



Unterrichtsmethoden

SKRIPT – WWU MÜNSTER

D. MEYER

Unterrichtsmethoden

Inhaltsverzeichnis

Einleitung.....	2
Unterrichtsmethoden.....	3
Der Begriff der Unterrichtsmethode	3
Methodentaxonomie anhand von Lernphasen und Wissensprozessen	3
Auswahl an geeigneten Unterrichtsmethoden für den Informatikunterricht	4
Praktische Beispiele.....	11
Literatur.....	13
Übungen	14

Einleitung

Methodik und Didaktik gehen Hand in Hand, daher wird der Begriff der 'Didaktik' auch als Überbegriff für beide Aspekte von Unterricht verwendet, also in der Didaktik die Methodik gleich mit betrachtet. Dies vor allem vor dem Hintergrund, dass das 'Was' stets mit dem 'Wie' verbunden ist und von diesem abhängt. Welche Inhalte (was) sollen vermittelt werden und mit welchen Methoden (wie) kann dies geschehen? Es kann jedoch auch hilfreich sein, methodische Fragen aus dem Gesamtkontext teilweise herauszulösen und sie separat zu betrachten. In diesem Kapitel soll genau das geschehen. Es wird ein genauerer Blick auf die Fragen des 'Wie' geworfen. Aus der Fülle möglicher Methoden für Unterricht generell sollen in der Praxis bewährte, für den Informatikunterricht besonders geeignete Methoden vorgestellt und mit Beispielen und Anregungen versehen werden. Grundlagenarbeit auf diesem Gebiet leistete Andreas Zandler, auf dessen Arbeiten die Darstellungen im Wesentlichen basieren. Das ausführliche Buch [Ze18] kann dabei zur Vertiefung empfohlen werden.

Unterrichtsmethoden

Unterrichtsmethoden arrangieren die Lerngruppe, sie bilden die Art der Vermittlung der Inhalte, antworten auf das 'Wie' von Unterricht. Zwar gibt es einen ganzen Kanon an verschiedenen Methoden über alle Fächer, werden jedoch zentralen Aspekte dieser extrahiert, dann bleiben nicht viele verschiedene Arten der Vermittlung übrig. Wichtig dabei ist, ein gewisses gemeinsames Begriffsverständnis festzulegen, daher werden die Methoden an der jeweiligen Stelle ausreichend beschrieben.

Die Leitaspekte für Unterrichtsmethoden sind Effektivität und Effizienz. Theoretisch könnten die Lerninhalte in der Informatik auch „einfach frontal“, also in Form einer Vorlesung, vermittelt werden. Nur sollen Lernprozesse zum einen durch die geeignete Wahl der Methoden unterstützt werden – also effektiv und effizient gestaltet werden – und zum anderen gehört zum Lernen – insbesondere an Schulen, also bei Heranwachsenden Lernenden – noch weiteres mit zu diesem Begriff, nicht nur das Fachliche. Methodenmonotonie behindert Lernen durch verschiedene Faktoren, sinnlose Methodenvielfalt oder die Wahl ungeeigneter Methoden aber ebenso. Auch ist aus der eigenen Erfahrung heraus bekannt, dass nicht jede Methode für alle Inhalte und Lernendenkonstellationen gleich geeignet ist. Aus diesem Problemkreis heraus stellt sich überhaupt erst die Frage nach passenden Unterrichtsmethoden. Auch war das der Ausgangspunkt für einen empirischen Zugang zur Methodenfrage, der insbesondere durch die vielfach referenzierte, sehr umfangreiche (Meta-)Studie durch John Hattie – die Hattie-Studie[Ha09] – von 2009 auch (wieder) verstärkt in den Blick einer breiteren didaktischen Diskussion rückte. Allerdings arbeitet die Hattie-Studie nicht Informatikspezifisch, kann jedoch zum Grundlagenstudium sehr angeraten werden. Wichtig in diesem Kontext ist insbesondere der bereits erwähnte Aspekt, dass zum einen nicht jede Methode in jedem Fach gleich gut funktioniert und zum anderen nicht für jede Lehrkraft geeignet ist. Dies wurde unter anderem von Hattie festgestellt und durch Zendler in [Ze18] und [Ze15] für das Fach Informatik bestätigt.

