

Herausforderungen und Angebote für heterogene Kohorten in der Softwareentwicklung

Veronika Thurner, Axel Böttcher¹

Abstract: In unseren Erstsemester-Kohorten werden die Studierenden immer heterogener, in verschiedenen Dimensionen. Wenn wir möglichst vielen der Studienanfänger:innen zu einem erfolgreichen Bildungsabschluss in der Informatik verhelfen wollen müssen wir unsere Lehr-Lernangebote so weiterentwickeln, dass sie den unterschiedlichen Bedürfnissen einer heterogenen Kohorte Rechnung tragen. Wir stellen eine Reihe von Ansätzen und Maßnahmen vor, die sich dafür in der Grundlagenausbildung zu Softwareentwicklung bewährt haben.

Keywords: Programmierausbildung, Heterogenität, Individuelle Lernpfade

1 Ausgangsbasis, Motivation und Ziele

In unseren Erstsemester-Kohorten werden die Studierenden immer heterogener, in verschiedenen Dimensionen. Während in den 1960-er Jahren noch weniger als 10% der Schüler:innen eines Jahrgangs eine Studienberechtigung erworben haben², lag der Anteil Mitte der 1990-er Jahre bereits bei rund 35% und pendelt sich seit 2010 auf etwa 50% ein³. Die Übergangsquote der Studienberechtigten ins Studium liegt dabei seit Mitte der 1990-er Jahre stabil zwischen 70% und 80%⁴. Heute nimmt also ein zunehmend breiterer Querschnitt der Bevölkerung ein Hochschulstudium auf.

Gleichzeitig wird migrationsbedingt auch die Bevölkerung selbst zunehmend kulturell vielfältiger. Des Weiteren wachsen die Möglichkeiten der weltweiten Bildungsmobilität. All diese Faktoren (und noch viele mehr) tragen dazu bei, dass wir in unserer Lehre auf Kohorten von Studierenden treffen, die auf sehr unterschiedliche persönliche Bildungswege zurückblicken und durch vielfältige soziale und kulturelle Herkunft geprägt sind.

¹ Hochschule München, Fakultät für Informatik und Mathematik, Lothstraße 64, D-80335 München, vorname.nachname@hm.edu

Gefördert durch die Stiftung Innovation in der Hochschullehre (StiL), <https://stiftung-hochschullehre.de/FoerdervertragFRFMM-455/2022>.

² https://www.destatis.de/DE/Presse/Pressemitteilungen/2023/06/PD23_N036_12.html, Pressemitteilung Nr. N036 vom 15. Juni 2023, Abbildung Studienberechtigtenquote

³ <https://www.bildungsbericht.de/de/bildungsberichte-seit-2006/bildungsbericht-2022/pdf-dateien-2022/bildungsbericht-2022-kapitel-f.pdf>, Bildung in Deutschland, Hrsg.: Autor:innengruppe Bildungsberichterstattung, wbv Media (Link), Bielefeld 2022, Abbildung F2-1 Studienberechtigte

⁴ ebenda, Abbildung F2-2 Übergangsquoten

Verschiedene Studien haben sich mit Heterogenität und Diversität in der Informatik-Ausbildung befasst (siehe dazu z. B. mehrere Beiträge im Tagungsband [De23]). Die Bandbreite der Untersuchungen reicht von der Betrachtung der verschiedenen Dimensionen von Heterogenität und der Analyse von Ursachen von Scheitern [SK22] bis zur Entwicklung von Beratungsangeboten und anderen Maßnahmen [TBH21]. Weitere Untersuchungen legen nahe, dass nicht nur die Studierenden heterogen sind, sondern auch die Erwartungshaltung der Lehrenden an die Vorfähigkeiten der Studierenden substanziell vielfältig ist [Be23].

In Summe machen die Studien deutlich, wie vielschichtig die Frage nach Heterogenität und Diversität der Studierenden ist und wie schwierig es ist, die Auswirkungen einzelner Dimensionen systematisch zu erfassen oder gar voneinander abzugrenzen. Insbesondere mit Blick auf den tatsächlichen Einfluss von Persönlichkeitseigenschaften bzw. von Schlüsselkompetenzen für den Studienerfolg wird die wissenschaftlich fundierte Datenlage sehr schnell sehr dünn – während das professorale Bauchgefühl sich im Lehr-Lernalltag hier durchaus Thesen aufstellt und diese für sich plausibilisiert. Ein Beispiel ist die These, dass Erstsemester-Studierende mit bereits abgeschlossener Berufsausbildung oft reifer, organisierter und zielstrebigere aber gleichzeitig akademisch eher aus der Übung bzw. weniger vorgebildet (insbesondere in Mathe) sind als Studierende, die direkt nach dem Abitur ins Studium einsteigen. Datenbasiert belegen lässt sich das derzeit aber nicht.

