital statt. Die Autoren dieses Positionspapiers haben Erfahrungen mit der digitalen Lehre in einem großen Einführungsmodul der Informatik mit insgesamt drei Open-Books-Klausuren gesammelt und festgestellt, dass sowohl die Informatik ein passendes Fach für Onlinelehre sowie Open-Books-Klausuren ist als auch, dass diese für die Informatik besonders gut geeignet sind. Die gewonnenen Erfahrungen und Gründe für diese Thesen werden in dem Positionspapier erläutert, bevor Empfehlungen für Lehrende aller Fächer abgeleitet werden.

Keywords: digitale Lehre, Online-Lehre, open books, digitale Prüfung, Informatik

1 Einleitung

Schon seit einiger Zeit gewinnt digitale Lehre an deutschen Hochschulen an Bedeutung. So gaben 2016 62 % der Befragten in einer Studie des HIS-Instituts für Hochschulentwicklung [Wa16] an, dass digitale Lehre an ihren Hochschulen einen mindestens überdurchschnittlichen Stellenwert besäße. Während in dieser Studie für 73 % der Befragten vor allem der punktuelle Einsatz digitaler Elemente im Vordergrund stand, wurde nur bei 17 % zum Teil auch rein digitale Lehre durchgeführt [Wa16]. Nur 17 % der Befragten gaben damals an den Ausbau rein online-basierter Lehr- und Lernangebote anzustreben. Die Covid19-Pandemie führte allerdings dazu, dass Lehrende an Hochschulen gezwungen waren nahezu vollständig auf digitale Lehre auszuweichen und Präsenz nur in Ausnahmefällen erlaubt wurde. Digitale Prüfungen zu summativen Zwecken, d.h. zur tatsächlichen Leistungsbewertung, wurden vor der Pandemie an 32 % der einbezogenen Hochschulen zumindest erprobt [Wa16]. Wie sich aus einer Reihe von internen Befragungen von Lehrenden und Studierenden ergab, scheint der plötzliche Umstieg auf digitale Szenarien in vielen Fällen besser gelungen zu sein als erwartet wurde. Basierend auf Beobachtungen, die auf einer Reihe von Befragungen und Lehrveranstaltungen, insbesondere auf einer großen Erstsemester-Veranstaltung mit etwa 500 Teilnehmern verschiedener Fächer im Flipped Classroom-Format mit interaktiven Lernmaterialien, beruhen, wird eine Reihe von Thesen formuliert, die begründen, warum das Fach Informatik sich besonders für die Einführung digitaler Lern- und Lehrformate eignet.

¹ Institut für Informatik, Universität Osnabrück, Wachsbleiche 27, 49090 Osnabrück, {ossovski, mbrinkmeier, dakalbreyer}@uos.de. E. Ossovski ist Stipendiatin des Ernst-Ludwig-Ehrlich-Studienwerks.

2 Wieso die Informatik und Onlinelehre zusammenpassen

Dass digitale Lehre und Informatik eine Verbindung zueinander haben, wurde spätestens mit der Dagstuhl-Erklärung [Br16] festgestellt. Im digitalen Wintersemester 2020/21 fiel den Autoren dieses Papiers jedoch überraschenderweise auf, dass die Zufriedenheit mit dem betreuten Informatikmodul im vollständig digitalen Lehrszenario gegenüber den (bereits positiven) vorherigen Jahren sogar gestiegen war. Bestätigt wird dies durch erste Ergebnisse der CHE-Sonderbefragung zum Studium während der Coronapandemie. Dort zeigt sich, dass die digitale Lehre im Fach Informatik sowohl an Universitäten als auch an Fachhochschulen in diversen Kategorien als gut bewertet wird [CHE21].

2.1 Hohe Technikaffinität aller Beteiligten

Sowohl bei den Lehrenden als auch den Studierenden der Informatik ist von einer erhöhten Technikaffinität auszugehen. Da in fachlichen Kontexten regelmäßig mit digitalen Werkzeugen und einem kurzen Innovationszyklus gearbeitet wird, bestehen vermutlich weniger Ängste neue Technologien auch in der Lehre und in ungewohnten Situationen zu nutzen. Zum anderen scheinen Informatiker*innen in der Regel auch über eine Reihe von für digitale Lehre nützlichen Kompetenzen zu verfügen. Das zugrundeliegende technische Wissen scheint die Einarbeitung in die Funktionsweise und den Aufbau von informatischen Systemen und in neue Werkzeuge zu erleichtern. Parallelen und Grundprinzipien, aber auch Unterschiede ähnlicher Angebote werden schneller erkannt und führen zu einer reflektierteren Auswahl passender Werkzeuge. Darüber hinaus sind informatisch gebildete Personen auch in der Lage Programme und Tools für die digitale Lehre selbst anzupassen oder zu entwickeln. Daher werden die existierenden Werkzeuge nicht als statisch und unveränderlich wahrgenommen. Im Gegensatz dazu sind Lehrende ohne entsprechende Kompetenzen häufig darauf angewiesen, dass die vorhandene Infrastruktur entsprechende Funktionalität zur Verfügung stellt und können nur aus existierenden Programmen wählen. Anpassungen, das Integrieren eigener Ideen oder Kombinieren verschiedener Werkzeuge stellen für sie eine deutlich höhere, zum Teil unüberwindbare, Hürde dar.

These 1: Informatische Kompetenzen der Lehrenden erlauben einen flexibleren, reflektierteren Umgang mit digitalen Werkzeugen und begünstigen die Entwicklung und Einführung digitaler Lern- und Lehrformate.

2.2 Natürlichkeit technisch-sozialer Interaktionen

Im späteren Berufsleben werden Informatiker*innen häufig Konzepten wie agilem Arbeiten sowie technisch-sozialen Interaktionen begegnen [Hi01]. Tatsächlich ist festzustellen, dass eine ganze Reihe von arbeitsorganisatorischen Innovationen, wie z. B. Scrum oder kooperative, asynchrone Arbeitsformen, aus der Informatik stammen und vermehrt in anderen Berufsfeldern adaptiert werden. Im Bereich der Informatik ist es sehr weit verbreitet technische Hilfsmittel und feste Regelwerke zur Organisation und

Kommunikation zu nutzen. Daher ist die Vermittlung entsprechender Kompetenzen ein wichtiges Ziel des Informatikstudiums. Die digitale Lehre bietet die Möglichkeit, dass Studierende bereits im Studium – idealerweise vom ersten Semester an – diese Formen der Arbeitsorganisation kennenlernen. So können zur Vermittlung und Organisation der Lernprozesse bereits entsprechende Werkzeuge genutzt werden. Außerdem lernen sie in digitalen Szenarien, die Komponenten des sozialen Lernens einschließen, die Kooperation in Gruppen mit Hilfe technischer Werkzeuge, Materialien und Informationen auszutauschen und an gemeinsamen Dokumenten asynchron zu arbeiten.

These 2: Eine ganze Reihe von Kompetenzen, die im Rahmen der digitalen Lehre von Studierenden erworben werden, sind inhärenter Bestandteil der informatischen Berufsfelder.

2.3 Offene Lehr- und Lernkultur

In überfachlichen kollegialen Gesprächen wurde berichtet, dass Studierende sich in Onlinekonferenzen häufig eher passiv verhalten und oft nur wenige Fragen stellen. Einige Lehrende führen dies darauf zurück, dass Studierende den Verlust ihrer Anonymität befürchten. Diese Beobachtung können die Autoren nicht bestätigen. Obwohl den Studierenden im Rahmen der Lehrveranstaltungen, insbesondere der Grundlagenvorlesung, diverse Möglichkeiten angeboten wurden anonym Fragen zu stellen, wurde kein einziges Mal davon Gebrauch gemacht, stattdessen wurden in allen Formaten (Livestreaming, Übung, Forum) Fragen unter Anzeige des Namens offen gestellt. Auch in der Evaluation wurde mehrfach geäußert keine Bedenken diesbezüglich gehabt zu haben. Bei der sonstigen Kommunikation konnten – mit Ausnahme einer grundsätzlich höheren Anzahl an E-Mails, die im Wesentlichen die Fragen vor und nach Präsenzveranstaltungen ersetzen – ebenfalls keine signifikanten Abweichungen zu vorherigen Semestern festgestellt werden.

Die Autoren führen dies darauf zurück, dass in der Informatik der Universität Osnabrück überwiegend eine zwanglose Umgangsweise zwischen den verschiedenen Statusgruppen, zu denen auch studentische Tutor*innen gehören, herrscht. Eine offene Fehler- und Kommunikationskultur scheint eine erhöhte Akzeptanz von Lehrexperimenten selbst bei auftretenden Schwierigkeiten zu bewirken. Dies erleichterte die Einführung einer neuen experimentellen technischen Lernplattform. Außerdem wurden z. B. Elemente von Gamification sowie humorvolle Einschübe oder Kontexte (z. B. in Form von „Eastereggs“) erprobt. Insbesondere diese, keinen direkten inhaltlichen Mehrwert bietenden Elemente, wurden von den Studierenden in der abschließenden Evaluation als besonders positiv hervorgehoben und sogar selbst aufgegriffen. Dies schien auch eine positive Wirkung auf das kooperative Verhalten zwischen den Studierenden gehabt zu haben.

These 3: Niedrige Hürden bei der Kommunikation und gegenseitige Wertschätzung fördern die aktive Kooperation zwischen Studierenden untereinander wie auch zu Lehrenden und erlauben Lehrexperimente.

2.4 Inhalte im Wandel

Im Vergleich zu vielen anderen Disziplinen ist die Informatik trotz ihrer teils lange existierenden mathematischen Grundlagen eine vergleichsweise junge wissenschaftliche Disziplin [De00]. Vor diesem Hintergrund sowie vor dem Hintergrund der hohen Geschwindigkeit technischer und informatischer Entwicklungen in den letzten Jahrzehnten [Mo65] befinden sich auch Teile der Lehrinhalte und die Werkzeuge im ständigen Wandel. Wurde in den 1970er Jahren beispielsweise noch rein imperative Programmierung mit Pascal gelehrt, fanden schon 20 Jahre später objektorientierte Sprachen Einzug in die Grundlehre. Mit den sich verändernden Inhalten und Werkzeugen ist daher auch für Lehrende in Grundlagenkursen der Informatik eine regelmäßige Neugestaltung der Inhalte und damit auch der Materialien und Vermittlungskonzepte notwendig. In diesem Rahmen können dann auch aktuelle Ergebnisse didaktischer Forschung und neue digital gestützte Methoden und Werkzeuge einbezogen werden. Die Umgestaltung eines Kurses nach didaktischen Kriterien stellt damit einen geringeren Aufwand dar als ohne konkreten Anlass überhaupt etwas an der Lehre zu ändern. Die didaktische Forschung setzt dabei in den letzten Jahren in diversen Fächern auf die Nutzung digitaler Technologien und stellt diese überwiegend als vorteilhaft dar, wie Erkenntnisse zum Flipped Classroom, Virtual/Augmented Reality oder auch etablierter Lernsoftware zeigen. Auffällig ist, dass in der CHE-Befragung [CHE21] insbesondere Fächer mit hohen Abbruchquoten, neben Informatik [HRS20] auch Physik und Mathematik, besonders gut bewertet werden. Neben dem Item „Vielfalt digitaler Lehrformate“ trifft dies auch auf die didaktischen Konzepte zu. Die digitale Lehre scheint hier als Katalysator für eine didaktisch bessere Aufbereitung gewirkt zu haben. Bezogen auf die diesem Papier zugrundeliegenden Veranstaltungen wurde dieser Effekt dadurch verstärkt, dass die Lehrenden zu einer fachdidaktischen Arbeitsgruppe gehören.

These 4: Eine hohe Dynamik der Inhalte und Methoden fördert eine Umstellung auf digitale Formate. Dadurch kann sowohl die redaktionelle Arbeit als auch die Grundstruktur der Lehrveranstaltung angepasst werden. Das Begreifen von Veränderungen als didaktische Gelegenheiten kann zur nachhaltigen Verbesserung der Lehre führen.

3 Wieso Informatik und Open-Books-Klausuren zusammenpassen

Obwohl zumindest für Prüfungen während der Pandemie häufig Ausnahmen für das Präsenzverbot galten, fanden zur digitalen Lehre überwiegend auch digitale Prüfungen statt. An der Universität Osnabrück fanden mit 32,3 % aller Fachbereichs-Prüfungen im Sommersemester 2020 die zweitmeisten digitalen Klausuren im Fachbereich Mathematik/Informatik statt [Ha21]². Bei genauerer Betrachtung der im Fachbereich

² Die meisten im Fachbereich Rechtswissenschaften, dies ist jedoch eher auf die deutlich höheren Studierendenzahlen sowie auf die grundsätzliche Prüfungsart zurückzuführen.

Mathematik/Informatik angebotenen Prüfungen fanden Präsenzprüfungen jedoch lediglich in der Mathematik statt. Die Lehrenden der Informatik setzten dabei vor allem auf Open-Books-Klausuren, bei denen alle Hilfsmittel außer dem Kontakt zu anderen Personen während der Klausur erlaubt waren. Auch vor der Pandemie gelang es einzelnen Lehrenden der Informatik in digitalen Open-Books-Klausuren mindestens vergleichbare, in jedem Fall aber angemessene Alternativen zur klassischen Papierklausur in Präsenz zu finden. Die möglichen Gründe dafür werden im Folgenden genauer betrachtet.