Der Begriff der Unterrichtsmethode

Begriffe nichtaxiomatischer Wissenschaften einer strengen Definition zu unterziehen missglückt für gewöhnlich und artet in kontraproduktive Wortgebilde aus. Das eingangs eingeführte „Unterrichtsmethoden arrangieren die Lerngruppe, sie bilden die Art der Vermittlung der Inhalte und antworten auf das 'Wie' von Unterricht.“ trägt weit genug für fast alle Zwecke. Eine Methode legt insbesondere die Form der Arbeit durch die Gruppenkonstellation fest. Wird alleine, in Partnerarbeit, still oder im Gespräch gearbeitet? Welche Hilfsmittel werden verwendet, wie sieht der Zeitrahmen aus und welche Ergebnissicherung erfolgt? Das sind wesentliche Fragen, auf die eine Methode in spezifischer Weise antwortet.

Methodentaxonomie anhand von Lernphasen und Wissensprozessen

Wird der Frage nachgegangen, welche Methoden denn für den Informatikunterricht geeignet erscheinen, so ist für die Überlegungen ein Bewertungsschema einzurichten, insbesondere wenn es um empirische Arbeiten geht. Anhand welcher Kriterien soll eine Methode in welcher Art und Weise getestet und bewertet werden?

Dabei ist es wichtig sich daran zu erinnern, dass es verschiedene Phasen des Lernens gibt. Es gibt einen Ausgangspunkt, ein Ziel und einen Weg vom Ausgangspunkt zum Ziel (Verlauf), der 'klassische Dreischritt'. Jede Phase hat eine andere didaktische Funktion. Zendler nennt in [Ze18], S. 4,

- für den Einstieg - Sieben didaktische Funktionen:

Ermitteln der Lernvoraussetzungen, Aktivieren von Vorwissen, Informieren über Lehrziele und Motivieren, Begründen, Legitimieren, Anbahnen

- für den Abschluss - Sechs didaktische Funktionen:

Abschließen, Sichern, Überprüfen, Rückmelden, Bewerten und Öffnen.

- für die Arbeitsphase (Verlauf/Prozess vom Einstieg zum Ziel) - Drei didaktische Funktionen: Lernhandeln anregen, Lernhandeln unterstützen, Lernhandeln lenken. Insbesondere ist diese Phase für drei Prozesse beim Lernvorgang von zentralem Interesse. Gemeint sind die Prozesse bei Wissenserwerb (Aufbauen, Durcharbeiten), bei der Wissenstransformation (Anwenden, Übertragen) und bei der Wissensbewertung (Bewerten, Integrieren).

Neues Wissen muss zunächst aufgenommen, dann – etwa in Form von Übungen – integriert/“scharf geschaltet“ und schließlich langfristig abgespeichert und auf neue Sachverhalte übertragen werden.

Diese drei wesentlichen Etappen werden von verschiedenen lehr-/lerntheoretischen Ansätzen unterschiedlich ausgearbeitet. So spricht etwa Bruner im Klassiker >The process of education< ([Br77], S. 48) von den drei Lernprozessen “acquisition of new information, transformation, evaluation“¹, während der aus dem Lernen im Handwerk entlehnte, im situierten Lernen verortete Ansatz der kognitiven Anlehre (cognitive apprenticeship) von Collins et al. ([CBH91], [CBN87]) den Lernprozess aus der Sicht der Lehrenden betrachtet und in sechs Phasen unterteilt: modelling, coaching, scaffolding and fading, articulation, reflection und exploration.²

Wie auch immer begründet diese Detailaufteilung erfolgt, es muss sich entschieden werden. Das hat auch Zendler getan, Wissensprozesse beim Lernvorgang in Erwerb, Transformation und Bewertung gegliedert und in sechs Punkten erfasst:

Wissenserwerb:

- Aufbauen (build)
- Durcharbeiten (process)

Wissenstransformation:

- Anwenden (apply)
- Übertragen (transfer)

Wissensbewertung:

- Bewerten (assess)
- Integrieren (integrate)

Jede dieser Phasen wird von unterschiedlichen Methoden bedient, insbesondere was ihren effektiven und effizienten Einsatz angeht.