Die Beobachtungen aus der eigenen Lehre und dem Diskurs im Kollegium legen nahe, dass ein wirkungsvoller „one size fits all“-Lehransatz schwer zu finden ist. Typische Phänomene in der Erstsemester-Programmierausbildung sind, dass ein (hoffentlich großer) Teil der Kohorte an den professoralen Lippen hängt und inhaltlich anschlussfähig ist, während ein anderer Teil sich gelangweilt fragt, wann denn endlich die spannenden Themen kommen, und wiederum eine andere Studierendengruppe bereits hoffnungslos überfordert ist.

Als Abschlussurkunden-verleihende Institution haben wir die Verpflichtung der Qualitätssicherung – einen Studienabschluss gibt es also ausschließlich bei vorhandenen Kompetenzen, nicht für abgesessene Zeit. Entsprechend haben wir als Lehrpersonen für unser eigenes Wirken den Anspruch, das möglichst viele unserer Studierenden möglichst viele Kompetenzen entwickeln, fachlich wie überfachlich. Insbesondere soll also jede:r etwas für sich mitnehmen können aus unserer Lehre.

Im Folgenden analysieren wir zunächst Beobachtungen und Indikatoren, um die vielschichtigen Dimensionen der Heterogenität besser zu verstehen. Darauf aufbauend stellen wir eine Reihe von Ansätzen und Maßnahmen vor, wie wir in heterogenen Kohorten einen breiten Kompetenzerwerb der Studierenden fördern und begleiten können.

2 Beobachtungen und Indikatoren

Damit wir überhaupt in der Lage sind, bedarfsgerechte Maßnahmen für eine heterogene Kohorte zu entwickeln und aufzusetzen ist es wichtig, zunächst ein Bild zu gewinnen

über die verschiedenen Dimensionen der Unterschiedlichkeiten, die uns in der Kohorte begegnen. Die nachfolgenden Zahlen beziehen sich, sofern nicht explizit anders angegeben, auf die Erstsemester-Kohorte IF1B des Bachelorstudiengangs Informatik im Wintersemester 2023/24 an der Fakultät für Informatik und Mathematik der Hochschule München. Die Vorlesungszeit begann am 01.10.2023 mit den Studieneinführungstagen, der reguläre Vorlesungsbetrieb nach Stundenplan dann am 09.10.2023.

2.1 Fachliche Vorfähigkeiten

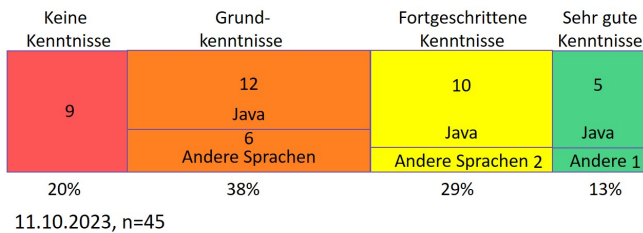


Abb. 1: Studentische Selbsteinschätzung der fachlichen Vorfähigkeiten

Um die fachlichen Vorerfahrungen zu Softwareentwicklung zu erheben haben wir in Präsenz in der ersten regulären Vorlesungseinheit nach den Studieneinführungstagen die studentische Selbsteinschätzung des initialen Kenntnisstandes erhoben (siehe Abbildung 1). Von den anwesenden 52 Studierenden haben 45 ein Votum abgegeben. Es zeigt sich eine heterogene Verteilung: Während ein Fünftel der Studierenden keinerlei Vorkenntnisse im Programmieren mitbringt attestieren sich 42% selbst mindestens fortgeschrittene Vorfähigkeiten, davon 13% sogar auf Profi-Niveau. (Das Ergebnis deckt sich weitestgehend mit dem der Vorgängerkohorte, für die im Wintersemester 2022/23 die gleiche Umfrage durchgeführt wurde.)

Die Umfrage wurde zwei Wochen später in identischer Form wiederholt. Zu diesem Zeitpunkt waren statt der initialen 52 nur noch 36 Studierende anwesend, was einem Schwund von rund 30% entspricht. Von den Anwesenden haben sich 30 an der Umfrage beteiligt.

Eine:r der Teilnehmenden hat dabei angegeben, die Selbsteinschätzung von Profi-Niveau auf Grundkenntnisse angepasst zu haben, während die Selbsteinschätzung der anderen Teilnehmenden im Wesentlichen stabil geblieben ist⁵. Basierend auf diesen Ergebnissen lässt sich grob zurückschließen, in welchen Niveau-Gruppen der Vorfähigkeiten bereits in den ersten Wochen des Studiums hinsichtlich Besuch der Präsenzveranstaltungen ein Schwund auftritt.