Ein hoher Kompetenzbezug im Lernstoff ermöglicht im Rahmen des Constructive Alignment [Bi96] auch kompetenzbezogene Aufgaben, die über die Abfrage von Faktenwissen hinausgehen. Die Informatik lebt vom Lösen konkreter Probleminstanzen, für die z. B. Algorithmen entwickelt oder auf die bereits existierenden Lösungen angewendet werden. Dadurch lassen sich auch in Klausuren Aufgaben stellen, die nicht trivial durch Nachschlagen, Internetrecherche oder bereits im Vorfeld festgehaltene Informationen beantwortbar sind. Dies bestätigte sich beim Überarbeiten von älteren Präsenzklausuren, für die keine Hilfsmittel zugelassen worden waren. Nur in wenigen Fällen war es möglich die Aufgaben mithilfe einer Internetrecherche direkt zu lösen. In der Regel handelte es sich dabei um Aufgaben, bei denen bekannte Algorithmen manuell durchgeführt werden sollten. Dafür waren häufig adaptive Werkzeuge verfügbar. Der Aufgabenkern muss sich daher für viele Aufgaben in Informatikklausuren daher gar nicht ändern, sondern lediglich die Art der Aufgabe muss an die verfügbaren Hilfsmittel angepasst werden. Durch Reproduktion, Anwendung, Variation und Implementation kann dasselbe informatische Artefakt (z. B. Algorithmus) in Aufgaben auf verschiedenen Anforderungsstufen, mit verschiedenen Zugängen und Abstraktionsniveaus variiert werden. Geeignete Variationen von Probleminstanzen erschweren den Einsatz von im Internet vorhandenen Lösungen und ermöglichen die Erstellung unterschiedlicher Varianten einer Aufgabe, die Absprachen zwischen Studierenden deutlich erschweren und insbesondere bei Open-Books-Klausuren ohne Aufsicht von hoher Bedeutung sind.

Im Kontext von digitalen Klausuren bieten syntaktisch formale Antworten einen Vorteil bei der automatisierten Korrektur. In der Informatik ist das Einhalten komplexer syntaktischer Regeln eine grundlegende Kompetenz, die u. a. beim Lernen von Programmiersprachen relevant ist. Daher bietet sich in der Informatik über die übliche Form von z. B. Multiple-Choice-Fragen hinaus auch die Möglichkeit für komplexere Aufgaben eine einheitliche Struktur zu fordern und deren Einhaltung auch zu bewerten.

These 5: Die Möglichkeit von stark kompetenz- und anwendungsbezogenen Aufgaben sowie die Formulierung von strikt syntaktischen Antworten erleichtern die Konzipierung (und Korrektur) digitaler Open-Books-Klausuren.

4 Folgerungen

Die erläuterten Thesen sollten keinesfalls als reine Begründung verstanden werden,

wieso die Informatik bei digitaler Lehre gewisse Vorteile hat, sondern viel mehr dazu motivieren Gemeinsamkeiten und Differenzen bewusst zu machen. Eine auch außerhalb digitaler Lehre häufig geforderte Strategie kann dabei sein, stark kompetenz- und problemorientiert zu lehren. In diesem Fall kann der Aufwand für digitale Klausuren, der neben technischen Anforderungen im Wesentlichen in der Findung geeigneter Aufgaben liegt, durchaus durch geringeren Korrekturaufwand mit automatisierter Korrektur ausgeglichen werden. Grundsätzlich sollten Ängste vor technischen Problemen abgebaut werden. Hierbei kann es hilfreich sein den Austausch zwischen Lehrenden verschiedener Fächer zu fördern. Da nicht jede Universität über Personal für Hochschuldidaktik und digitale Lehre verfügt, sollten besonders digital affine Lehrende, die es in der Informatik häufig gibt, offen für Austausch sein und nach Möglichkeit unterstützen, wenn technische Entwicklung benötigt wird. Je größer die Nachfrage nach digitalen Lösungen dann wird, desto mehr Anreize bestehen auch für solche Entwicklungen, um auch nach der Pandemie die Vorteile beider Lehrkonzepte miteinander vereinen zu können.

Literaturverzeichnis

- [Bi96] Biggs, J.: Enhancing teaching through constructive alignment, in: Higher Education 32, S. 347–364, Kluwer Academic Publishers, 1996.
- [Br16] Brinda, T., Diethelm, I., Gemulla, R., Romeike, R., Schöning, J., Schulte, C. et al.: Dagstuhl-Erklärung: Bildung in der digitalen vernetzten Welt. Gesellschaft für Informatik e.V., 2016.
- [CHE21] CHE: Centrum für Hochschulentwicklung: Ergebnisse der CHE Studierendenbefragung 2020/21 zur Corona-Pandemie und zur digitalen Lehre, <https://shiny.che-ranking.de/corona2021/>, 2021, letzter Zugriff am 24.03.2021.
- [De00] Denning, P. J.: Computer Science: The Discipline in: Ralston, A.; Hemmendinger, D. (Herausgeber): Encyclopedia of Computer Science, 2000.
- [Ha21] Hamborg, K., Mainka, A., Kukharenska, N., Kötter, J., Nordholt, S.: Befragung der Lehrenden zum digitalen Sommersemester 2020, Working Paper, virtOUS Zentrum für Informationsmanagement und virtuelle Lehre, Universität Osnabrück, 2021.
- [Hi01] Highsmith, J.: History: The Agile Manifesto, abrufbar unter <https://agilemanifesto.org/history.html>, 2001, letzter Zugriff am 08.04.2021.
- [HRS20] Heublein, U., Richter, J., Schmelzer, R.: Die Entwicklung der Studienabbruchquoten in Deutschland. (DZHW Brief 3/2020). Hannover: DZHW, 2020.
- [Mo65] Moore, G. E. : Cramming more components onto integrated circuits, Reprinted from Electronics, volume 38, number 8, April 19, 1965, pp.114 ff., in *IEEE Solid-State Circuits Society Newsletter*, vol. 11, no. 3, S. 33-35, 2006.
- [Wa16] Wannemacher, K., Jungermann, I., Osterfeld, S., Scholz, J., von Villiez, A. (HIS-Institut für Hochschulentwicklung (HIS-HE)): Organisation digitaler Lehre in den deutschen Hochschulen, 2016.

A.4 Beitrag 4

Übersicht zu Beitrag 4

Titel:	COMPARING VIDEO AND INTERACTIVE LEARNING MATERIAL STYLES FOR PROGRAMMING
Autoren:	Elisaweta Ossovski Michael Brinkmeier
Jahr:	2022
Status:	veröffentlicht
Bibliographische Informationen:	Ossovski, Elisaweta; Brinkmeier, Michael: COMPARING VIDEO AND INTERACTIVE LEARNING MATERIAL STYLES FOR PROGRAMMING. In: EDULEARN22 Proceedings, IATED, 2022 (14th International Conference on Education and New Learning Technologies). – ISBN 978–84–09–42484–9. http://dx.doi.org/10.21125/edulearn.2022.1273 . S.5381–5390
Link zur Veröffentlichung:	https://library.iated.org/view/OSSOVSKI2022COM

COMPARING VIDEO AND INTERACTIVE LEARNING MATERIAL STYLES FOR PROGRAMMING

Elisaweta Ossovski, Michael Brinkmeier

*University of Osnabrück, Department of Computer Science, Didactics of Computer Science
(GERMANY)*

Abstract

Learning to program is considered to be a difficult activity [1]. As in other subjects, flipped classroom approaches are therefore increasing in Computer Science [2]. For a successful flipped classroom, there is a need for asynchronous learning material, for example videos. Educational videos for learning programming can be produced in various styles. These differences can for instance relate to the production effort or how the code is displayed. During Live-Coding the development of a program code is shown as a process. Depending on the preparation mistakes can occur, which can be shown to the learners and solved afterwards or cut out in the post-production if no further attempts to write the program without mistakes should be undertaken. Live-Coding videos that should appear to be perfect from many lecturers' point of view need extensive preparation and at least some production effort.

In this study low effort Live-Coding videos with nearly no post-production, Live-Coding videos with some (post-)production effort and an explanation video with finalized code are compared to a commented code development in Jupyter Notebooks and pure commented code. Jupyter Notebooks offer the possibility to interact with the content, e. g. to copy, change and run the code and are suitable for adjustments by the instructors without high effort. Students from an introductory Computer Science university course got a random learning material about a graph problem and should then perform a transfer coding task on a similar problem.

Besides the learning transfer success, also the students' preferences and attitudes towards the different learning material types were analyzed. A total of 85 students with varying levels of prior knowledge completed the study. The results suggest that the type of learning material has no significant effect on the learning transfer success, although not all participants were able to solve the rather difficult transfer task. The success correlated with the previous knowledge of the participants across all groups, but no significant differences could be found between the groups with different learning materials. Despite the concerns of some lecturers, especially low production effort seems not to be a major problem. Some students even mentioned that occurring errors can be helpful in order to understand typical mistakes. Additionally, students have preferences regarding the style as videos and more extensive insights in the development are preferred to pure commented code.

Keywords: videos, programming, live coding, worked examples, Jupyter Notebook, production effort.

1 INTRODUCTION

With the increasing digitalisation in education and asynchronous teaching formats, the question of the design of digital asynchronous learning materials becomes relevant. On the one hand, these should obviously achieve the best possible learning success, but on the other hand, the materials should also fulfil the individual criteria of the learners and teachers. Although several connections between these criteria and the actual learning success are definitely clear, some design decisions also result from personal preferences or assumptions by the lecturers. For example, experience has shown that some teachers are concerned about providing learning materials that are not completely impeccable. However, they overlook the fact that in-class teaching, e.g. in a lecture, is not always perfect as well. These perceived imperfections may also contribute to authenticity and thus to learner satisfaction.

The aim of this study was to explore what kind of learning material has which effect on the teaching of programming. Specifically, the effort of post-production or the freedom from errors, whether a learning material is a video and whether the programming process itself or finalized code is shown should be varied. The impact should be analysed with regard to the learning performance on the one hand and the reception by the learners on the other. In particular, a target was to find out whether a more extensive (post-)processing of the video, which often takes a lot of time and corresponds to the perfectionist thinking of many lecturers, actually leads to better performance and higher satisfaction or not. With the

distinction between videos and other materials, the main aim was to compare interactive materials that allow learners to make their own changes and exploration to more common videos.

2 LITERATURE REVIEW

The discourse on designing media for learning has been going on for a long time. Since the Clark-Kozma debate, which began in 1983 with the statement that media merely provide content but do not lead to a learning process [3], different opinions have existed about what effect media and their design have on the learning process. Clark argued with the interchangeability of the material and distinguished between material and method. Kozma, on the other hand, argued that media is used in a methodical context and thus can have an impact on learning [4]. However, in times of MOOCs and digital teaching, media can encompass both the pure medium as a tool used and an underlying methodology. In 2018, Poquet et al. [5] did a meta analysis of studies focussing on learning with videos. The majority of students in the analysed studies were from STEM subjects and among other types of animated videos studies on worked examples were also included. The results suggest that different aspects may have an impact to recall and transfer.

In addition to a number of theoretical principles for designing media like Sweller's Cognitive Load Theory [6], Mayer's principles for Multimedia Learning [7] or Butcher's Multimedia Principle [8], there exist several studies on the design of videos, which, like in this study, compare different types of videos with each other. In this context, Altinpulluk et al. [9] found, that video segmentation is beneficial for the Cognitive Load. De Koning et al. [10] analysed the effect of graphic design elements such as animations or arrows that point to selected content.

In the context of Cognitive Load, Worked Examples have also been recognised as a beneficial way to learn [11], [12]. Within programming learning, there are also approaches as to how Worked Examples can be designed. Although some interactive approaches with self-explanation prompts or scaffolding have led to an improvement concerning Cognitive Load, a direct design recommendation in terms of learning effect is still difficult to define [13], [14], [15]. The conclusion that interaction is more helpful in learning than watching videos is supported by Koedinger et al. [16]. The authors analysed learning success and dropout rates among MOOC participants in a psychology course, some of whom had access to an interactive learning portal. However, this study is only partly transferable to the design of individual materials for learning programming, since the activity is followed by exercises at the end of each learning unit.

Focusing on video design, an aspect that has been studied several times but is still controversial is the question of whether the instructor should be visible on the video next to the presentation. While Bateman and Schmidt-Borcherding [17] as well as Wang and Antonenko [18] found positive influences of instructor presence on several aspects of students' perception and Cognitive Load, a positive influence on the actual learning could not be found and other research found no effects at all [19]. Schmidt-Bocherding and Drendel [20] investigated in comparing videos with different coherence in terms of dynamic development and visibility of the current topic or completely visible slides. Their results indicate that coherence is more important than the instructor's presence. For the implementation of dynamic development in videos, Cross et al. [21] investigated in comparing handwriting in videos with typing and automatic straightening. They found that the latter is a good way to combine the advantages of both previous ways.

Several research studies show that coherent auditive explanations of learning materials increase learning success [22], [23], [24]. Comparing traditional book learning to talking head and tablet drawing videos Ilioudi, Giannakos and Chorianopoulos [25] found better learning effects by videos with 36 students, especially with the talking head style. In a larger scenario, Guo, Kim and Rubin [26] compared captured lectures with talking head, writing and slides in videos. They found, that overall short videos are preferred and talking head and writing videos cause better engagement in terms of spent time and willingness to answer questions. Furthermore, they suggest that extensive post-production is not necessary. Regarding errors in programming, the trial and error approach is generally common among novice programmers. However, with approaches like Test Driven Development, a stronger reflection can take place [27]. Nevertheless, different errors must be differentiated with regard to the cause and the learning opportunities [28], [29].

In this study, the focus is primarily on teaching the process of programming. This topic is different from frequent knowledge transfer as it is a demanding task from a learning taxonomy point of view. In particular, the dynamics of learning materials in terms of programme development as well as the

interactivity of the learning material are analysed. However, the question of post-production is also relevant, since mistakes can often occur, especially in live coding, and these are natural in the real programming process.

