
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 Koepe et al. (2015) und Koepe 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 Abgabeaufgaben 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 Abgabeaufgaben 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 Abgabearbeiten 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 Abgabearbeiten integriert und stärker auf diese abgestimmt. Häufig stellen die Abgabearbeiten 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 Abgabebefragungen 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 Abgabepflichten 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.

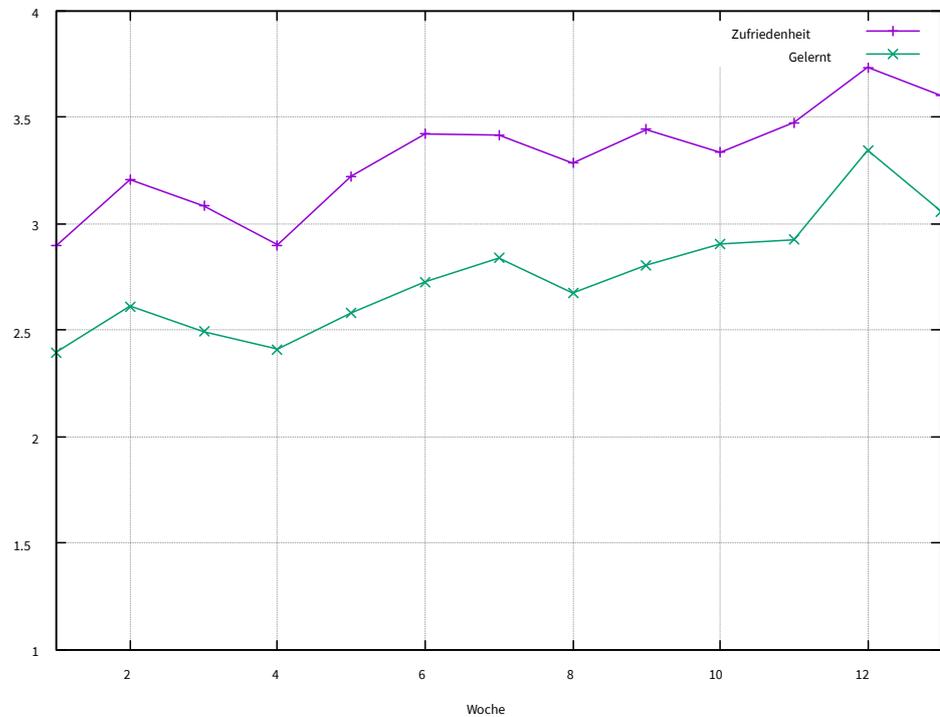


Abb. 2: Verlauf der Mittelwerte der Items «Zufriedenheit» und «Gelernt» in der wöchentlichen Coding-Class-Evaluation im Wintersemester 2019/20 (erstes Jahr der Umstellung), Skala von 1 für «gar nichts [gelernt]» bis 4 für «viel [gelernt]» bzw. 1 für «gar nicht [zufrieden]» bis 4 für «voll und ganz [zufrieden]»,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

Bei den Abgabebenen 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 Abgaberythmen 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 Programmierelemente 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 Abgabebefragungen 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 Abgabenaufgaben 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 Abgabearbeiten 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>.
- Osovski, 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.