⁵ Wir nutzten für beide Umfragen den Dienst polleverywhere.com. Dabei können die Studierenden anonym teilnehmen. Die meisten unserer Studierenden verwenden dabei aber über mehrere Sessions hinweg identische Pseudonyme.

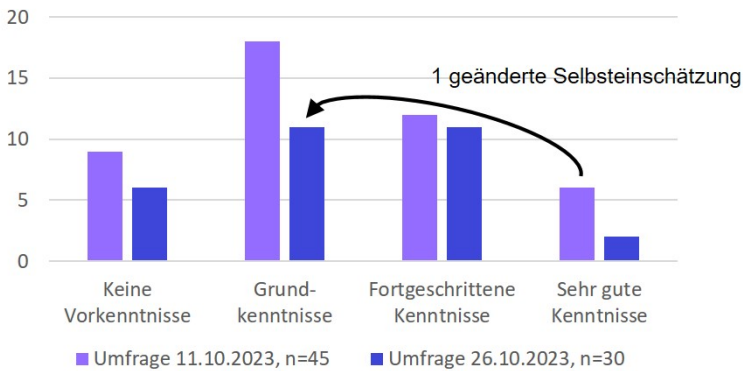


Abb. 2: Schwund differenziert nach fachlichen Vorfähigkeiten

Abbildung 2 visualisiert, dass die Teilnahme bei den Studierenden mit fortgeschrittenen Vorfähigkeiten im Wesentlichen stabil geblieben ist. Von den (wenigen) Teilnehmenden auf Profi-Niveau hat sich noch rund die Hälfte an der zweiten Abstimmung beteiligt. Von den Studierenden, die ohne jegliche Vorkenntnisse oder lediglich mit Grundkenntnissen ins Studium gestartet sind, tritt jeweils rund ein Drittel bei der zweiten Abstimmung nicht mehr in Erscheinung. Das bedeutet, dass insbesondere diejenigen Studierenden mit wenig oder gar keinen fachlichen Vorfähigkeiten frühzeitig aus der Veranstaltung aussteigen.

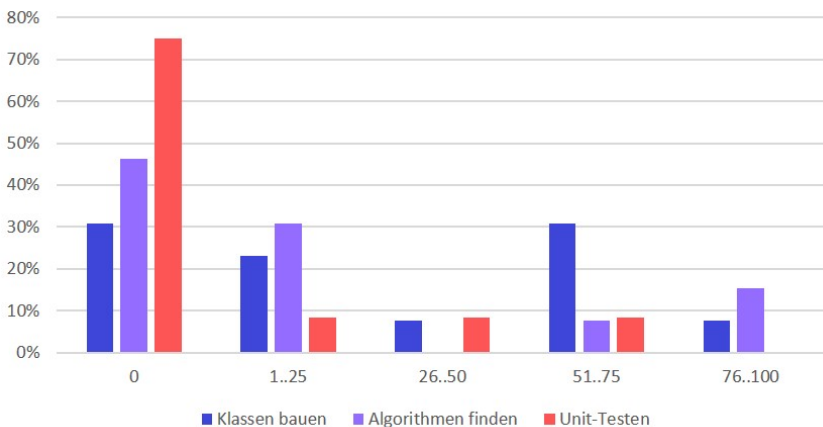


Abb. 3: Fachlichen Vorfähigkeiten differenziert nach Themenbereichen, in % der Lernzielerfüllung

Die Vorfähigkeiten selbst sind dabei auch wieder sehr heterogen, je nach Themenbereich. Abbildung 3 zeigt die Ergebnisse einer Erhebung mit der Vorjahreskohorte, in der die

Studierenden am Ende des Semesters rückblickend ihre Startfähigkeiten relativ zu den für die Veranstaltung definierten Lernzielen [Th15a] einordnen, auf einer Skala von „0 – gar nichts“ bis „100 – alles“. Dabei zeigen sich deutliche Unterschiede zwischen den Themenbereichen. Während über zwei Drittel der Teilnehmenden schon irgendwann einmal mit dem Thema „Klassen bauen“ konfrontiert war und immerhin knapp die Hälfte der Teilnehmenden schon *vor* Beginn der Veranstaltung die Hälfte der Lernziele zu diesem Thema zu erfüllen glaubt, geben dreiviertel der Kohorte an, absolut nichts über das Thema Unit-Testen zu wissen.

Auffällig ist, dass ausgerechnet beim Thema Unit-Testen im Hörsaal-Geschehen die Interaktion der Kohorte mit der Lehrperson aus Wahrnehmung der Lehrenden substanziell eingebrochen ist dergestalt, dass sich de facto niemand gefunden hat, der Fragen bzw. Aufgaben der Lehrperson an die Kohorte freiwillig beantwortet hat. Kombiniert mit der Beobachtung, dass es ansonsten im Wesentlichen immer die gleichen Studierenden sind, die in der Hörsaal-Einheit aktiv mit der Lehrperson interagieren, weckt das die These, dass sich ohnehin lediglich diejenigen Studierenden aktiv einbringen, die die gerade behandelten Inhalte schon vorher beherrscht haben – was für den Lernprozess der anderen natürlich eher kontraproduktiv ist.