3 METHODOLOGY

In order to better understand the effects of different types of live coding videos and other coding materials, different materials on the same topic were created and compared with each other using a transfer task and subsequent student evaluation. To ensure good comparability, in all variants the same code was implemented and explained using nearly the same words by the same person. The videos were recorded with the same technology and all materials use the same code editor (JupyterNotebooks). To ensure a suitable quality, the design principles mentioned in section 2 were applied wherever it was possible. For the study five types of learning material were created, whose properties are explained in table 1:

- Type 1: Live-Coding video with nearly no post-production (just one cut was needed), which includes mistakes occurred during the coding process, length: 15:48 minutes
- Type 2: Live-Coding video with some post-production including cutting, where several retakes were needed and only minor mistakes were allowed, length: 10:41 minutes
- Type 3: Video with an explanation of finalized code worked example, length: 6:21 minutes
- Type 4: Code development in Jupyter Notebook, including possibilities to run and change the code
- Type 5: Pure commented code worked example, including possibilities to run and change it

Table 1. Properties of the different learning materials, LC = Live Coding.

	1. LC-video long	2. LC-video short	3. explanation video	4. Jupyter Notebook	5. explained code
Video	yes	yes	yes	no	no
Interactive (running and changing code)	no	no	no	yes	yes
Code	development	development	finalized	development + finalized	finalized
Production effort	medium	high	low/medium	medium	low
Programming mistakes	possible	not desired	not possible	possible if desired	not possible

Screenshots of the material types 1 and 4 are shown in figure 1. The other two video materials (types 2 and 3) look very similar to the type 1 material. The only difference to the type 2 material consists in the more extensive production effort including multiple captures and post-production, so that there are no major mistakes. In the type 3 material the code is not typed during the video but the finalized code is commented. The type 5 material consists only of code cells which can also be seen in material 4.

Although there is some empirical evidence that the speaker's visual appearance in a video can be supportive for satisfaction [17], it also attracts visual attention [19] and could distract from the actual content [20]. In order to avoid this factor influencing the distinction between videos and interactive material, all videos were created without a speaker's appearance and only with the voice.

The study was guided by a questionnaire, in which first general personal information as well as previous knowledge was asked for, while preserving the anonymity of the participants. Then a learning material was displayed, which was accessible further on throughout the study. Participants were encouraged to study the learning material until they felt comfortable with it. They were then given a similar transfer task to complete in their development environment, which is also used for the course. The output of their solution should then be submitted to the questionnaire. Afterwards, the participants were asked how they felt about the learning material, whether they consciously stopped after a certain time with the task and what kind of learning material they would prefer in the future. They were therefore provided with examples of where the types used in the study had already been used in the course, so that they did not have to review the rest of the study materials.

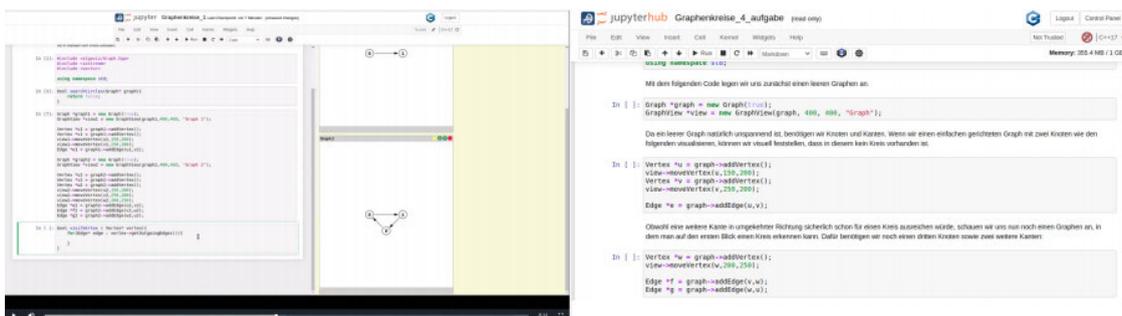


Figure 1. Screenshots of the Type 1 long Live Coding video (left) and of the Type 4 Jupyter Notebooks material (right).

3.1 Topic and Task

In line with the course content at this time, graph algorithms were chosen as the topic for the learning material and transfer task. The learning material presented how to algorithmically determine whether a cycle is contained in a given graph. For this purpose, the already known depth-first search was modified and adapted to the problem. The basic implementation of a graph was also already known. In the transfer task, the participants were then asked to complete a method for a given setting, which contained some test graphs, that checks whether a given graph is strongly connected.

To check the solution, four different test graphs were tested for the strong correlation property using the students' solution. If this was correct, a random test code with specific characteristics was generated. The students entered this code into the questionnaire so that the functionality of the implementation could be checked.

There was indeed a mistake in the type 1 material, as the marking of nodes at a certain point was initially forgotten during the development of the program. This was then resolved and explained. In the type 2 material, there were only minor typing errors that were directly corrected, but no logical programming mistakes. In the type 4 material (Jupyter Notebook), the program components were explained in a step-by-step manner, building on each other, whereby the existing implementations of the graph and the depth search were used and these were adapted to the given problem.

3.2 Participants and Conducting

The data collection took place in the second half of a CS1 course with about 550 students from different courses of study, including many non-majors in CS. The questionnaire was accessible via an online platform for one week and the students had the opportunity to work on the learning materials and the task within a practical exercise in the course. A total of 342 clicks were detected, with 85 students completing the study completely. The processing time was not limited, but measured. The allocation of a learning material was conducted on a random basis at the beginning of the questionnaire. Because of dropouts, there are small differences in the number of participants for the different materials. The distribution of participants and their mean prior knowledge self-assessment score is shown in Table 2. This score was determined by asking the participants to rate themselves in the six categories consisting of computer science in general, programming, C++, algorithms, logical thinking and pair/small groups programming on a 4-level scale between "no previous knowledge at all" and "I can already do it well". There are no significant differences between the self-assessed prior knowledge between the groups with different learning materials (Kruskal-Wallis: Chi-square = 1.7223, $p = 0.7867$). Among the participants 58.82 % were female and 41.18 % male with an average age of 21.21 years. As an incentive to participate, points were awarded as part of a gamification concept, which led to rewards for the whole course.

Table 2. Participants per learning material style (*PKSC = Prior knowledge self-assessment score, range 6 - 24).

	1. LC-video long	2. LC-video short	3. explanation video	4. Jupyter Notebook	5. explained code	Total
Number	13	16	14	22	20	85
Percentage	15.29 %	18.82 %	16.47 %	25.88 %	23.53 %	100 %
Mean PKSC*	12.46	13.56	13.36	12.32	13.15	12.94

4 RESULTS

In addition to the success in the transfer task, the participants were also asked for their assessment of how they felt about the material and which type of material they preferred after working through the learning material.

4.1 Learning transfer success

Table 3 shows the results for the transfer task performed by the different learning material groups. Although there is some variation, there are no statistically significant differences between the groups (Kruskal-Wallis: Chi-square = 0.63782, $p = 0.9588$). However, when looking at the participants' self-assessed prior knowledge, a strongly significant correlation to success in solving the transfer task can be identified (Spearman's rank correlation: $p < 0.001$, $r = 0.4342$). It was also checked whether, possibly due to the type of material, different rates of participants in the groups dropped out before completing the questionnaire. However, no correlation could be found between the type of material and the progress in the questionnaire (Kruskal-Wallis: Chi-square = 2.6095, $p = 0.6251$). The deliberately stated stopping before a satisfactory solution was reached does also not show any correlation with the material type (Kruskal-Wallis: Chi-square = 0.6768, $p = 0.9542$).

Table 3. Results for transfer tasks by learning material.

	1. LC-video long	2. LC-video short	3. explanation video	4. Jupyter Notebook	5. explained code
Right solution	38.46 %	43.75 %	35.71 %	31.82 %	40 %
Wrong or missing solution	61.54 %	56.25 %	54.28 %	68.18 %	60 %

Regarding the number and length of breaks during the processing of the learning material and the task according to the participants' self-reports, no differences between the material types could be found (Kruskal-Wallis: Chi-square = 2.2299, $p = 0.6936$ for number of breaks and Chi-square = 2.0361, $p = 0.7291$ for length). There were also no significant differences between the material types with regard to the measured times needed for learning and solving the transfer task (Wallis-Kruskal: Chi-square = 4,5741, $p = 0.3339$ for learning time and Chi-square = 4,891, $p = 0.2987$ for solving time). Further, between the learning time and the success in the transfer task there is a slightly positive correlation (Spearman's rank correlation: $p = 0.024$, $r = 0.244$), which cannot be found in the total or solving time.

4.2 Evaluation and students' preferences

Table 4 shows the students' preferred learning material style when they were allowed to choose for the next topic. With 42.35 %, a large proportion of participants prefer the long video, where mistakes are accepted and the live coding process is shown as realistically as possible. The least popular first choice, with 3.53 %, was the finalized and commented code. The second favorite choices were finalized code that is explained in the video and code development in the Jupyter Notebook, with 29.41 % and 28.24 % respectively. For a combined preference with the weighting of 2/3 of the first choice and 1/3 of the second choice, the result is similar. The long live coding video is preferred the most, while finalized and commented code is less wanted. The edited video also performs a little worse compared to the other two types of material.

Table 4. Participants' preferences for learning material style.

	1. LC-video long	2. LC-video short	3. explanation video	4. Jupyter Notebook	5. explained code
First preference	42.35 %	15.29 %	17.65 %	18.82 %	3.53 %
Second preference	12.94 %	15.29 %	29.41 %	28.24 %	14.12 %
Combined (2/3 1st + 1/3 2nd)	32.55 %	15.29 %	21.57 %	21.96 %	7.06 %

Table 5 shows the means and standard deviations of the items difficulty, scope, comprehensibility and motivation, each of which was to be assessed by the participants on 5-level Likert scales. For the assessment of difficulty, no significant differences could be found between the material type groups. Regarding the scope, although there is a significant result by Kruskal-Wallis (Chi-square = 13.847, $p < 0.01$), no significant differences can be seen in the post-hoc analysis (Pairwise Wilcoxon rank sum test/bonferroni: $p > 0.05$). In terms of comprehensibility, the third material performs best, with significant differences to material 5 (Kruskal-Wallis: Chi-square = 10.57, $p < 0.01$, Pairwise Wilcoxon rank sum test/bonferroni: $p < 0.05$ for material 3 and 5, $p > 0.05$ for all other). Material 3 is rated as more comprehensible than the Jupyter Notebook with the development and the commented code, with an effect size of $f = 0.2461$.

Table 5. Students' assessments means and SD on a Likert-scale from 1 to 5 according to the item.

	All	1. LC-video long	2. LC-video short	3. explanation video	4. Jupyter Notebook	5. explained code
Difficulty (Very easy - very difficult)	3.78 (0.98)	3.42 (0.79)	3.81 (1.17)	3.77 (0.83)	4.05 (1.05)	3.68 (0.95)
Scope (Strongly too little extensive - strongly too extensive)	2.78 (1.09)	2.3 (0.67)	2.94 (0.93)	2.31 (1.12)	3.48 (1.15)	2.45 (1.09)
Comprehensibility (Very difficult - very easy to understand)	3.42 (0.94)	3.62 (1.12)	3.63 (0.96)	3.93 (0.73)	3.14 (0.94)	3.1 (0.79)
Motivation (Strongly demotivating - strongly motivating)	2.86 (0.89)	2.92 (0.76)	3 (0.82)	3.07 (0.92)	2.86 (1.04)	2.55 (0.83)

In addition to the closed questions with mostly Likert scales, two open questions were also included in the evaluation. First, the participants were asked if they found anything specific in the learning material to be disturbing and second, they were basically allowed to comment freely on the learning material and the transfer task. Four participants who had seen the long video as learning material explicitly stated in the comments that they perceived the mistake shown there as a learning opportunity. They reflected that although it might take longer to watch the video, the effect of the mistake would bring a correspondingly meaningful learning outcome. Only one participant experienced the length of this material as critical. One comment suggested copying less relevant parts of the live coding as code instead of typing to reduce the length.

Seven participants expressed a basic request for more side explanations, alternatives and for a more practical look at the problem beforehand without directly starting the coding. Three comments criticize the sound or picture quality in general. Another two comments refer to weaknesses in the type 3 (scrolling is not possible) and type 5 (too few explanations) materials, which is possible in other types of material in both cases.

The participants were also asked how long they could imagine learning continuously with this type of learning material. Five choices were given, ranging from "up to 10 minutes" to "over 90 minutes". These values were chosen in such a way that both short learning units with individual videos or learning materials would be possible as well as entire lectures, whose usual duration is about 90 minutes, could be replaced by such materials in the context of asynchronous digital teaching. Figure 2 shows the

students' answers. The students who were asked to estimate the possible learning time with the Jupyter Notebook gave the clearest answers. Most participants can imagine a learning period of 30 to 60 or even 90 minutes. For most participants with materials of types 2, 3 and 5, just 60 minutes are the acceptable maximum. Only for participants who have learned with the long video, 60 to 90 minutes of learning time are also possible to a large extent.