Auswahl an geeigneten Unterrichtsmethoden für den Informatikunterricht

Besagte Vorüberlegungen legte Zendler seiner Studie [Ze15] zu Grunde, die als eine der wenigen ausführlichen und informatikspezifischen Methodenstudien als Grundlage für dieses Kapitel verwendet wird, ergänzt um ausgedehnte praktische Erfahrungen in der Informatiklehre an Hochschulen und Schulen. Das folgende Sechs-Kriterien-Raster wendete Zendler auf eine Vorauswahl von 20 als prinzipiell geeignet eingeschätzte Methoden an, auf denen dann die empirische Arbeit stattfand. Die Methoden werden in [Ze15], S. 925, explizit kurz beschrieben, damit ein einheitliches Verständnis der Befragten bzgl. der genannten Methode bestand.

¹ Acquisition/transformation/evaluation – deutsch: Wissenserwerb/Wissenstransformation/Wissensbewertung

² Modellierung – erste modellhafte Vorstellung des zu lernenden, coaching – Nachahmen unter Anleitung, scaffolding – Unterstützung, fading – immer geringer werdende Unterstützung, articulation – Artikulation der Problemlösungsprozesse durch den Lernenden selbst, reflection – Reflexion des neu Gelernten, seine Grenzen und Vor-/Nachteile, exploration – flexible, vielseitige Anwendung und Übertragung des Gelernten

not significant 0 1 2 3 4 5 very significant

	Knowledge processes (Explanations, see Booklet)						
	Instructional methods (Explanations, see Booklet)	1. build	2. process	3. apply	4. transfer	5. assess	6. Integrate
1	Case study						
2	Computer simulation						
3	Concept mapping						
4	Direct Instruction						
5	Discovery learning						
6	Experiment						
7	Guidelines text method						
8	Jigsaw method						
9	Learning at stations						
10	Learning by teaching						
11	Learning tasks						
12	Models method						
13	Portfolio method						
14	Presentation						
15	Problem-based learning						
16	Programmed instruction						
17	Project work						
18	Reciprocal teaching						
19	Role-play						
20	Web quest						

Fig. A 1. Instructional methods and knowledge processes in the act of learning

3

Es wurden 120 erfahrene Oberstufenlehrkräfte befragt, die ihre *Einschätzungen* in Form obiger Tabelle abgeben sollten. 24 Rückmeldungen wurden ausgewertet, auf verschiedene Weisen visualisiert und lieferten interessante Ergebnisse.

Die Mittelwerte (je größer die Zahl, desto besser) stellen sich wie folgt dar:

³ [Ze15], S. 926

Knowledge processes

Instructional methods

- a_1 = Case study
- a_2 = Computer simulation
- a_3 = Concept mapping
- a_4 = Direct instruction
- a_5 = Discovery learning
- a_6 = Experiment
- a_7 = Guidelines text method
- a_8 = Jigsaw method
- a_9 = Learning at stations
- a_{10} = Learning by teaching
- a_{11} = Learning tasks
- a_{12} = Models method
- a_{13} = Portfolio method
- a_{14} = Presentation
- a_{15} = Problem-based learning
- a_{16} = Programmed instruction
- a_{17} = Project work
- a_{18} = Reciprocal teaching
- a_{19} = Role-play
- a_{20} = Web quest

	build	process	apply	transfer	assess	integrate
	2.29	2.24	2.82	2.81	2.86	2.43
	3.23	3.00	3.14	3.18	2.73	2.21
	1.90	2.38	1.64	2.08	2.33	2.23
	4.32	3.55	3.09	2.55	2.00	2.09
	3.66	2.62	2.82	2.90	2.90	2.90
	3.35	2.15	2.42	2.05	2.10	1.90
	2.81	2.57	1.90	1.91	1.29	1.29
	2.96	2.91	2.26	1.70	1.78	1.57
	2.95	3.18	2.95	2.45	1.82	1.82
	2.73	2.82	2.86	2.64	2.41	2.18
	3.00	3.91	3.64	3.41	2.18	1.95
	2.14	2.64	3.00	2.95	2.91	2.86
	1.67	1.90	1.48	1.43	1.90	1.62
	3.10	2.82	2.36	1.92	2.00	1.94
	3.64	3.73	4.23	4.00	4.00	3.50
	3.43	3.61	3.13	2.43	2.04	1.70
	2.26	2.91	3.35	3.30	3.13	2.90
	2.10	1.93	1.74	1.79	1.74	1.69
	1.64	1.77	2.32	2.77	2.77	2.55
	2.38	2.18	1.61	1.75	1.89	1.66