2.2 Organisatorischer Hintergrund – dual oder nicht

Im aktuellen Wintersemester studieren rund zwei Fünftel der betrachteten Kohorte dual (konkret 21 der initial eingeschriebenen 50). Diese Studierenden haben also parallel zum Studium einen Arbeitsvertrag mit einer Firma, die das Studium finanziert mit der Intention, die Studierenden nach ihrem erfolgreichen Abschluss als Beschäftigte zu übernehmen.

Die Erfahrungen der letzten Jahre zeigen durchgängig, dass dual Studierende in der Regel ihr Studium erfolgreich zu Ende bringen. Da ziemlich genau die Hälfte der dual Studierenden bereits zu Beginn des Studiums über mindestens fortgeschrittene Programmierkenntnisse verfügt, die andere Hälfte jedoch über wenig bis gar keine Vorkenntnisse, können diese fachlichen Vorfähigkeiten nicht alleine ausschlaggebend sein für den Studienerfolg.

Aus Gesprächen mit den dual Studierenden wird deutlich, dass die begleitende Betreuung durch die Firma für die Studierenden eine wichtige Rolle spielt, und das nicht zwingend nur fachlich, sondern insbesondere auch organisatorisch. Viele der Firmen erlegen ihren dual Studierenden eine Anwesenheitspflicht in den Lehrveranstaltungen auf. Des Weiteren riskieren dual Studierende ihren Arbeitsvertrag, wenn sie Prüfungen nicht wie im Studienplan vorgegeben antreten, sondern nach hinten schieben. Entsprechend ist bei den dual Studierenden oft in erheblichem Maße Zug dahinter.

Gleichzeitig haben die dual Studierenden durch ihren Vertrag mit der Firma ein zwar moderates, aber gesichertes Einkommen über das gesamte Studium hinweg. Sie können sich also in der Vorlesungs- und Prüfungsphase voll auf das Studium konzentrieren.

2.3 Persönlichkeit und Schlüsselkompetenzen

Des Weiteren beobachten wir bei unseren Studierenden im Lernprozess unterschiedliche Herangehensweisen, die wir auf eine Mischung aus Persönlichkeitseigenschaften und Sozialisation zurückführen. Relevant für jeden Lernprozess erscheinen uns die beiden komplementären Eigenschaften „fleißig“ und „verstandesmäßig herangehend“, die Abbildung 4 im Sinne eines Wertequadrates nach Helwig [He48] einander gegenüberstellt. Um Programmieren zu lernen braucht es beides: den Fleiß, um z. B. das Fachvokabular sowie grundlegende Prinzipien zu lernen und den Verstand, weil jede Programmierlösung neu und einzigartig ist, damit nicht rein schematisch erstellt werden kann sondern vielmehr kreatives Problemlösen erfordert. Beide Fähigkeiten müssen also kontextsensitiv passend nach Bedarf abrufbar sein.

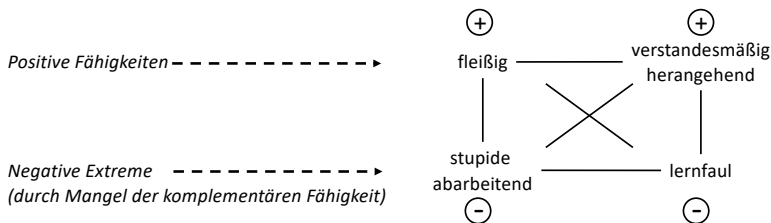


Abb. 4: Wertequadrat zu den Kompetenzen „fleißig“ und „verstandesmäßig herangehend“.

Ist eine der beiden positiven Fähigkeiten in diesem Bipol nicht angemessen ausgeprägt, so kann die andere Kompetenz ohne diese Gegenbalance ausarten zu einem negativen Extrem. Wer fleißig ist, aber den Verstand nicht nutzt, verfällt ins stupide Abarbeiten, während wer ausschließlich verstandesmäßig an Dinge herangeht, aber keinerlei Fleiß zeigt, lernfaul ist auch mit Blick auf grundlegendes Faktenwissen.