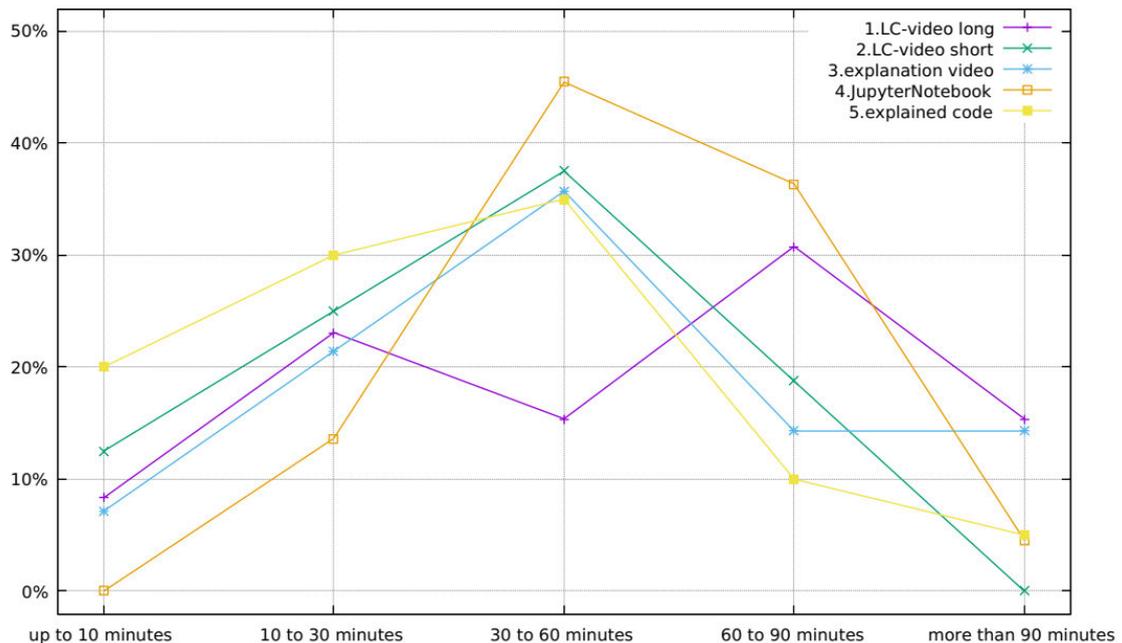


Figure 2. Students' answers for appropriate length for continuous learning with their material type (relative numbers).

5 DISCUSSION

First of all, it should be considered as a limitation that only one learning material with only one transfer task for a specific topic was explored in this study. For a generalisation of the results, further trials would be necessary. Additionally, it cannot be guaranteed that the full potential of every material type was reached, but this problem can occur in every used learning material and is thus more comparable to a realistic scenario. Nevertheless, especially through the reflection of some participants in the comments, there are comprehensible reasons why the supposedly less perfectionist video with errors is preferred by the participants. When looking at the composition of the participants, a slightly increased quota of women compared to the expected in the course is noticeable. In general, it appears probable that mainly more motivated students participated completely and thus motivational effects, e.g. due to shorter videos, cannot be measured in a reliable way. However, the motivation to participate was slightly moved by the gamification concept rewards in the course, so that presumably not only students interested in the topic itself participated, but also those who wanted to collect the points. Another limitation of the study is that only completely correct solutions in the transfer task were included in the evaluation. This prevents the identification of possible effects on learning that resulted in partially correct solutions. Furthermore, there was no analysis of which types of errors occurred frequently and whether these could be related to the material.

A positive observation for many lecturers who fear high effort when creating videos is that extensive post-production does not seem to be necessary for the learning success. On the contrary, many students perceive mistakes in the development of code as a learning opportunity. It can also give them a more realistic picture of the process, so that the barriers are not that high, especially for novice programmers. Due to less effort in post-production, often longer videos are created that also consciously address difficulties, but the production process as a whole is kept within limits. With less demand for perfectionism, fewer recording attempts are often needed and the production is simplified overall.

The measured times should be considered with caution, as access to the learning material was guaranteed during the entire time and clicking further early on can lead to distortions. When looking at the measured times itself, a causality of learning success to the learning time can rather be excluded. However, on the basis of similar learning times for videos and learning materials of different lengths, it is interesting that possibly shorter videos are viewed for a similar length of time. This would weaken the assumption that for longer videos more time has to be invested in principle. According to the students' assessment, they tend to prefer learning with the long video and Jupyter Notebook for a longer period of time. This could be explained with shorter learning materials continuously requiring a high level of attention. Missing a short period of time can already lead to a lot of missed content, whereas with longer materials, the content is discussed in more detail. Additionally, long videos in particular are closer to reality and thus also more authentic and possibly less demanding to watch. In Jupyter Notebooks, the variation between reading phases and trying out code could also help to maintain concentration more easily. Many participants asked for more background information, alternative solutions and considerations of the overall scenario. This can be realised most easily in the type 4 material without becoming too extensive.

Regarding the choice of the preferred material type, the question explicitly asks about the choice for a topic that is not known yet. It is possible that the answers could differ if, for example, it is clear that the topic is particularly easy or difficult, so that different types of material would be preferred for different difficulty levels.

At least for the scope of the study, the style of the learning material does not seem to be relevant for the success of transfer learning. However, the participants' assessments and preferences show that videos that illustrate actual development as well as possible are preferred. The better rated comprehensibility of the type 3 material could indicate a more superficial but concise understanding in the context of the scope. For interactive materials that are not videos, code development is clearly preferred to finalized commented code. On the one hand, this can probably be explained by the more detailed information, and on the other hand, step-by-step development could encourage more experimenting and modifying than ready-made code, which seems to be untouchable.

In relation to the results of previous research the study suggests that the design of learning media may have effects on learning transfer and learners' satisfaction, but in case of learning transfer for a quite difficult task the choice of how to design a code explanation material in terms of interactivity and development is less relevant.

6 SUMMARY AND CONCLUSIONS

In the presented study, different types of learning materials, which can be used in asynchronous digital learning scenarios, were compared using the example of graph algorithms. According to the results, the type of material does not seem to be of great importance for learning success. Nevertheless, learners prefer certain types of material. The development of code appears to be preferred, as noted in previous research. Some learners even explicitly recognize the learning opportunities in mistakes in the material when they are appropriately addressed. However, participants who have worked on the Jupyter Notebook with the code development are most likely considering a fundamentally longer learning period with their learning material. In this type of learning material content and code sections can be actively tried out and changed.

Although in occasional cases the sound or picture quality of the videos, which were all recorded with normal work equipment rather than in a professional setting, was criticized, the participants were generally rather tolerant. Lecturers should therefore not hesitate to provide students with learning materials that are not perfect. Mistakes should be explicitly addressed and discussed in order to use the opportunity to learn from them.

When choosing the type of material, the lecturer's own teaching personality should also be taken into account, as well as future use. If lecturers know that their materials will need to be modified frequently, it is more advisable to use easy-to-modify material types such as Jupyter Notebooks instead of videos. The increasing availability of open educational resources should also be taken into account. As a result, learning materials are already available for many topics. Although sufficient quality should be ensured here, the study results suggest that the actual design of the videos is only of minor importance for the actual learning success. Therefore, materials that do not correspond to the preferred type can also be used without concerns.

In the further research, it should be investigated which factors in the learner's personality and in the topic the might have an influence on learning and preferences. It may also be appropriate to offer combinations of different materials. Overall, more research is needed in the area of asynchronous learning materials with a focus on programming.

ACKNOWLEDGMENTS

The author Elisaweta Ossovski is an ELES research fellow.

REFERENCES

- [1] T. Jenkins, "On the Difficulty of Learning to Program," In 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences, 2002.
- [2] B. A. Becker and K. Quille, "50 Years of CS1 at SIGCSE: A Review of the Evolution of Introductory Programming Education Research," In: Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19). Association for Computing Machinery, New York, NY, USA, pp. 338–344, 2019. Retrieved from doi:10.1145/3287324.3287432
- [3] R. E. Clark, "Reconsidering Research on Learning from Media," *Review of Educational Research*, vol. 53, no. 4, pp. 445–459, 1983. Retrieved from <http://www.jstor.org/stable/1170217>
- [4] R. B. Kozma, "Will Media Influence Learning? Reframing the Debate," *Educational Technology Research and Development* vol. 42, no. 2, pp. 7–19, 1994. Retrieved from <http://www.jstor.org/stable/30218683>
- [5] O. Poquet, L. Lim, N. Mirriahi, and S. Dawson, "Video and Learning: A Systematic Review (2007-2017)," In Proceedings of the 8th International Conference on Learning Analytics and Knowledge (Sydney, New South Wales, Australia) (LAK '18), Association for Computing Machinery, New York, NY, USA, pp. 151–160, 2018. Retrieved from doi:10.1145/3170358.3170376
- [6] J. Sweller, "Implications of Cognitive Load Theory for Multimedia Learning," Cambridge University Press, pp. 19–30, 2005. Retrieved from doi:10.1017/CBO9780511816819.003
- [7] R. E. Mayer, "Multimedia Learning", 3 ed., Cambridge University Press, 2020. Retrieved from doi:10.1017/9781316941355
- [8] K. R. Butcher, "The Multimedia Principle", 2 ed., Cambridge University Press, pp. 174–205, 2014. Retrieved from doi:10.1017/CBO9781139547369.010
- [9] H. Altinpulluk, H. Kilinc, M. Firat, and O. Yumurtaci, "The influence of segmented and complete educational videos on the cognitive load, satisfaction, engagement, and academic achievement levels of learners," *Journal of Computers in Education*, vol. 7, 2019. Retrieved from doi:10.1007/s40692-019-00151-7
- [10] B. de Koning, H. Tabbers, R. Rikers, and F. Paas, "Towards a Framework for Attention Cueing in Instructional Animations: Guidelines for Research and Design," *Educational Psychology Review*, vol. 21, pp. 113–140, 2009. Retrieved from doi:10.1007/s10648-009-9098-7
- [11] J. Sweller, "The worked example effect and human cognition," *Learning and Instruction* vol. 16, pp. 165–169, 2006.
- [12] A. Renkl, "The Worked-Out Examples Principle in Multimedia Learning," Cambridge University Press, pp. 229–246, 2005. Retrieved from doi:10.1017/CBO9780511816819.016
- [13] S. Abdul-Rahman and B. Du Boulay, "Learning Programming via Worked-Examples: Relation of Learning Styles to Cognitive Load," *Comput. Hum. Behav.* vol. 30, pp. 286–298, 2014. Retrieved from doi:10.1016/j.chb.2013.09.007
- [14] M. Nainan and B. Balakrishnan. "Design and Evaluation of Worked Examples for Teaching and Learning Introductory Programming at Tertiary Level," *Malaysian Online Journal of Educational Technology (MOJET)*, vol. 7, pp. 30–44, 2019. Retrieved from doi:10.17220/mojet.2019.04.003
- [15] R. Zhi, T. Price, S. Marwan, A. Milliken, T. Barnes, and M. Chi, "Exploring the Impact of Worked Examples in a Novice Programming Environment," 2019. Retrieved from doi:10.1145/3287324.3287385

- [16] K. R. Koedinger, J. Kim, J. Zhuxin Jia, E. A. McLaughlin, and N. L. Bier. 2015, "Learning is Not a Spectator Sport: Doing is Better than Watching for Learning from a MOOC," In Proceedings of the Second ACM Conference on Learning@Scale (Vancouver, BC, Canada) (L@S '15), Association for Computing Machinery, New York, NY, USA, pp. 111–120. Retrieved from doi:10.1145/2724660.2724681
- [17] J. A. Bateman and F. Schmidt-Borcherding, "The Communicative Effectiveness of Education Videos: Towards an Empirically-Motivated Multimodal Account," In Multimodal Technologies and Interaction vol. 2, no. 3, 2018. Retrieved from doi:10.3390/mti2030059
- [18] J. Wang and P. D. Antonenko, "Instructor presence in instructional video: Effects on visual attention, recall, and perceived learning," In Computers in Human Behavior, vol. 71, pp. 79–89, 2017. Retrieved from doi:10.1016/j.chb.2017.01.049
- [19] T. Colliot and E. Jamet, "Understanding the effects of a teacher video on learning from a multimedia document: an eye-tracking study," In Educational Technology Research and Development, vol. 66, pp. 1415–1433, 2008.
- [20] F. Schmidt-Borcherding and L. Drendel, "Erklärvideos in der digitalen Hochschullehre: Welche Rolle spielen Sprecherpräsenz und Kohärenz?," In die hochschullehre vol. 7, 2021. Retrieved from doi:10.3278/HSL2108W
- [21] A. Cross, M. Bayyapureddy, E. Cutrell, A. Agarwal, and W. Thies, "TypeRighting: Combining the Benefits of Handwriting and Typeface in Online Educational Videos," In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Paris, France) (CHI '13), Association for Computing Machinery, New York, NY, USA, pp. 793–796, 2013. Retrieved from doi:10.1145/2470654.2470766
- [22] A. Ali, K. Samsudin, M. Hassan, and S. Firdaus Sidek, "Does Screencast Teaching Software Application Needs Narration For Effective Learning," In Turkish Online Journal of Educational Technology, vol. 10, pp. 76–82, 2011.
- [23] S. A. Lloyd and C. L. Robertson, "Screencast Tutorials Enhance Student Learning of Statistics," In Teaching of Psychology, vol. 39, no. 1, pp. 67–71, 2012. Retrieved from doi:10.1177/0098628311430640
- [24] D. K. Griffin, D. Mitchell, and S. J. Thompson. "Podcasting by synchronising PowerPoint and voice: What are the pedagogical benefits?," In Computers & Education, vol. 53, no. 2, pp. 532–539, 2009. Retrieved from doi:10.1016/j.compedu.2009.03.011
- [25] C. Ilioudi, M. Giannakos, and K. Chorianopoulos, "Investigating Differences among the Commonly Used Video Lecture Styles," 2013. Retrieved from doi:10.13140/2.1.3524.9284
- [26] P. J. Guo, J. Kim, and R. Rubin, "How Video Production Affects Student Engagement: An Empirical Study of MOOC Videos," In Proceedings of the First ACM Conference on Learning@Scale Conference (Atlanta, Georgia, USA) (L@S '14). Association for Computing Machinery, New York, NY, USA, pp. 41–50, 2014. Retrieved from doi:10.1145/2556325.2566239
- [27] S. H. Edwards, "Using Software Testing to Move Students from Trial-and-Error to Reflection-in-Action," In Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education (Norfolk, Virginia, USA) (SIGCSE'04), Association for Computing Machinery, New York, NY, USA, pp. 26–30, 2004. Retrieved from doi:10.1145/971300.971312
- [28] F. Oser, T. Hascher, and M. Spsychiger, "Lernen aus Fehlern Zur Psychologie des "negativen Wissens"," VS Verlag für Sozialwissenschaften, Wiesbaden, pp. 11–41, 1999. Retrieved from doi:10.1007/978-3-663-07878-4_1
- [29] F.Z. Weinert, "Aus Fehlern lernen und Fehler vermeiden lernen," VS Verlag für Sozialwissenschaften, Wiesbaden, pp. 101–109, 1999. Retrieved from doi:10.1007/978-3-663-07878-4_5

A.5 Beitrag 5

Übersicht zu Beitrag 5

Titel:	Frustration Tolerance among Computer-Science-related Novice University Students
Autoren:	Elisaweta Ossovski Michael Brinkmeier
Status:	eingereicht, "Revision Required"
Eingereicht bei:	European Journal of Engineering Education

Frustration Tolerance among Computer-Science-related Novice University Students

Elisaweta Ossovski^{1*} and Michael Brinkmeier¹

¹Institut für Informatik, Universität Osnabrück, Wachsbleiche 27,
49090 Osnabrück, Deutschland.

