

Entwicklung und Erprobung eines Rahmenkonzepts zum Programmierenlernen an Hochschulen

Das Programmierenlernen ist bereits seit den 1980er Jahren Gegenstand der allgemeinen Hochschullehre. Dennoch besitzt es auch nach über vierzig Jahren eine hohe Aktualität für die Forschung im Bereich des Lehrens und Lernens. Robins (2019) gibt einen umfassenden Überblick zum aktuellen Stand der Forschung auf diesem Gebiet. Darin wird deutlich, dass es unter den Lehrenden für Grundlagenkurse zum Programmierenlernen kaum Übereinstimmung bezüglich der Methodik, zu allgemein akzeptierten Lehrinhalten und zur Diagnostik der Erreichung von Lernzielen gibt. Konsenz besteht darin, dass für das Programmierenlernen komplexe kognitive Anforderungen in den Teilbereichen des Programmverstehens und des Programmschreibens bestehen, da beim Programmieren dynamische Abläufe generischer Problemlöseprozesse mit den Mitteln formaler Programmiersprachen zu beschreiben sind.

Das Ziel dieser Arbeit besteht darin, ein lernpsychologisch fundiertes Rahmenmodell zum Programmierenlernen zu entwerfen und dieses im realen Lehrbetrieb an einer Hochschule zu erproben.

Für das Programmierenlernen sind eine Vielzahl unterschiedlicher Begriffe, Strategien und Fertigkeiten in wechselseitigen Abhängigkeiten aufzubauen. Diese komplexen Lernprozesse bedürfen eines hohen Anteils praktischer Programmierfähigkeit. Damit erscheinen Methodiken des aufgabenbasierten Lernens als Grundlage des Rahmenmodells geeignet. Four-Components Instructional Design (4C/ID) ist eine Methodik des aufgabenbasierten Lernens zur Entwicklung komplexer kognitiver Fertigkeiten, die seit mehr als 30 Jahren vor allem in der berufspraktischen Ausbildung, aber auch in Teilbereichen der Informatik, zum Einsatz kommt (van Merriënboer und Kirschner, 2018). Es handelt sich um eine Methodik des induktiven Lernens, deren zentrales Element Folgen von Lernaufgaben darstellen. Diese Aufgabenfolgen sind eingebettet in unterstützende Information und Anleitungen zur Handlungsausführung. Task-Based Learning (TBL) nach Willis (2004) ist ein Rahmenmodell des aufgabenbasierten Lernens aus dem Bereich des Fremdsprachunterrichtes. Es verfügt u.a. über methodische Elemente zur Erkenntnissicherung und -konsolidierung sowie zur praktischen Anwendung. Damit kann TBL das Defizit des Fehlens genau dieser methodischen Elemente in 4C/ID kompensieren. Das kombinierte Rahmenmodell zum Programmierenlernen verbindet die Methodiken von 4C/ID und TBL derart, dass der methodische Rahmen von TBL die Aufgabenfolgen von 4C/ID einbettet. Damit wird erreicht, dass innerhalb des kombinierten Rahmenmodells Elemente der Einführung und Motivation für die Lernaufgaben bestehen, Erkenntnisse der Aufgabenbearbeitung gesichert werden und dass eine abschließende komplexe Anwendung der Lernergebnisse erfolgt. Die Ausgestaltung der Lernaufgaben mit Inhalten der Programmierfähigkeit führt zu einem kombinierten Rahmenmodell für das Programmierenlernen.

Folgen von Lernaufgaben sind das methodische Instrument zum Aufbau von Begriffen und zur Entwicklung praktischer Fertigkeiten entsprechend der jeweiligen Lernziele. Dabei sind es die spezifischen Lernhandlungen, die bei der Aufgabenbearbeitung vollzogen werden, die den Kompetenzaufbau ermöglichen sollen. Folglich gilt es die charakteristischen Handlungen des Programmierens zu identifizieren. Für die Prozesse des Programmverstehens gibt Schulte (2008) ein Blockmodell an, aus dem sich Handlungen des Programmverstehens in vier Ebenen und den Bereichen der Codestruktur, der Codeausführung und der Codefunktion ableiten lassen. Ein gleichartiges Modell des Programmschreibens ist aus der Literatur nicht bekannt. Daher wurde im Rahmen dieser Arbeit ein Blockmodell zum Programmschreiben auf Basis der von Schulte definierten Grundstruktur entworfen. Mit diesem Modell können die Handlungen des Programmschreibens in vier Abstraktionsebenen und drei Funktionsbereichen identifiziert werden. Die typischen Prozesse des Programmierens zeigen sich als Handlungen in den einzelnen Zellen beider Blockmodelle. Die klassischen Top-Down- oder Bottom-Up-Prozesse beim Programmieren stellen navigierende Handlungsfolgen über den Zellen der Blockmodelle dar. Ein Katalog von Beispielaufgaben zu allen Zellen der Blockmo-