Ein hohes Maß an Fleiß bei gleichzeitigem Mangel am Einsatz des eigenen Verstandes manifestiert sich oft in guter bis sehr guter Beherrschung der Lernziel-Kompetenzen auf den Ebenen 1: Wissen und 3: Anwenden, bei gleichzeitigen Schwierigkeiten auf den Ebenen 2: Verstehen, 4: Analysieren und 6: Kreieren gemäß der Lernziel-Taxonomie nach Anderson und Krathwohl [An01]. Umgekehrt sind Indikatoren für viel Verstand in Kombination mit großer Faulheit gute bis sehr gute Kompetenzen auf den Ebenen 2: Verstehen, 4: Analysieren und 6: Kreieren, bei gleichzeitigen Ausfällen auf den Ebenen 1: Wissen und 3: Anwenden. Ebene 5: Evaluieren wiederum braucht sowohl den Verstand insbesondere für die Analyse und das kritische Hinterfragen, dabei aber gleichzeitig solides Wissen über die anzulegenden Qualitätsstandards bzw. Evaluationskriterien – hier tun sich in der Regel beide nicht ausbalancierten Varianten schwer.

In der Lehre beobachten wir immer wieder Studierende, die intellektuelles Potenzial errahnen lassen, für die der Weg ins eigenständige, kritische Denken aber eine echte Herausforderung darstellt. Teilweise führen wir das auf den Sozialisationskontext zurück, der beispielsweise von Aspekten wie Geschlechter-Stereotypen, dem in der Herkunftsfamilie bzw. Gesellschaft

geltenden Grad an Meinungsfreiheit oder dem Verhältnis von Drill zu Selberdenken in der genossenen Schulbildung beeinflusst ist – alles Phänomene, die sowohl national als auch international zu finden sind. Uns als Lehrenden muss in derartigen Konstellationen bewusst sein, wie groß und hürdenreich der Schritt zum „Selber denken macht schlau“ für diese Studierenden sein kann.

2.4 Herausforderungen

Aus den obigen Aspekten der unterschiedlichen fachlichen Vorfähigkeiten, organisatorischen Hintergründe, Persönlichkeitseigenschaften und Schlüsselkompetenzen ergibt sich für jede:n einzelne:n Studierende:n eine individuelle Gemengelage, die wesentlichen Einfluss darauf hat, wie gut jede:r Einzelne mit dem eigenen Lernprozess vorankommt. Auch das haben wir für die Erstsemester-Kohorten der Wintersemester 2022/23 und 2023/24 erhoben, mittels einer Umfrage nach dem Grad der Forderung in der Veranstaltung „Softwareentwicklung 1“ drei bis vier Wochen nach Semesterbeginn. Abbildung 5 stellt die Ergebnisse gegenüber und

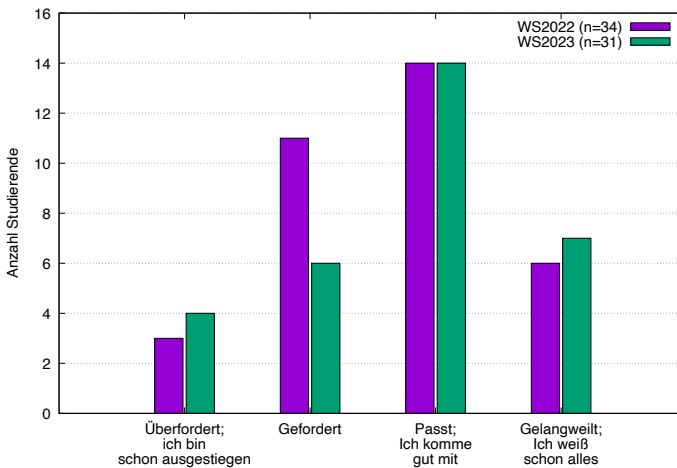


Abb. 5: Grad der gefühlten Herausforderung etwa drei bis vier Wochen nach Semesterbeginn

verdeutlicht deren große Ähnlichkeit. Lediglich knapp die Hälfte der Teilnehmenden sagt aus, dass das Tempo passt und sie gut mitkommen, etwa 20% der Studierenden langweilen sich und der Rest ist stark gefordert oder gar überfordert.

3 Mögliche Maßnahmen

Im Folgenden stellen wir einige Maßnahmen vor, die sich in unserer Lehrpraxis im Umgang mit den heterogenen Studierenden und insbesondere deren heterogenen Vorkenntnissen

bewährt haben. Grundlage unserer Lehre sind dabei explizit definierte, kompetenzorientierte Lernziele [Th15a] sowie das Prinzip des Constructive Alignments [BT11], gemäß dem wir die Prüfung abgestimmt mit den Lernzielen gestalten und Lehr-Lernmethoden konsequent auf das Erreichen dieser Ziele hin ausrichten.

3.1 Differenzierte Übungsblätter

Je nach dem Detaillierungsgrad eines Übungsblatts bzw. einer Aufgabenstellung für eine Praktikumsaufgabe können Studierende überfordert oder gelangweilt werden. Wir experimentieren dabei bereits seit einiger Zeit damit, Übungsblätter in zwei Varianten zu erstellen und diese explizit den Studierenden zur Auswahl anzubieten [ZBT17]. Dabei adressiert eine Variante die Studierenden mit guten Vorkenntnissen in der Softwareentwicklung und die andere Variante die eher herausgeforderten Studierenden.