*Corresponding author(s). E-mail(s): eossovski@uos.de;
Contributing authors: mbrinkmeier@uos.de;

Abstract

Among students and lecturers of Computer Science (CS)-related subjects, there is often anecdotal talk that Frustration Tolerance (FT) is a requirement for success in CS studies or that students would learn it during their studies. In order to relate the anecdotal perception and the psychological concept of FT and to explore the anecdotal narratives, the Frustration Discomfort Scale (FDS) and self-assessment among other factors were conducted with students from a university CS1 course at two times. There seems to be no effect from one semester of CS on FT and no actual differences in prerequisites, although further studies will be needed. Students with extensive prior knowledge, male students and students in low CS-related majors rate their FT as higher than the opposite groups at the beginning of the course. Lecturers should reconsider the use of the term as it may support negative feelings related to Imposter Syndrome.

Keywords: FDS, frustration, Distress Tolerance, CS novices

1 Introduction

Among students and lecturers of Computer Science (CS)-related subjects, there is often mostly anecdotal talk that either Frustration Tolerance is a requirement for success in CS studies or that students of such subjects

2 Frustration Tolerance among Computer-Science-related Students

would learn Frustration Tolerance during their studies (Thompson, 2022). This matched the authors' experiences. In this context, difficult tasks and tasks perceived as nearly unsolvable would systematically cause frustration (Egan, 2005; Fraser, 2014). According to the narrative, students who master the course requirements are able to overcome the frustration. Frustration Tolerance is used here as a general language term. Individual studies show, that lecturers in STEM courses also consider Frustration Tolerance to be a necessary trait that students must have in order to be successful in their studies (Neumann, Pigge, & Heinze, 2017) and some of them even mention Frustration Tolerance among the competencies to be achieved through training in introductory CS courses (Pfeffer, 2019).

In psychological research, the term Frustration Tolerance also exists and is predominantly considered in the context of Distress Tolerance. According to Harrington (2007), Frustration Tolerance is a multidimensional concept. There are indications for positive correlations between Frustration Tolerance and (academic) achievement (Kaufmann & Ferguson-Lucas, 2020; McCormick, Tooke, Winston, & Kjellander, 1991; Meindl et al., 2018) and negative correlations between Frustration Tolerance and procrastination (Harrington, 2007; Uzun Ozer, Demir, & Harrington, 2012).

In Engineering, there is some evidence that students decide about their feeling of belonging to the subject depending on having the skills they think are required or not having them (Rohde, Benson, Potvin, Kirn, & Godwin, 2018). Considering Frustration Tolerance to be such a skill like stated by some interventions (Vidal et al., 2021), this could lead to a low feeling of belonging and possibly even to the problematic Imposter Syndrom.

The aim of this study is to relate the anecdotal perception and the psychological concept in order to explore a possible veracity of the anecdotal narratives and be able to determine a correct use of the term to avoid wrong self-perceptions. The following research questions were investigated:

- RQ1: Do beginners in CS-related subjects have a remarkable Frustration Tolerance in general or problem-related Frustration Tolerance and does it depend on the relation of their major to CS?
- RQ2: Does the measured or self-assessed Frustration Tolerance of students in general or for a specific scale change after attending a CS course?
- RQ3: Does the measured Frustration Tolerance align with students' self-assessment and how well does the general language term fit to the psychological concept?
- RQ4: Which influence do factors like prior knowledge, previous studied semesters or gender have on the measured or self-assessed Frustration Tolerance in general or problem-related Frustration Tolerance?
- RQ5: Does the Covid19-pandemic have an impact on the Frustration Tolerance?

2 Theoretical Background and Literature

Frustration Tolerance is a complex construct and is associated with cognitive functions and behaviour. It can be seen as a sub-area of Distress Tolerance (Zvolensky, Vujanovic, Bernstein, & Leyro, 2010) and is basis for the rationalemotive behaviour therapy in clinical contexts (Harrington, 2005).

Distress Tolerance encompasses a broader range of negative emotions or experiences that are endured. It can be either, in a passive way, the "perceived capacity to withstand negative emotional and/or other aversive states" (Leyro, Zvolensky, & Bernstein, 2010) or, in an active way, "the behavioural act of withstanding distressing internal states elicited by some type of stressor" (Leyro et al., 2010). Yang, Li, Nian, Xu, and He (2021) define Frustration Tolerance as something passive in comparison to active anti-frustration ability.

In a systematic literature review, Leyro et al. (2010) characterize self-report and behavioural tasks for Distress Tolerance, the constructs established in these measures, and related traits. Distress Tolerance is mostly reported to be related to anxiety and depression in self-report measures, to panic or negative mood in behavioural tasks and to substances use like smoking in both. It should be noted, that self-report means to answer questionnaires with appropriate items for the measured concept and not assessing the concepts as a whole for oneself. If and how Distress Tolerance is malleable is unclear in current research.

Andrés et al. (2021) conducted another systematic review of research concerning relationships between Distress Tolerance and executive functions like inhibitory control, working memory capacity or cognitive flexibility. They found studies for the tolerance of discomfort, negative emotions and uncertainty, but no studies were found concerning the relationship between executive functions and tolerance of frustration explicitly. Fillauer et al. (2019) explicitly used Frustration Tolerance measurements including the Frustration Discomfort Scale to investigate a relationship to working memory capacity. They did not found significant results, but argue for a need of further research.

Evanovich, Marshall, David, and Mumma (2019) described several instruments for self-reported and experimental measurement of Distress Tolerance. They estimate Distress Tolerance to be a complex construct and network of many factors that have not been fully studied. However, they rate the Frustration Discomfort Scale to be an appropriate self-report measurement.

Kaufmann and Ferguson-Lucas (2020) compared Discomfort Intolerance, Entitlement, Emotional Intolerance and Achievement Expectation between regular on-site students, students who studied online and video gamers. They choose gamers as the last group to control if online students could perceive their studies as a game challenge as it is often task-oriented. They found significant, but small differences concerning Discomfort Intolerance and Achievement Expectation. Video gamers had a lower expectation for achievement, while online students reported to be able to handle discomfort better than the other groups.

4 Frustration Tolerance among Computer-Science-related Students

Arguing for a relevance of Frustration Tolerance in CS, [Egelman et al. \(2020\)](#) analysed frustration caused by code review. They found, that negative emotions during the review of someone else's code cause frustration and conflicts. To remedy the problem, they hope to use CS education to prevent frustration and advocate further research using established measurement tools such as Harrington's FDS questionnaire ([Harrington, 2005](#)).

[Strnadová, Vobornik, and Cz \(2018\)](#) used a self developed questionnaire to investigate Stress Tolerance and also Frustration Tolerance among students at the Institute for CS and Management. For Frustration Tolerance, 18 items were answered on a dichotomous scale ("yes"/"no") to calculate a percentage score. They found that the majority of students are in the expected range, but did not examined dependencies or differences by the major. For students who did the survey at the beginning and at the end of a semester, they found the Frustration Tolerance to be lower at the end of the semester than at the beginning.

In addition, there is a number of studies that have examined certain contexts or specific relationships that may also have relevance to the research questions of this paper. For instance, in the context of supporting children with ADHD, it has been observed that Frustration Tolerance is higher when the Mirror Trace Frustration Task is performed with augmented reality ([Oçay, Rustia, & Palaoag, 2018](#)). It is possible that this finding could also be transferred to other digital-related areas.

Regarding the influence of the Covid19-pandemic, there is also some initial evidence for a role of Frustration Tolerance. Thereby, pre-pandemic behavior and Distress Tolerance were matched with behavior during the pandemic. It was found that the Distress Tolerance may have an effect on strategies that are applied to deal with the pandemic ([Sauer, Jungmann, & Witthöft, 2020](#)). The influence of the pandemic itself on distress or Frustration Tolerance was not investigated.

In an attempt to measure the Frustration Tolerance of mathematics students in comparison to students of other subjects, [Pouyamanesh and Firoozeh \(2013\)](#) measured and compared Failure Tolerance among students. The mathematics students showed a significantly higher Failure Tolerance and it was possible to deduce a connection to better performance from higher Failure Tolerance.

In some studies related to the sense of belonging, self-efficacy or Imposter Syndrome, questionnaires include items on frustration ([Laura K. Alford & DeOrio, 2020](#); [Settle, Lator, & Steinbach, 2015](#); [Wiggins, Grafsgaard, Boyer, Wiebe, & Lester, 2016](#)). The Imposter Syndrome refers to the feeling of not having actually performed according to the required stage of achievement ([Clance & Imes, 1978](#)). It appears especially among high-achieving women and under-represented groups ([Clance & Imes, 1978](#)). However, the academic self-concept in combination with racial identity showed to be a stronger predictor for the Imposter Syndrome than the racial identity itself ([Ewing, Richardson, James-Myers, & Russell, 1996](#)). Arguing for a special role of self-concepts in

general, this may be important for the self-concept of the students' Frustration Tolerance. For men, imposter feelings are related to ability-avoid goals, whereas women suffering from imposter feelings set ability-approach goals, but avoid task goals (Kumar & Jagacinski, 2006). These feelings can cause serious problems (Chrousos & Mentis, 2020).

In CS, the imposter feelings may be enhanced by the often low enrolment barriers, as several universities especially in the country where this study was conducted do not require specific grades or entrance tests. Some students may feel that they do not belong in the class, which can be also provoked by the expectation of Frustration Tolerance as a prerequisite. There is some general evidence concerning gender related differences in CS education. For example, women tend to rate their achievement as lower than men, although actual results suggest the opposite (Finlayson, 2020).

Overall, the concept of Frustration Tolerance is already explored outside of clinical research in certain contexts, but there are still many open questions. Less research deals with the professional implications and the relation to general language usage of the term, although frustration is partly investigated in the context of self-efficiency and Imposter Syndrome.

3 Methodology

In order to answer the research questions a questionnaire was conducted with CS-related novice university students. The collected data were analysed on the basis of these questions.

3.1 Questionnaire

To measure the Frustration Tolerance the authors decided to choose a self-report questionnaire, as the common tasks to measure Frustration Tolerance in an active way (Leyro et al., 2010; Meindl et al., 2018) seemed to be too different from usual tasks in CS. Questionnaires based on self-report were assessed to be closer to the self-assessment, which is relevant in general language usage. Additionally, extensive participation effort should be avoided in order to get a higher number of participants. Among other questionnaires on that topic the Frustration Discomfort Scale (FDS) by Harrington (2005) was chosen. In contrast to others, it has a widely developed underlying concept of Frustration Tolerance and is much more detailed. It consists of 35 items and four major subscales, one of which can be further divided into two scales. All items are answered on a 5-point Likert scale with 1 - absent, 2- mild, 3- moderate, 4 - strong and 5 - very strong. This results in the scales Discomfort Tolerance, Entitlement, Emotional Intolerance and Achievement summing up to a 4-factor total score, whereas splitting up Entitlement to Gratification and Fairness including additional items for each of this subscales leads to a 5-factor total score. A lower score means a higher Frustration Tolerance.

From a CS competencies point of view the most interesting subscale is the Achievement subscale. With items like "I can't stand doing a job if I'm unable

6 Frustration Tolerance among Computer-Science-related Students

to do it well” (Harrington, 2005) or ”I can’t stand feeling that I’m not on top of my work” (Harrington, 2005) it included the most similar Frustration Tolerance items related to the anecdotal one mentioned by students. Thus the Achievement scale was assumed to approach a problem-related Frustration Tolerance and to be the closest one to describe Frustration Tolerance from an academical achievement point of view. Additionally, there is some research on the effects of achievement goals among CS1-students (Zingaro et al., 2018).

The original scale is in English, but Spanish (Medrano, Franco, & Mustaca, 2018; Ruiz-Ortega, Sánchez-Álvarez, & Martos, 2021), French (Chamayou, Tsenova, Gonthier, Blatier, & Yahyaoui, 2015), Italian (Tripaldi et al., 2018), Urdu (Jibeen, 2013) and Turkish (Uzun Ozer et al., 2012) validated translations already exist. As there is no validated German version, the scale had to be translated. For the translation the TRAPD-method according to European Social Survey (2014); Harkness et al. (2003) was used. At first two independent translations were made, one of them by a bilingual advanced student of Cognitive Science with experience in translation and one by an author of this paper (T-translation). Afterwards, the differences were discussed with a third person, who is experienced in research (R-review) in order to elaborate a joint version for each item. This version was again discussed as a whole questionnaire (A-adjudication) and then went through a pre-test (P) with colleagues from the institute. In this phase only one item was adjusted slightly. The documentation (D) is available from the corresponding author upon request. As the authors are no psychologists, a validation from a psychological view was not intended, but the authors invite other researchers to use the translated items and collected data for further analysis.