delle illustriert die Möglichkeiten zur Gestaltung entsprechender Lernaufgaben. Dabei wird der jeweilige Prozess der Aufgabenbearbeitung unter Nutzung des Toulmin-Schemas (Toulmin und Berk, 1996) dokumentiert, wodurch die einzelnen Lernhandlungen explizit sichtbar gemacht werden können.

Um die Programmierkompetenz diagnostizieren zu können bedarf es eines geeigneten Kompetenzmodells. Leighton und Gierl (2011) stellen unterschiedliche kognitive Modelle zur pädagogischen Diagnostik vor. Dabei erscheint das 5-Strands-Model (Kilpatrick, Swafford und Findell, 2001) aus dem Bereich der US-amerikanischen Mathematik-Didaktik geeignet, auch im Bereich des Programmierenlernens nutzbar zu sein. Dieses Modell basiert auf den Elementen Konzeptverständnis, Handlungsfähigkeit, Problemlösen, Transfer und produktive Grundeinstellung, wodurch es in inhaltlicher Nähe zur allgemeinen Kompetenzdefinition von Weinert (2002) steht. Die Anwendung des 5-Strands-Modells in der Domäne des Programmierenlernens erfordert geeignete Testaufgaben zur Diagnose der einzelnen Modellkomponenten. Diese Testaufgaben sind gemäß den Lernzielen für die fachlichen Themenbereiche des Programmierenlernens zu gestalten. Das Lösen der Testaufgaben kann im Rahmen von Lernstandserhebungen erfolgen, die als formatives Diagnoseinstrument zur Erreichung der Lernziele in das kombinierte Rahmenmodell zum Programmierenlernen integriert werden.

Die Methodik des kombinierten Rahmenmodells zum Programmierenlernen wurde in den Grundlagenkurs zur Python-Programmierung implementiert. Dafür wurden in realen Lehrveranstaltungen mit der Dauer von jeweils einem Semester drei Machbarkeitsstudien durchgeführt. Die erste Machbarkeitsstudie fand in einer Wahlveranstaltung im Wintersemester 2020/21 unter den Bedingungen der Online-Lehre statt. Das Ergebnis hat gezeigt, dass die Methodik des kombinierten Rahmenmodells grundsätzlich funktioniert und dass die Lernstandserhebungen mit Testaufgaben basierend auf dem 5-Strands-Model die Programmierkompetenzen differenziert erfassen können. Die folgenden Machbarkeitsstudien wurden im 1. Semester des Bachelorstudiengangs "Wirtschaftsingenieurwesen" als Pflichtlehrveranstaltung realisiert. Dabei haben im Wintersemester 2021/22 und 2022/23 jeweils 70-80 Studierende unter Nutzung von Jupyter Notebook als digitale Lernumgebung mit dem eigenen Notebook oder Tablet teilgenommen. Die klassische Vorlesung wurde als Lehrvideo im Umfang von 20-30 Minuten aufgezeichnet. Sie war jeweils vor den einzelnen Lehrveranstaltungen anzuschauen, so dass die Studierenden zwei Mal pro Woche jeweils eine Doppelstunde im Hörsaal und im Seminarraum aktiv an den Programmieraufgaben arbeiten konnten. Im Verlauf des Semesters wurden drei Lernstandserhebungen durchgeführt. Diese wurden, angeleitet durch die Lehrkraft, von den Studierenden selbst bewertet. Die Ergebnisse wurden anonymisiert digital erfasst und als Stichprobe von der Lehrkraft zweitbewertet. Dabei hat sich gezeigt, dass die Methode der angeleiteten Selbstbewertung unter den Rahmenbedingungen dieser Lehrveranstaltung objektive Ergebnisse liefert. Zugleich erhielten die Studierenden und die Lehrkraft im Verlauf der Lehrveranstaltung Feedback zur Erreichung der Lernziele. Zudem war es möglich, die Ergebnisse der Lernstandserhebungen im Kontext der konkreten Lehr-Lernsituation zu erklären. Damit konnte die pädagogische Bedeutsamkeit der Lernstandserhebungen für den Kursverlauf demonstriert werden. Die Entwicklung und Erprobung der Lernstandserhebungen als formatives Diagnose- und Feedbackinstrument ist als wichtigstes Ergebnis der Programmierkurse in den Wintersemestern 2021/22 und 2022/23 anzusehen.