3.2 Homogenisierung von Gruppen

Um gezielt auf die Bedarfe einer bestimmten Gruppe von Studierenden eingehen zu können kann es hilfreich sein, die Kohorte für ausgewählte Phasen zu homogenisieren.

3.2.1 Erstsemester-Einstiegsprojekt

Beim Erstsemester-Einstiegsprojekt durchlaufen die Studierenden anhand eines Beispiels und mit Hilfe von haptischen und visuellen Materialien die Kernaktivitäten des Software Life Cycles, also Anforderungsanalyse, Entwurf, Umsetzung und Test, in einem iterativ-inkrementellen Prozess [Th15b]. Da die Studierenden der Kohorte sich zu diesem Zeitpunkt gegenseitig so gut wie nicht kennen, entstehen bei freier Gruppeneinteilung nach dem Zufallsprinzip meist heterogene Gruppen, in denen Leute mit Vorkenntnissen auf Leute ohne Vorkenntnisse treffen.

Die Erfahrung zeigt, dass diejenigen Studierenden, die schon fachliche Vorkenntnisse haben (oder zu haben glauben) das Geschehen in diesen Gruppen stark dominieren. In der Folge halten sich Studierende ohne Vorkenntnisse mit ihren Ideen oft eher zurück und beobachten weit gehend passiv, was die (vermeintlich) vorgebildeten als Lösungsweg erarbeiten.

Hier haben wir gute Erfahrungen gemacht mit einer Homogenisierung nach Vorkenntnissen, auf der Grundlage einer studentischen Selbsteinschätzung. Dabei challengen sich die Studierenden in den Gruppen mit Vorkenntnissen untereinander und wirken dabei als gegenseitiges Korrektiv. Die Gruppen ohne Vorkenntnisse orientieren sich gemeinschaftlich, Schritt für Schritt und in untereinander ähnlichem Tempo an der Anleitung und diskutieren auf dem Weg bestehende Unklarheiten.

3.2.2 Zusatztutorial

Ein weiteres Beispiel für ein Lernsetting mit einer homogenisierten Gruppe ist das Zusatztutorial, das wir seit dem Wintersemester 2022/23 anbieten. Hierfür sprechen wir gezielt Studierende an und ermuntern sie zur Teilnahme. Grundlage dafür ist die in Abschnitt 2.4 vorgestellte Umfrage nach dem Grad der Forderung in der Veranstaltung drei bis vier Wochen nach Semesterbeginn (siehe Abbildung 5).

Dieses Zusatztutorial bieten wir gezielt für die Studierenden aus der Gruppe der Geforderten bzw. Überforderten an. Anderen Studierenden verwehren wir explizit den Zugang, um die Gruppe von der fachlichen Leistungsfähigkeit her möglichst homogen zu halten. Gleichzeitig schaffen wir so einen geschützten Raum, in dem sich alle trauen, ihre Fragen offen zu stellen, weil allen klar ist, dass die anderen auch nicht mehr wissen als man selbst.

Durch diese Vorgehensweise vermeiden wir außerdem den üblicher Weise zu beobachtenden Effekt, dass «exakt die „falschen“ Student*innen zu Tutorien und zu Vorkursen kommen»⁶ – also die, die es eigentlich gar nicht nötig haben.

Von den Studierenden, die wir über die Umfrage zur Klassifizierung des eigenen Gefordert-Seins gezielt angesprochen haben, hat im Wintersemester 2022/23 rund die Hälfte, im Wintersemester 2023/24 rund zwei Drittel der von Forderung oder Überforderung betroffenen Studierenden von unserem Angebot Gebrauch gemacht.

Das Tutorium wird von zwei Dozierenden im Modus des Pair-Teaching durchgeführt [ZBB18], um unterschiedliche Zugänge zu den Themen zu ermöglichen.

3.3 Heterogenisierte Gruppen

Heterogenisierte Gruppen sind insbesondere bei Projektarbeiten nützlich, wenn die einzelnen Beteiligten sich mit ihren individuellen Skill-Sets gegenseitig gut ergänzen. Hilfreich ist dabei allerdings ein gemeinsam zugrunde liegendes Wertesystem, um nicht permanent Grundsatzdiskussionen führen zu müssen, sowie eine offene Wertschätzung für eben genau die Diversität in der Gruppe.