For the self-assessment of Frustration Tolerance the participants were asked directly to rate their general and their problem-related Frustration Tolerance on a 5-point Likert scale. In addition, gender, age, semester, and prior knowledge in CS were surveyed. For the prior knowledge, a self-assessment was asked. The prior knowledge for the 6 thematically relevant categories CS in general, programming, C++ (the programming language used in the module), algorithms, logical thinking and pair programming/programming in small teams should be assessed on a 4-point Likert scale. In previous research in this module, this self-assessment has been observed to match the actual prior knowledge of the students.

The validations of the Frustration Discomfort Scale and some of the translations also include published values for mainly psychology students in their first terms of study. Although these vary between different languages, they can be used for an orientation (Harrington, 2005; Jibeen, 2013).

The evaluation of the quantitative data collected was carried out using statistical methods appropriate to the research questions, which are explained in the individual results presentations.

3.2 Conducting and participants

The collection of data took place at two times during the winter semester 2020/21 and for the last research question additionally at one time during the winter semester 2021/22 with students in a multidisciplinary CS1 module. The module is attended by about 400 to 550 students in every winter semester and mandatory for different study programmes including CS Bachelor, Cognitive Science Bachelor, different applied Informatics as well as Mathematics and teacher training in this subjects. This results in a high heterogeneity concerning not only the previous knowledge, as CS is not mandatory at schools where the study was conducted, but also concerning the goals of the students in the module. For those with highly CS-related majors the module is a main part of their first semester at the university. In contrast, for those, who study Mathematics teaching training or a Master in Environmental System Sciences, the module often seems to be unnecessary from the students' point of view. There is a gamification concept in the module, which includes rewards for the whole course for certain activities. Among those the students get rewards for participating in studies in order to improve teaching. This was used as an incentive in this study, although participation in the questionnaire was on a fully voluntary basis.

The study was conducted completely online via Limesurvey and was advertised via mail and in the learning-management-system. The first conducting time was in October 2020 at the beginning of the lecture period in their first CS module. The second conducting took place shortly before the first exam in February 2021. This times were chosen for RQ2, as the main learning outcomes should be reached at this point.

134 students participated in both questionnaires at the beginning and at the end of the semester, but 50 of them did not fill in at least one questionnaire completely and were thus removed from the analysis. 84 complete questionnaires for both conducting times were included in the analysis for this group.

After collecting the data with the first group in the winter semester 2020/21 and comparing them to the values of Harrington's validation, the question of whether the values in the German translation might have been influenced by the Covid19-pandemic conditions arose. The authors suggested that the pandemic situation may have contributed to higher general frustration, which had not yet been overcome especially during the winter and could lead to lowered Frustration Tolerance, if Frustration Tolerance can change that fast. This resulted in RQ5.

The problems with restrictive conditions were further intensified in the following winter semester. Scheduling of face-to-face-classes had to be limited again towards more online teaching during the semester. The students in the 2020/21 winter semester, on the other hand, were prepared for online teaching in advance and have only had a few negative experiences with it in comparison, since they have not yet experienced a full year of adapted teaching.

8 *Frustration Tolerance among Computer-Science-related Students*

The mood perceived by the instructors in online teaching also differed considerably from that in the previous year. Therefore, a moment of perceived high frustration was used to ask the students of the module in the winter semester 2021/22 to fill in the questionnaire for one point in time as well.

A generally lower Frustration Tolerance for this point in time could be a sign that the mood presumably influenced by the pandemic also has an effect on Frustration Tolerance and that the values from the first survey should be considered against this background.

This conducting took place similarly to the first group, but it was carried out at the time of perceived highest frustration at the midterm in December 2021, together with an midterm evaluation of the module. 89 students gave complete answers for the FDS scale and self-assessment of Frustration Tolerance.

As the conducting took place in a slightly different scenario as described above, the data of the second group was used only for RQ5.

4 Results

In this section the results for the five research questions are presented. Table 3 shows the means and standard deviations for the Frustration Discomfort Scale and its subscales as well as the self-assessment for the two conducting times for group 1 and for group 2. These general scores are relevant for all research questions.

4.1 Research Question 1: Frustration Tolerance in General and for CS-related students

The first research question aimed at the Frustration Tolerance of CS-related students in general, by measuring it with the FDS as well as by self-assessment, and to determine possible noticeable differences regarding the CS-relation of the participants' majors.

When looking at the results in general (see table 3) and comparing them to previously known and available values from the validations of the English and the Urdu version (see table 1), the values are between those found previously. Therefore, no noticeable findings for the Frustration Tolerance of CS-related students in general can be proven from these results.

For the second part three groups among the participants were analysed. The first group consists of students with a highly CS-related major (n=10), including all CS Bachelor degrees and Embedded Systems Bachelor. The second group (n=55) consists of students with a medium CS-related major. Besides students with a Cognitive Science major, this includes all applicational CS Bachelors, for example Business Informatics and some Master students of the same majors, who have to catch up CS after their application-related Bachelor. The third group (n=19) includes the students with rather low CS-relation, which are mainly Mathematics and Physics Bachelors.

Frustration Tolerance among Computer-Science-related Students 9**Table 1** Means and SD results for FDS and its subscales for students from English validation (Harrington, 2005) with $n = 79$ and Urdu validation (Jibeen, 2013) with $n = 332$ (RQ1)

Scale	English validation	Urdu validation
4-factor	75.7 (15.9)	not specified
1. Discomfort Tolerance	17.6 (4.7)	24.77 (4.9)
2. Entitlement	19.6 (4.4)	24.28 (5.41)
2a. Gratification	17.2 (4.1)	not specified
2b. Fairness	22.2 (5.1)	not specified
3. Emotional Intolerance	18.3 (5.1)	23.75 (5.01)
4. Achievement	20.2 (5.2)	24.87 (5.29)
5-factor	95.4 (19.4)	not specified

For these groups, Kruskal-Wallis rank sum tests were conducted to investigate differences concerning the FDS-measured Frustration Tolerance for both conducting times. The same was done for the the Achievement scale, the self-assessment concerning Frustration Tolerance in general and problem-related Frustration Tolerance. Additionally, the change between the two conducting times was analysed. The only two significant results can be found for the self-assessed general Frustration Tolerance. The results are shown in table 2. For both measured times, a significant difference between at least two groups can be assumed.

For this reason, pairwise Wilcoxon tests with Bonferroni correction were performed as post-hoc tests for this dependent variable. The results are presented in table 4. For the first time point, there are significant differences between the self-assessed general Frustration Tolerance of the group of low CS-related students and the other two groups. For the second time point, the difference is only significant between the group of students with low CS-relation to the group of students with high CS-relation.

The mean value for the self-assessed general Frustration Tolerance remained constant in the group with low CS relation, while the mean values of the other two groups increased. A higher value thus means a higher self-assessed Frustration Tolerance. At the first time point, the CS-relation allocation has a medium effect size, whereas the effect size at the second time point can be classified as weak.

Overall, the Frustration Tolerance measured with the FDS questionnaire does not show any remarkable results in the context of previous research. In a comparison of students with different CS-relations in their majors, students with higher CS-relation tend to consider themselves less frustration-tolerant at the beginning of the CS-module, but the differences become smaller at the end of the module.

4.2 Research Question 2: Change of Frustration Tolerance

To analyse if there are significant differences in Frustration Tolerance between the two measuring times, Wilcoxon signed rank tests were performed for all

10 *Frustration Tolerance among Computer-Science-related Students***Table 2** Results of Kruskal-Wallis rank sum tests for groups of different CS-related majors (group sizes: 10/55/19), FT = Frustration Tolerance, n = 84 , * = significant (RQ1)

Scale	beginning of semester	end of semester
4-factor FDS	$\chi^2 = 2.2019, p = 0.3326$	$\chi^2 = 1.3005, p = 0.5219$
Achievement	$\chi^2 = 0.18337, p = 0.9124$	$\chi^2 = 4.9115, p = 0.0858$
self-assessment		
general FT	$\chi^2 = 12.182, p = 0.0023*$	$\chi^2 = 6.5598, p = 0.0376*$
self-assessment		
problem-related FT	$\chi^2 = 2.9284, p = 0.2313$	$\chi^2 = 1.0192, p = 0.6007$
4-factor FDS difference		
between conducting times	$\chi^2 = 0.51634, p = 0.7725$	
Achievement difference		
between conducting times	$\chi^2 = 1.6685, p = 0.4342$	

FDS subscales as well as the two summarizing scales. These are suitable for non-parametric Likert scales and dependent samples.

As table 5 shows, only for one FDS subscale a significant difference was found between the two measured times October 2020 and February 2021, which were the beginning and the end of the lecture period. The mean for the fairness scale (2b) is significantly lower at the end of the semester with an effect size of $r = 0.2411$. No significant differences could be found in the other scales or in the self-assessment. For the Discomfort Tolerance subscale, there is a tendency for a higher mean at the end of the lecture period, which is not significant.

4.3 Research Question 3: Relation of FDS score to Self-Assessment

Since Frustration Tolerance is a psychological concept that is applied in the clinical field as well, research question 3 aimed to examine the extent to which the used scale corresponds to the students' general concept of Frustration Tolerance. For this purpose, the values of the FDS scale were correlated with the students' self-assessments. This involved determining the correlation between the 4-factor FDS score and the assessment of general Frustration Tolerance and also testing whether the Achievement subscale was related to the students' self-assessment of problem-related Frustration Tolerance. Spearman correlation was calculated due to the non-parametric Likert scales.

Table 6 shows the results. For the first conducting time, in October 2020, there is a highly significant correlation with a medium correlation coefficient between the self-assessed general frustration and 4-factor FDS scores. This means, that a higher self-assessed Frustration Tolerance correlated with a lower 4-factor FDS score corresponding in a higher Frustration Tolerance. For the second conducting time, there is no significant correlation between these scores, but the same tendency.

No significant correlation can be found between the self-assessed problem-related Frustration Tolerance and the Achievement subscale of the FDS questionnaire. However, there is a tendency for a positive correlation for the second conducting time, which means higher self-assessed problem-related

Table 3 Means and SD results for FDS and its subscales and self-assessment (5-point Likert scale, 1 for "very low" to 5 for "very high") of general and problem-related Frustration Tolerance (FT), $n = 84$ for group 1 and $n = 89$ for group 2

Scale	Group 1 (Oct. 2020)	Group 1 (Feb. 2021)	Group 2 (Dec. 2021)
4-factor	82.42 (12.62)	81.82 (14.89)	84.96 (13.1)
1. Discomfort Tolerance	18.39 (4.39)	19.04 (4.9)	20.75 (4.39)
2. Entitlement	20.46 (4.39)	19.94 (5.01)	19.85 (4.83)
2a. Gratification	19.71 (4.63)	19.46 (4.59)	19.48 (4.21)
2b. Fairness	23.56 (4.68)	22.75 (5.21)	23.06 (4.67)
3. Emotional Intolerance	21.45 (4.83)	21.02 (5.25)	21.88 (4.59)
4. Achievement	22.11 (4.27)	21.82 (4.3)	22.47 (3.95)
5-factor	105.22 (15.87)	104.1 (18.44)	107.64 (15.25)
self-assessment general FT	3.214 (0.8077)	3.369 (0.8033)	3.386 (0.8085)
self-assessment problem-related FT	3.3095 (0.7912)	3.4286 (0.7959)	3.3837 (0.8284)

Table 4 Post hoc Paired Wilcoxon test with Bonferroni correction for significant Kruskal-Wallis rank sum tests: CS-related groups differences concerning self-assessed general Frustration Tolerance, $n = 84$, * = significant (RQ1)

Time 1	High CSR	Medium CSR	Low CSR	Time 2	High CSR	Med. CSR	Low CSR
Med. CSR	0.23	-		Med. CSR	0.551	-	
Low CSR	0.01*	0.017*		Low CSR	0.035*	0.206	
mean	2.6	3.15	3.74	mean	3	3.31	3.74
effect size	$r = 0.3352$			effect size	$r = 0.1806$		

12 *Frustration Tolerance among Computer-Science-related Students***Table 5** Results of Wilcoxon signed rank tests for two measuring times October 2020 and February 2021, $n = 84$, * = significant (RQ2)

Scale	Wilcoxon test result
4-factor FDS	$V = 1689, p = 0.4592$
1. Discomfort Tolerance	$V = 1149.5, p = 0.0725$
2. Entitlement	$V = 1512.5, p = 0.178$
2a. Gratification	$V = 1474, p = 0.3685$
2b. Fairness	$V = 1706.5, p = 0.0271*$
3. Emotional Intolerance	$V = 1461, p = 0.2006$
4. Achievement	$V = 1455, p = 0.4273$
5-factor FDS	$V = 1852, p = 0.2665$
self-assessment general	$V = 306, p = 0.1371$
self-assessment problem-related	$V = 440.5, p = 0.356$

Table 6 Results of Spearman's rank correlation rho for FDS scale and self-assessment, FT= Frustration Tolerance, $n = 84$, * = significant (RQ3)

Spearman's correlation	October 2020	February 2021
self-assessment general FT	$S = 129275, p = 0.0043*$	$S = 116880, p = 0.095$
x 4-factor FDS	$\rho = -0.3088$	$\rho = -0.1834$
self-assessment problem-related FT	$S = 87532, p = 0.3027$	$S = 78681, p = 0.0635$
x Achievement	$\rho = 0.1138$	$\rho = 0.2034$

Frustration Tolerance may correlate with higher Achievement scores and thus lower Achievement-related Frustration Tolerance.

4.4 Research Question 4: Effects of Prior Knowledge, Previous Studies and Gender

In order to specify possible influencing factors for measured (FDS scale) and perceived (self-assessment) Frustration Tolerance, it was analysed if prior knowledge, previous studied semesters or gender may have an influence on those. For the first two factors, Spearman correlation was calculated due to the non-parametric Likert scales.