Nach Abschluss jedes Programmierkurses wurden zehn Studierende zufällig für ein Interview zur Retrospektive auf den Kurs ausgewählt. Davon haben ein Mal acht und ein Mal neun Personen am Interview teilgenommen. Die Gespräche fanden gemäß der Methode der Leitfadengespräche statt. Die inhaltliche Auswertung der Interviews erfolgte nach einem Mixed-Methods-Design, um die Vorteile qualitativer und quantitativer Analysemethoden zu kombinieren. In den Ergebnissen zeigte sich, dass die Studierenden die methodische Gestaltung des Programmierkurses und dessen Fokussierung auf die eigene praktische Programmierarbeit positiv einschätzten. Ebenso wurden Jupyter-Notebook als digitale Lernumgebung, die qualitative Gestaltung der Aufgabenfolgen und der fachliche Inhalt der Arbeitsmaterialien positiv bewertet. Die Lernstandserhebungen und dabei insbeson-

dere das elaborierte Feedback der Lehrkraft während der Phase der angeleiteten Selbstbewertung der eigenen Aufgabenlösungen wurden von den Studierenden als hilfreich für das eigene Lernen wahrgenommen. Kritische Aussagen bezogen sich auf das teilweise als zu hoch eingeschätzte Anforderungsniveau der Lernaufgaben sowie auf die als mehrfach sehr knapp wahrgenommene Zeit zur Aufgabebearbeitung.

Mit dieser Arbeit ist es gelungen, ein Rahmenmodell zum Programmierenlernen auf der Grundlage lernpsychologisch fundierter Methodiken zu gestalten. Dieses Rahmenmodell konnte als Grundlagenkurs zum Erlernen der Python-Programmierung im realen Lehrbetrieb der HTW Dresden in mehreren Durchgängen unter unterschiedlichen Rahmenbedingungen erprobt werden. Dabei sind die Lernstandserhebungen als lernförderliches formatives Diagnose- und Feedbackinstrument in das Rahmenmodell integriert worden. Studentische Aussagen in retrospektiven Interviews bestätigen den Erfolg der Lehrveranstaltung auf Grundlage des methodischen Rahmenmodells und der zugehörigen Gestaltung der Lernaufgaben.

In der weiterführenden Forschung erscheint es interessant, die Perspektive von Lehrpersonen bei der Nutzung dieses Rahmenmodells zu dokumentieren, um daraus hochschuldidaktische Erkenntnisse abzuleiten.

Quellenangaben:

- Kilpatrick, J., Swafford, J. & Findell, B. (2001). Adding It Up: Helping Children Learn Mathematics. Doi:10.17226/9822
- Leighton, J. P. & Gierl, M. J. (2011). The learning sciences in educational assessment: The role of cognitive models. doi:10.1017/CBO9780511996276
- Robins, A. V. (2019). Novice Programmers and Introductory Programming. In S. A. Fincher & A. V. Robins (Hrsg.), *The Cambridge Handbook of Computing Education Research* (S. 327–376). Cambridge University Press.
- Schulte, C. (2008). Block Model: An Educational Model of Program Comprehension as a Tool for a Scholarly Approach to Teaching. In *Proceedings of the Fourth International Workshop on Computing Education Research* (S. 149–160). doi:10.1145/1404520.1404535
- Toulmin, S. E. & Berk, U. (1996). *Der Gebrauch von Argumenten* (2. Auflage). Weinheim: Beltz Athenäum.
- van Merriënboer, J. & Kirschner, P. A. (2018). *Ten steps to complex learning a systematic approach to four-component instructional design* (3. Auflage). New York, Routledge, Taylor & Francis Group.
- Weinert, F. E. (2002). *Leistungsmessungen in Schulen* (2. unveränd. Auflage). Weinheim: Beltz.
- Willis, J. (2004). *A framework for task-based learning* (1. publ., 8. impr.). Publication Title: Longman handbooks for language teachers. Harlow: Longman.