Da in der Veranstaltung Softwareentwicklung 1 alle Erstsemester-Studierenden das Programmieren selbst erlernen sollen, fordern wir von den Studierenden individuelle Abgaben. Um den gegenseitigen Austausch auf dem Weg zur Lösung zu fördern, fordern wir von den Studierenden immer wieder gegenseitige Peer Reviews ihrer Lösungen ein, inklusive einer anschließenden kritischen Auseinandersetzung mit dem Feedback, das die Studierenden selbst zu ihrer eigenen Lösung erhalten haben.

⁶ Siehe dazu beispielsweise diesen Blog-Eintrag von Jörn Loviscach: https://j317h.de/blog/2023-05-17_09_44_VomVerschwindenderTutorien

Als besonders Erkenntnis stiftend herauskristallisiert hat sich dabei das gezielte Bilden von Review-Tandems, die in ihren fachlichen Fähigkeiten möglichst heterogen sind. Die Schwächeren erhalten von den leistungsstärkeren Studierenden Hinweise zu möglichem Verbesserungspotenzial ihrer Lösung und sehen gleichzeitig aufgeräumte Lösungen mit einer gewissen Quelltext-Qualität. Umgekehrt erhalten die leistungsstärkeren Studierenden Hinweise zu Lücken in der Dokumentation bzw. zu Quelltext-Stellen, die schwer lesbar sind, und lernen gleichzeitig, sich in andere Lösungen und die dahinter liegenden Gedankengänge einzuarbeiten.

Dieser Ansatz lässt sich ausbauen bis hin zum kollegialen studentischen Coaching über einen längeren Zeitraum, beispielsweise die gesamte Studieneingangsphase.

3.4 Videos aufzeichnen

Im letzten der Corona-Semester, also im Sommer 2022, konnten wir Erfahrungen mit hybrider Lehre bei gleichzeitiger Aufzeichnung der Lehrveranstaltung machen. Dabei hat überrascht, dass die Aufzeichnungen deutlich stärker genutzt wurden als erwartet [Bö23]. Es zeigt sich, dass insbesondere das Aufzeichnen von Aktivitäten in der Entwicklungsumgebung für die Studierenden sehr hilfreich ist, weil die einzelnen Bedienungsschritte oft so schnell ablaufen, dass die Studierenden diese nicht mental aufnehmen und verarbeiten können. Über die Aufzeichnung bekommen die Studierenden die Möglichkeit, die Bedienungsschritte in ihrem eigenen Tempo noch einmal nachzuvollziehen und ggf. am eigenen System mit durchzuführen.

4 Wirkung

Der Nachweis der Wirksamkeit der Maßnahmen ist im Bildungsbereich meist schwer zu führen. So lässt sich nicht nachweisen, ob Studierende die Prüfung ohne eine Maßnahme wie einem zusätzlichen Tutorium nicht bestanden hätten.

4.1 Zusatztutorium

Im Wintersemester 2022/23 haben von sieben regelmäßig Teilnehmenden des Zusatztutoriums vier die Modulabschlussprüfung im ersten und zwei im zweiten Versuch (Sommer 2023) bestanden. Zur Evaluation des Zusatztutoriums wurden von einer studentischen Hilfskraft aus dem Studiengang „Soziale Arbeit“ Interviews durchgeführt. Insbesondere wurden die Teilnehmenden gefragt, ob das Tutorium in dieser Form geholfen hat und weiter angeboten und entwickelt werden soll.

Folgende Rückmeldungen sind repräsentativ:

- «Raum für Fragen»
- «Es werden alle Fragen beantwortet»
- «Das Zusatztutorial, was der Herr Böttcher anbietet. Weil wir sind in einer recht kleinen Gruppe, ich glaube dieses Semester sind wir fünf Leute, und dann traut man sich eher Fragen zu stellen. Und man kann auch dumme Fragen stellen, weil man weiß halt, der andere würde auch vielleicht seine Fragen und – so alle auf demselben Level.»

4.2 Allgemeiner Lernfortschritt

Im Wintersemester 2022/23 haben wir die Studierenden am Ende des Semesters rückblickend ihre Startfähigkeiten relativ zu den für die Veranstaltung definierten Lernzielen einordnen lassen (siehe Abbildung 3), sowie zusätzlich ihren individuellen Lernfortschritt in den verschiedenen Themenbereichen. Abbildung 6 zeigt beispielhaft für die Themenbereich „Klassen bauen“, „Algorithmus finden“ und „Unit-Testen“, dass zu jedem Thema *alle* Studierenden aus der Veranstaltung etwas für sich mitgenommen haben, unabhängig von ihren Vorfähigkeiten.