4

Aus diesen Daten wurden Gruppen gebildet (S. 920), die interessante Zusammenhänge aufzeigten:

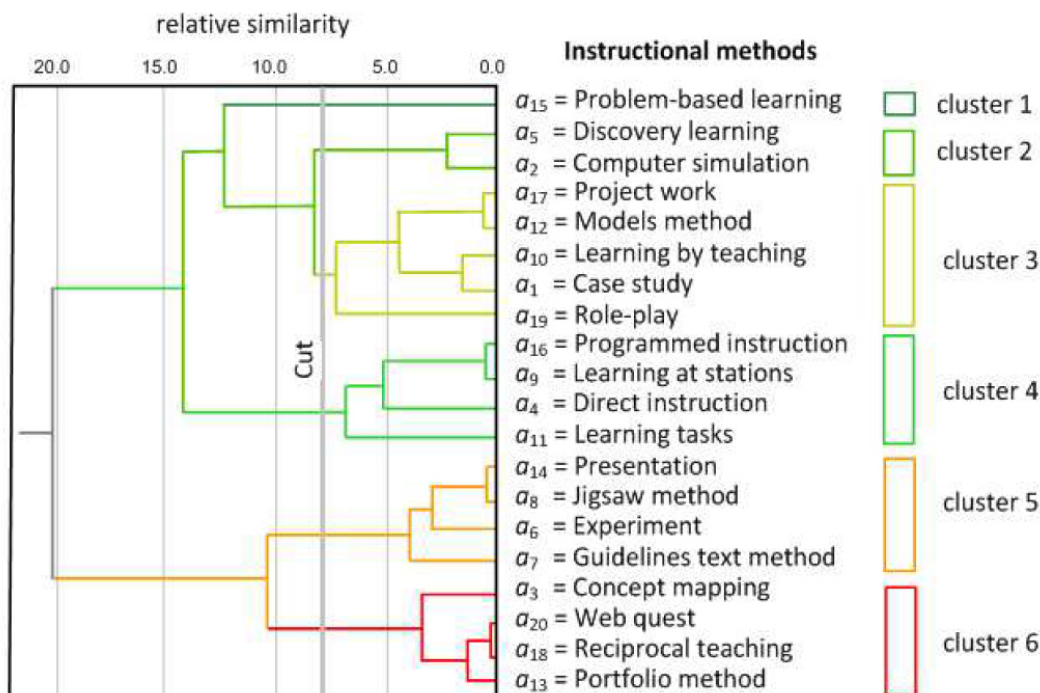


Fig. 3. Dendrogram and clusters of instructional methods (N = 24)