The results are shown in table 7. There is a significant correlation between prior knowledge and the self-assessed problem-related Frustration Tolerance for the first conducting time in October 2020. The low positive correlation coefficient means that higher prior knowledge is related to higher self-assessed problem-related Frustration Tolerance. However, for the second conducting time in February 2021, no correlation can be found.

Another significant correlation exists between the number of studied semesters and the Achievement subscale of the FDS score. The low positive correlation coefficient means that a higher number of studied semesters is related to a higher Achievement score and thus a lower achievement-related Frustration Tolerance.

To determine possible gender-based differences concerning measured or self-assessed Frustration Tolerance, Wilcoxon rank sum tests were conducted for the groups male and female students. 30 participants stated to be male, 54 to be female and no participant stated to be diverse.

Frustration Tolerance among Computer-Science-related Students 13

Table 7 Results of Spearman's rank correlation rho for effects of prior knowledge and previous studies, FT= Frustration Tolerance, n = 84, * = significant (RQ4)

Spearman's correlation	October 2020	February 2021
Prior Knowledge x 4-factor FDS	$S = 113596, p = 0.1729$ $\rho = -0.1501$	$S = 110470, p = 0.2832$ $\rho = -0.1185$
Prior Knowledge x Achievement	$S = 101450, p = 0.8065$ $\rho = -0.0271$	$S = 119350, p = 0.0572$ $\rho = -0.2084$
Prior Knowledge x self-assessment general FT	$S = 94262, p = 0.6801$ $\rho = 0.0456$	$S = 99719, p = 0.9309$ $\rho = -0.0096$
Prior Knowledge x self-assessment problem-related FT	$S = 72130, p = 0.0131*$ $\rho = 0.2697$	$S = 102771, p = 0.7145$ $\rho = -0.0405$
Number of studied semesters x 4-factor FDS	$S = 83197, p = 0.152$ $\rho = 0.1577$	$S = 77671, p = 0.051$ $\rho = 0.2136$
Number of studied semesters x Achievement	$S = 79797, p = 0.08$ $\rho = 0.1921$	$S = 74817, p = 0.0026*$ $\rho = 0.2425$
Number of studied semesters x self-assessment general FT	$S = 91034, p = 0.4789$ $\rho = 0.0783$	$S = 98456, p = 0.9771$ $\rho = 0.0032$
Number of studied semesters x self-assessment problem-related FT	$S = 94407, p = 0.6899$ $\rho = 0.0442$	$S = 95466, p = 0.7626$ $\rho = 0.0335$

Table 8 Results of Wilcoxon rank sum tests for male and female students, n = 84, * = significant (RQ4)

Scale	October 2020	February 2021
4-factor FDS	$W = 668, p = 0.1862$	$W = 768.5, p = 0.7018$
Achievement	$W = 785, p = 0.8186$	$W = 706, p = 0.3324$
self-assessment general FT	$W = 842, p = 0.7528$	$W = 996.5, p = 0.0627$
self-assessment problem-related FT	$W = 1048, p = 0.0173*$	$W = 941, p = 0.1896$

The results are shown in table 8. No significant differences could be found between male and female students in the FDS-measured Frustration Tolerance. However, there is a significant difference for these students in the self-assessed problem-related Frustration Tolerance at least for the first conducting time in October 2020. With means of 3.6 for male students and 3.1481 for female students on a 5-point Likert scale with 1 for a "very low" and 5 for a "very high" self-assessed Frustration Tolerance, there is a tendency for male students to assess their problem-related Frustration Tolerance as higher as the female students with an effect size of $r = 0.2596$. The same tendency can be seen in the self-assessment of the general Frustration Tolerance for the second conducting time in February 2021 with a mean of 3.6 for male and 3.2407 for female students, but this is not significant.

To summarize the results for research question 4, there are some significant correlations, but the most of them concerns only the self-assessment of Frustration Tolerance. Students with a higher prior knowledge and male students tend to assess themselves as more tolerant to frustration. Only the correlation of the number of studied semesters and the Achievement subscale of the FDS questionnaire concerns the measured Frustration Tolerance and indicates a lower achievement-related Frustration Tolerance for students of higher semesters.

14 *Frustration Tolerance among Computer-Science-related Students***4.5 Research Question 5: Effects of Covid19-pandemic**

As described in the methods section (3.2), the first data conducting took place during the Covid19-pandemic in a semester with completely digital teaching. Thus the authors considered that there may be an effect of these circumstances on the Frustration Tolerance of the students. Nevertheless, the students were prepared to get only digital help from the lecturers and had experienced only one previous digital semester, which was organized more spontaneous than the following in which the conducting took place. One year later, the lecturers had the impression that students had developed digital fatigue. One possible reason was, that these students had then experienced several digital semesters or home schooling. Another possible explanation was, that they had different expectations of their studies, which were initially announced to include more face-to-face teaching. Therefore, this group was also given the questionnaire, but at the time of perceived highest frustration at mid-term in December 2021.

The means for the subscales and self-assessment are shown in table 3. Because of the non-parametric Likert-scales, Wilcoxon rank sum tests were performed in order to compare the Frustration Tolerance scores for the groups.

Table 9 Results of Wilcoxon rank sum tests for , n = 84/85, * = significant (RQ5)

Scale/group 2	x group 1 October 2020	x group 1 February 2021
4-factor FDS	$W = 3339.5, p = 0.2265$	$W = 3172, p = 0.0858$
1. Discomfort Tolerance	$W = 2545.5, p = 0.0003*$	$W = 2858.5, p = 0.0075*$
2. Entitlement	$W = 4018, p = 0.3949$	$W = 3773, p = 0.9163$
2a. Gratification	$W = 3816.5, p = 0.8123$	$W = 3678, p = 0.8562$
2b. Fairness	$W = 4066, p = 0.3185$	$W = 3710, p = 0.9333$
3. Emotional Intolerance	$W = 3507, p = 0.4828$	$W = 3313, p = 0.1962$
4. Achievement	$W = 3509, p = 0.4864$	$W = 3376, p = 0.2708$
self-assessment		
general FT	$W = 3316, p = 0.2116$	$W = 3655.5, p = 0.8952$
self-assessment		
problem-related FT	$W = 3480.5, p = 0.6622$	$W = 3710, p = 0.7446$

Looking at the results shown in table 9, only the means for the Discomfort Tolerance subscale differ significantly with a higher mean and thus a lower Discomfort Tolerance for group 2. Thus it can be assumed, that the pandemic may have an impact on the Frustration Tolerance. However, this is significant only for a subscale which is not relevant for the main research questions in this paper, as there are no differences for the whole FDS score, the Achievement score and the self-assessment.

5 Limitations and Discussion

One limitation of the study is that all students participated in the survey during the Covid19-pandemic and thus experienced a different studying routine than students in CS-related subjects in general. However, as can be seen from the results for research question 5, the pandemic situation seems to have an

Frustration Tolerance among Computer-Science-related Students 15

impact primarily on only one subscale, which tends to have less influence on the subject of the study.

Due to the high heterogeneity of students in the course, it is possible to examine differences between them, but the number of students per property decreases considerably as a result. In particular in research question 1, where the group of highly CS-related students consists of only 10 students. On the other hand, when looking at the overall results for the respondents, it is difficult to discuss possible causal relationships taking this heterogeneity in consideration. In addition, general bias effects due to the voluntary nature of participation and the incentive applied should be taken into account. Even if the topic was not explained before filling out the questionnaire, it can be assumed that primarily motivated students were reached.

The participation rate is in line with the usual participation rates in the course. Due to the voluntary nature, the two measurement points, and the exclusion of incomplete responses, data are only available for a part of the students in the module. While there may be a bias due to the incentive used, which tends to attract more motivated students, the distribution of majors and prior knowledge does not show any unusual characteristics that might suggest that a bias in the results should be expected.

Comparing the sample to previous results on FDS is also difficult, as shifts due to translations have already occurred. For example, compared to the validation of the original English questionnaire (Harrington, 2005), the results seem to indicate an unusually low Frustration Tolerance. In contrast, when compared to the translation in Urdu (Jibeen, 2013), the scores appear to indicate a higher Frustration Tolerance. For the classification of the students as a whole group of CS-related students a comparison with the same translation under mostly the same conditions therefore remains necessary.

Besides that, the previous publications, which targeted students, mainly focused on connecting the concept of Frustration Tolerance with other psychological concepts. Thereby, constructs that are relevant for learning should be found in order to facilitate improvements. In this paper, in contrast, the focus is rather on exploring factors influencing Frustration Tolerance and on clarifying the connection to the field of study that has so far appeared independently of the psychological concept.

Although the results show that no subject-related differences are detectable in the Frustration Tolerance measured with the FDS questionnaire, the self-assessment of the students and further analyses revealed several interesting correlations.

Considering the correlation between the self-assessed Frustration Tolerance and the Frustration Tolerance measured with the FDS questionnaire, which is significant at least for the first measuring time, it is likely that the psychological concept in general has a connection to the students' perspective on Frustration Tolerance.

16 *Frustration Tolerance among Computer-Science-related Students*

It becomes clear that the Achievement scale and the self-assessment of problem-related Frustration Tolerance, which were assumed to be relevant, seem to be less important.

The general Frustration Tolerance appears to be related to CS, especially in the self-assessment of the students. It is noticeable that students with a high level of prior knowledge consider themselves to be more frustration-tolerant at the beginning of the semester. By the end of the lecture period, this difference is no longer apparent. The same effect can be found for male students, who also rate their Frustration Tolerance higher than the female students at the beginning, but not at the end of the lecture period. This indicates that at the beginning there is an overestimation, possibly due to the expectations of lecturers or students themselves, which becomes more realistic towards the end of the course. This reality could have been developed through actual experiences of frustration or through contact with other students and the matching of expectations and abilities.

The opposite was found for highly CS-related students, who rated themselves as less frustration tolerant compared to students with lower CS-related majors. Again, false expectations could be a factor, possibly involving a comparison to an ideal image of a CS student and devaluation due to awareness of one's own limitations.

Another result is the positive correlation between the previous studied semesters and the Achievement FDS subscale, which means that students from higher semesters tend to a lower measured achievement-related Frustration Tolerance. From the anecdotal narrative, the opposite would be more likely. The longer students had already been confronted with the challenges of their studies, the higher their Frustration Tolerance would be in response. However, in order to interpret the result, it should be considered who exactly the students with a higher semester are, who took the module in which the survey took place. First, there are repeaters who have already taken the module once but did not pass the exam. In this case, it can be assumed that the anecdotal narrative has not yet led to success and thus not to a higher Frustration Tolerance, but more likely to a higher frustration. On the other hand, they are more often students from majors with a rather low CS relation, since students with a strong CS-relation are required to take the module in their first semester. Even if there are no significant differences between the groups with different levels of CS relatedness in the FDS-measured Frustration Tolerance, this could have a measurable effect in conjunction with the repetition of the module.

To assess the effect of one CS module on the students' Frustration Tolerance the questionnaire was conducted at the beginning and the end of the lecture period. Except for the fairness scale, there were no significant changes. It can be assumed that the Frustration Tolerance of students can therefore either hardly be changed significantly at all by studying CS or that this cannot be achieved within one module. Against this background, the development

Frustration Tolerance among Computer-Science-related Students 17

of Frustration Tolerance as a competence goal should be reconsidered and, if necessary, specified.

For the fairness scale the result shows a slightly lower mean and thus a slightly higher fairness-related Frustration Tolerance. This could have resulted from experiences during the semester, when working with fellow students was perceived as unfair, but could be solved in the end. The survey took place shortly before the first exam, so most students already knew that they had at least mastered the admission to the exam.

Overall, it can be concluded that general Frustration Tolerance does not change as a result of taking a computer science module. This supports the assumption that general Frustration Tolerance is a psychological trait and thus not easily modifiable. Nevertheless, there are no differences in the measured Frustration Tolerance between the different groups considered. It can therefore be concluded that general Frustration Tolerance is not a mandatory prerequisite for success in computer science studies. This is also supported by the fact that the Achievement scale shows no significant correlation to the self-assessment of problem-related Frustration Tolerance. Due to the origin of the questionnaire, it can be assumed that only in the case of mental disorder a particularly low Frustration Tolerance would have relevance.

However, if lecturers declare Frustration Tolerance as a prerequisite, they possibly create a false expectation among the groups that consider themselves to be rather less frustration tolerant. In the study, this primarily affected women and students with little or no prior knowledge. Similar to the research on Imposter Syndrome, this may lead to a lower sense of belonging and lower academic success if students believe they do not fit in the class. Being able to cope well with arising difficulties can be helpful for study CS, but this should not be assumed to be a prerequisite. A similar situation deals with prior knowledge, which is generally helpful but not a prerequisite. Otherwise, there is a risk that students in certain groups for this will tend to feel they do not fit the course due to their lower self-assessment and may develop Imposter Syndrome.

6 Implications

Through the study, an approach to the relevance of Frustration Tolerance for students of CS-related subjects could be achieved. It was found that there are connections between the general language term and the psychological concept, but these need to be explored further. In particular, it has been noticed that some students' self-assessment may align with expectations of them. For instance, students who already have prior knowledge tend to have an initially presumably overestimated self-assessment of their Frustration Tolerance, which is relativized by the end of the semester. Students who study highly CS-related majors and women tend to rate themselves as less tolerant to frustration on the opposite.

18 *Frustration Tolerance among Computer-Science-related Students*

In addition to the general conclusion that further research is needed in this area, practical conclusions also follow from the results. Lecturers should be clear about the concept of Frustration Tolerance if they consider this to be a prerequisite or a target for CS studies in order to be able to optimize course guidance and to give students a suitable picture of CS against this background. Additionally, they should be aware of mentioning Frustration Tolerance as a prerequisite for success in CS studies, as the study results show that groups at risk for Imposter Syndrome may rate themselves as lower tolerant to frustration, although their real Frustration Tolerance is not in the range of concern.