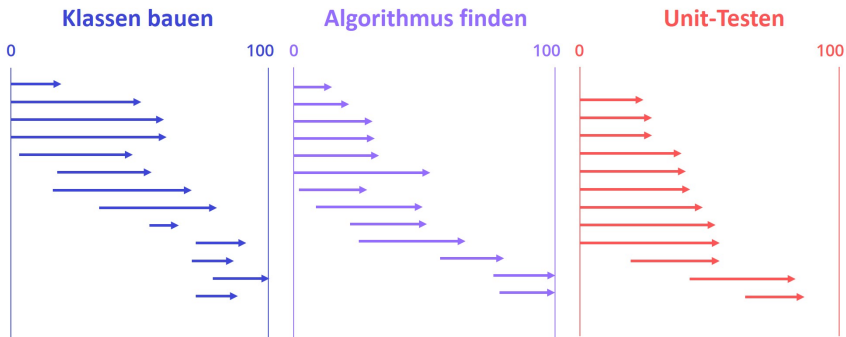


Abb. 6: Fachlicher Lernfortschritt differenziert nach Themenbereichen, in % der Lernzielerfüllung

Welche der durchgeführten Maßnahmen dabei wie stark zum Lernerfolg beigetragen haben lässt sich jedoch nicht identifizieren.

5 Fazit

In dieser Arbeit haben wir eine Reihe von Ansätzen und Maßnahmen vorgestellt, die sich in der Grundlagenausbildung zu Softwareentwicklung für das Lehren und Lernen von und mit heterogenen Kohorten in der informatischen Studieneingangsphase bewährt haben. Viele dieser Ansätze sind von ihrer jeweiligen Grundidee her auch in andere Lehr-Lernkontexte übertragbar.

Literatur

- [An01] Anderson, L. W.; Krathwohl, D. R.; Airasian, P. W.; Cruikshank, K. A.; Mayer, R. E.; Pintrich, P. R.; Rath, J.; Wittrock, M. C.: A Taxonomy for Learning, Teaching, and Assessing. A Revision of Bloom's Taxonomy of Educational Objectives. Longman, New York, 2001.
- [Be23] Bender, E.; Barbas, H.; Hamann, F.; Soll, M.; Sitzmann, D.: Fähigkeiten und Kenntnisse bei Studienanfänger*innen in der Informatik: Was erwarten die Dozent*innen? Hochschuldidaktik Informatik HDI 2021 (Commentarii informaticae didacticae)/13, S. 279–299, 2023.
- [Bö23] Böttcher, A.: Experiences with Low-to-Medium-Effort Hybrid Teaching. In: IEEE Global Engineering Education Conference (EDUCON). S. 1–7, 2023.
- [BT11] Biggs, J.; Tang, C.: Teaching For Quality Learning At University. McGraw-Hill Education, 2011, ISBN: 9780335242757.
- [De23] Desel, J.; Opel, S.; Siegeris, J. (Hrsg.): Hochschuldidaktik Informatik HDI 2021. In: Commentarii informaticae didacticae. 9. Fachtagung des GI-Fachbereichs Informatik und Ausbildung/Didaktik der Informatik 15.-16. September 2021 in Dortmund 13, 2023.
- [He48] Helwig, P.: Das Wertequadrat. *Psyche* 2/1, S. 121–127, 1948.
- [SK22] Soll, M.; Kobras, L.: What Were We Expecting? Analysing Expectations of German University Teachers of Study Beginners in Computer Science as Experienced by Students. In: 2022 IEEE German Education Conference (GeCon). S. 1–6, 2022.
- [TBH21] Thurner, V.; Böttcher, A.; Häfner, T.: A Detailed Analysis of Gender Differences in the Course of CS-Studies. In: 2021 IEEE Global Engineering Education Conference (EDUCON). S. 482–491, 2021.
- [Th15a] Thurner, V.; Böttcher, A.; Schlierkamp, K.; Zehetmeier, D.: Lernziele für die Kompetenzentwicklung auf höheren Taxonomiestufen. In: Tagungsband des 14. Workshops Software Engineering im Unterricht der Hochschulen 2015, Dresden, Deutschland, 26. - 27. Februar 2015. S. 9–20, 2015.
- [Th15b] Thurner, V.; Schlierkamp, K.; Zehetmeier, D.; Böttcher, A.: Software Engineering Project Simulation in Student Entry Phase of Computer Scientists-to-be. In: Proceedings of the 2015 IEEE Global Engineering Education Conference (EDUCON), Tallinn. S. 486–493, März 2015.
- [ZBB18] Zehetmeier, D.; Böttcher, A.; Brüggemann-Klein, A.: Designing Lectures as a Team and Teaching in Pairs. In: Proc. 4th International Conference on Higher Education Advances (HEAd'18), Valencia. S. 873–880, 2018.
- [ZBT17] Zehetmeier, D.; Böttcher, A.; Thurner, V.: Differenzierte Übungsblätter – Für Experten und solche die es werden wollen. In: Tagungsband zum 3. Symposium zur Hochschullehre in den MINT-Fächern. S. 94–98, Sep. 2017.