Taking into account that preliminary studies associate actual CS practice with Frustration Tolerance as well, this term should be related more concretely to challenges in CS and supported appropriately. Instead of aiming at a problematic challenge through difficult tasks per se, examples like code review for fellow students can be taken up here to build up a topic-related and thus probably necessary Frustration Tolerance.

Concerning the psychological concept of Frustration Tolerance in an academic context, there are still several open questions, which should be further investigated also in general psychological and educational research. Research should be conducted to understand which factors are influenced by a particular Frustration Tolerance and which factors can influence Frustration Tolerance. For example, it should be investigated further if a high Frustration Tolerance leads to an improvement in academic success and how the Frustration Tolerance can be enhanced. The impact of changing circumstances, such as those caused by the Covid19-pandemic, on Frustration Tolerance should also be further investigated.

More research is needed to strengthen the results from this study. Further research should deal with the parts of Frustration Tolerance which may really be important for success in CS, how these can be supported and how lecturers can avoid students to get negative self-concepts because of not suitable prerequisite expectations.

The data conducted and analysed in this study are available from the corresponding author upon request, as the authors want to encourage other researchers to use it for further investigations concerning Frustration Tolerance among students.

References

- Andrés, M.L., del Valle, M., Richaud, M., Introzzi, I., Canet Juric, L., Navarro-Guzmán, J. (2021, 07). Distress tolerance and executive functions: A systematic review. *Psychology and Neuroscience*, 14. Retrieved from <https://doi.org/10.1037/pne0000259>
- Chamayou, J., Tsenova, V., Gonthier, C., Blatier, C., Yahyaoui, A. (2015, 09). [french validation of the frustration discomfort scale]. *L'Encephale*, 42.

Frustration Tolerance among Computer-Science-related Students 19

Retrieved from <https://doi.org/10.1016/j.encep.2015.08.005>

Chrousos, G., & Mentis, A.-F.A. (2020, 02). Imposter syndrome threatens diversity. *Science*, *367*, 749.2-750.

10.1126/science.aba8039

Clance, P.R., & Imes, S.A. (1978). The imposter phenomenon in high achieving women: Dynamics and therapeutic intervention. *Psychotherapy*, *15*, 241-247.

Egan, M.A.L. (2005, feb). Students with asperger's syndrome in the cs classroom. *SIGCSE Bull.*, *37*(1), 27-30. Retrieved from <https://doi.org/10.1145/1047124.1047369>

10.1145/1047124.1047369

Egelman, C.D., Murphy-Hill, E., Kammer, E., Hodges, M.M., Green, C., Jaspán, C., Lin, J. (2020). Predicting developers' negative feelings about code review. *Proceedings of the acm/ieee 42nd international conference on software engineering* (p. 174185). New York, NY, USA: Association for Computing Machinery. Retrieved from <https://doi.org/10.1145/3377811.3380414>

European Social Survey (2014, March). *Ess round 7 translation guidelines*. http://www.europeansocialsurvey.org/docs/round7/methods/ESS7_translation_guidelines.pdf. (London: ESS ERIC Headquarters, Centre for Comparative Social Surveys, City University London, last accessed on 15.02.2022)

Evanovich, E., Marshall, A., David, S., Mumma, G. (2019, 07). A network conceptualization of the multiple facets of distress tolerance. *Anxiety, Stress, & Coping*, *32*, 1-16. Retrieved from <https://doi.org/10.1080/10615806.2019.1641799>

Ewing, K.M., Richardson, T.Q., James-Myers, L., Russell, R.K. (1996). The relationship between racial identity attitudes, worldview, and african american graduate students' experience of the imposter phenomenon. *Journal of Black Psychology*, *22*(1), 53-66.

10.1177/00957984960221005

20 *Frustration Tolerance among Computer-Science-related Students*

Fillauer, J., Bolden, J., Jacobson, M., Partlow, B., Benavides, A., Shultz, J. (2019, 07). Examining the effects of frustration on working memory capacity. *Applied Cognitive Psychology*, 34. Retrieved from <https://doi.org/10.1002/acp.3587>

Finlayson, I. (2020, oct). The effect of gender on student self-assessment in introductory computer science classes. *J. Comput. Sci. Coll.*, 36(3), 102110.

Fraser, R. (2014). Collaboration, collusion and plagiarism in computer science coursework. *Informatics in Education*, 13(2), 179–195.

10.15388/infedu.2014.10

Harkness, J., van de Vijver, F., Vijver, A., Mohler, P., Sons, J.W., Zentrum fur Umfragen, m.u.A. (2003). *Cross-cultural survey methods*. Wiley.

Harrington, N. (2005, 09). The frustration discomfort scale: development and psychometric properties. *Clinical Psychology & Psychotherapy*, 12, 374 - 387. Retrieved from <https://doi.org/10.1002/cpp.465>

Harrington, N. (2007, 07). Frustration intolerance as a multidimensional concept. *Journal of Rational-Emotive & Cognitive-Behavior Therapy*, 25, 191-211. Retrieved from <https://doi.org/10.1007/s10942-006-0051-0>

Jibeen, T. (2013, 03). Frustration intolerance beliefs as predictors of emotional problems in university undergraduates. *Journal of Rational-Emotive & Cognitive-Behavior Therapy*, 31. Retrieved from <https://doi.org/10.1007/s10942-012-0154-8>

Kaufmann, D., & Ferguson-Lucas, T. (2020, 01). Using frustration awareness to assist counseling students in succeeding with their online curriculum. *Journal of Instructional Research*, 9, 24 - 33.

Kumar, S., & Jagacinski, C.M. (2006). Imposters have goals too: The imposter phenomenon and its relationship to achievement goal theory. *Personality and Individual Differences*, 40(1), 147-157. Retrieved from <https://www.sciencedirect.com/science/article/pii/S0191886905002333>

<https://doi.org/10.1016/j.paid.2005.05.014>

Frustration Tolerance among Computer-Science-related Students 21

- Laura K. Alford, A.K., & DeOrío, A. (2020, June). Student sense of community through an introductory computer programming course sequence. *2020 asee virtual annual conference content access*. Virtual On line: ASEE Conferences. (<https://peer.asee.org/35240>)
- Leyro, T., Zvolensky, M., Bernstein, A. (2010, 07). Distress tolerance and psychopathological symptoms and disorders: A review of the empirical literature among adults. *Psychological bulletin*, *136*, 576 - 600. Retrieved from <https://doi.org/10.1037/a0019712>
- McCormick, N., Tooke, W., Winston, S., Kjellander, C. (1991, 06). Ret in the college classroom. *Journal of Rational-Emotive and Cognitive-Behavior Therapy*, *9*, 95-111. Retrieved from <https://doi.org/10.1007/BF01062113>
- 10.1007/BF01062113
- Medrano, L., Franco, P., Mustaca, A. (2018, 01). Argentinean adaptation of the frustration intolerance scale. *Behavioral Psychology/ Psicología Conductual*, *26*, 303-321.
- Meindl, P., Yu, A., Galla, B., Quirk, A., Haeck, C., Goyer, P., ... Duckworth, A. (2018, 11). A brief behavioral measure of frustration tolerance predicts academic achievement immediately and two years later. *Emotion*, *19*. Retrieved from <https://doi.org/10.1037/emo0000492>
- Neumann, I., Pigge, C., Heinze, A. (2017). Mathematische lernvoraussetzungen für mint-studiengänge aus sicht der hochschulen - eine empirische studie mit hochschullehrenden. *Mitteilungen der Deutschen Mathematiker-Vereinigung*, *25*, 240 - 244. Retrieved from <https://doi.org/10.1515/dmvm-2017-0070>
- Ocay, A.B., Rustia, R.A., Palaoag, T.D. (2018). Utilizing augmented reality in improving the frustration tolerance of adhd learners: An experimental study. *Proceedings of the 2nd international conference on digital technology in education* (p. 58-63). New York, NY, USA: Association for Computing Machinery. Retrieved from <https://doi.org/10.1145/3284497.3284499>
- Pfeffer, M. (2019). It-kernkompetenzen im bachelorstudiengang „informationswissenschaften“ an der hochschule der medien stuttgart. *Bibliothek Forschung und Praxis*, *43*(2), 266-274. Retrieved from <https://doi.org/10.1515/bfp-2019-2052>

22 *Frustration Tolerance among Computer-Science-related Students*

- Pouyamanesh, J., & Firoozeh, L. (2013, 07). Compared the learning outcomes of students with math in high and low frustration tolerance. *Procedia - Social and Behavioral Sciences*, *84*, 837 - 840. Retrieved from <https://doi.org/10.1016/j.sbspro.2013.06.657>
- Rohde, J., Benson, L., Potvin, G., Kirn, A., Godwin, A. (2018, June). You either have it or you dont: First year engineering students experiences of belongingness. *2018 asee annual conference & exposition*. Salt Lake City, Utah: ASEE Conferences. (<https://peer.asee.org/31320>)
- Ruiz-Ortega, A., Sánchez-Álvarez, N., Martos, M. (2021, 08). Chilean validation of the frustration discomfort scale: relation between intolerance to frustration and discomfort and emotional intelligence. *Current Psychology*. Retrieved from <https://doi.org/10.1007/s12144-021-02135-2>
- Sauer, K., Jungmann, S., Witthöft, M. (2020, 10). Emotional and behavioral consequences of the covid-19 pandemic: The role of health anxiety, intolerance of uncertainty, and distress (in)tolerance. *International Journal of Environmental Research and Public Health*, *17*, 1-18. Retrieved from <https://doi.org/10.3390/ijerph17197241>
- Settle, A., Lalor, J., Steinbach, T. (2015). Reconsidering the impact of cs1 on novice attitudes. *Proceedings of the 46th acm technical symposium on computer science education* (p. 229234). New York, NY, USA: Association for Computing Machinery. Retrieved from <https://doi.org/10.1145/2676723.2677235>
- Strnadová, V., Vobornik, P., Cz, P. (2018, 11). Statistical processing of correlations in the field of handling stress among university students. (p. 89-99).
- Thompson, C. (2022, 05). *Programming isnt hard but its frustrating*. <https://levelup.gitconnected.com/programming-isnt-hard-but-it-s-frustrating-6cb740085243>. (last accessed on 11.10.2022)
- Tripaldi, S., Papparuso, M., Amabili, M., Manfredi, C., Caselli, G., Scarinci, A., ... Mezzaluna, C. (2018, 09). Frustration discomfort scale (fds). a psychometric study of the italian version. *Journal of Rational-Emotive & Cognitive-Behavior Therapy*, *36*. Retrieved from <https://doi.org/10.1007/s10942-018-0286-6>

Frustration Tolerance among Computer-Science-related Students 23

- Uzun Ozer, B., Demir, A., Harrington, N. (2012, 08). Psychometric properties of frustration discomfort scale in a turkish sample. *Psychological reports*, 111, 117-28. Retrieved from <https://doi.org/10.2466/08.02.18.PR0.111.4.117-128>
- Vidal, M., Maldonado-Flores, J., Bracamonte, T., Miranda, F., Labarca, A., Simmonds, J. (2021, 10). Nias pro: an initiative to educate, inspire and empower women. *Proceedings xiii congress of latin american women in computing 2021*. CEUR Workshop Proceedings. <http://ceur-ws.org/Vol-3000/paper93.pdf>.
- Wiggins, J., Grafsgaard, J., Boyer, K., Wiebe, E., Lester, J. (2016, 02). Do you think you can? the influence of student self-efficacy on the effectiveness of tutorial dialogue for computer science. *International Journal of Artificial Intelligence in Education*, 27.
10.1007/s40593-015-0091-7
- Yang, X., Li, X., Nian, D., Xu, J., He, H. (2021, 06). Development and psychometric analysis of the anti-frustration ability scale for primary and secondary school students. *Current Psychology*. Retrieved from <https://doi.org/10.1007/s12144-021-01913-2>
- Zingaro, D., Craig, M., Porter, L., Becker, B.A., Cao, Y., Conrad, P., ... Thota, N. (2018). Achievement goals in cs1: Replication and extension. *Proceedings of the 49th acm technical symposium on computer science education* (p. 687-692). New York, NY, USA: Association for Computing Machinery. Retrieved from <https://doi.org/10.1145/3159450.3159452>
- Zvolensky, M., Vujanovic, A., Bernstein, A., Leyro, T. (2010, 12). Distress tolerance: Theory, measurement, and relations to psychopathology. *Current Directions in Psychological Science - CURR DIRECTIONS PSYCHOL SCI*, 19, 406-410. Retrieved from <https://doi.org/10.1177/0963721410388642>

Erklärung über die Eigenständigkeit der erbrachten wissenschaftlichen Leistung

Ich erkläre hiermit, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet.

Bei der Auswahl und Auswertung folgenden Materials haben mir die nachstehend aufgeführten Personen in der jeweils beschriebenen Weise entgeltlich / unentgeltlich geholfen.

1. Bei den Publikationen haben, insbesondere bei Konzeptionierung und Durchführung, die aufgeführten Co-Autoren Anteil an den Arbeiten gehabt. Die genauen Anteile sind in Kapitel 3.2 auf den Seiten 34 bis 36 erläutert.
2. Beim Korrekturlesen dieser Arbeit haben mich Fiona Kazarovytska und Agnes Mainka unterstützt.

Weitere Personen waren an der inhaltlichen materiellen Erstellung der vorliegenden Arbeit nicht beteiligt. Insbesondere habe ich hierfür nicht die entgeltliche Hilfe von Vermittlungs- bzw. Beratungsdiensten (Promotionsberater oder andere Personen) in Anspruch genommen. Niemand hat von mir unmittelbar oder mittelbar geldwerte Leistungen für Arbeiten erhalten, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen.

Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form einer anderen Prüfungsbehörde vorgelegt.

Osnabrück, den 27. März 2023

(Elisaweta Ossovski)