In einer anderen Darstellungsform, gut erfassbar farbcodiert, ergibt das

⁴ [Ze15], S. 927

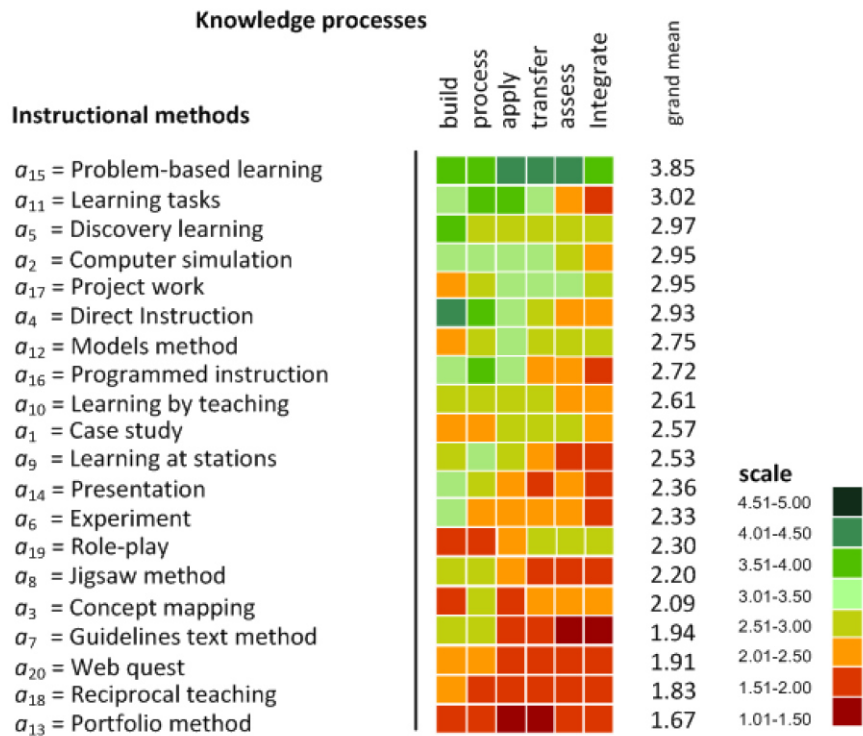


Fig. 2. Means of the instructional methods visualized for the knowledge processes ⁵

und im Buch ([Ze18], S. 136) in der deutschen Übersetzung

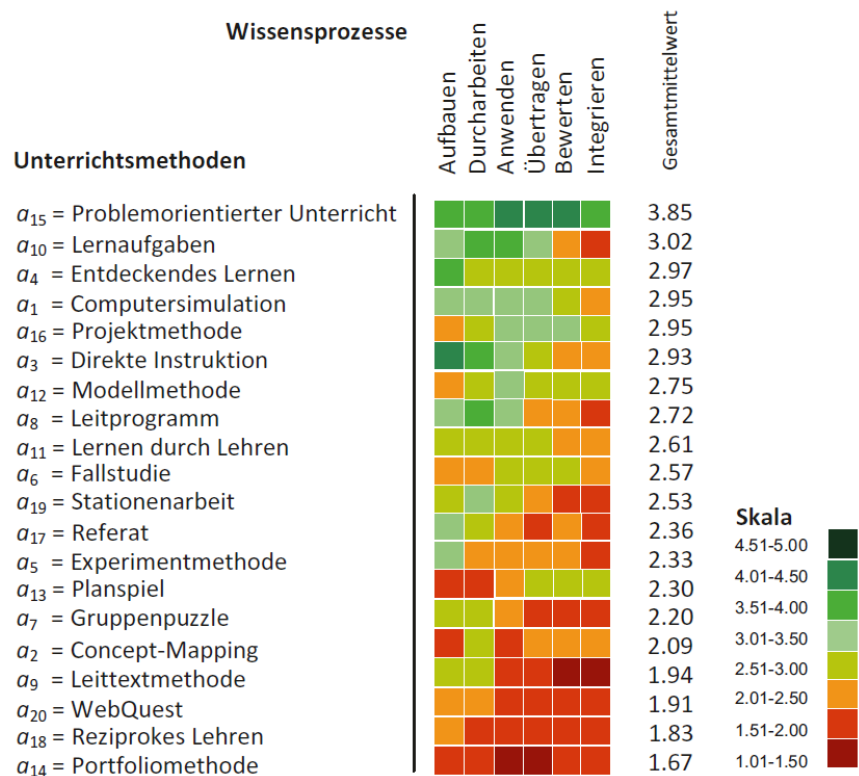


Abb. 4.2 Visualisierte Mittelwerte für die Wissensprozesse beim Lernvorgang

Es gibt also offenbar Methoden, die für die Informatik als besonders gut geeignet angesehen werden, während – in anderen Fächern bewährte – Methoden als wenig geeignet eingeschätzt werden. Zu

⁵ [Ze15], S. 920

beachten ist, dass eine Methode in aller Regel nicht in allen Phasen des Lernens gleich gut funktioniert. Der 'Problemorientierte Unterricht' („an instructional method enabling learners to acquire skills in the resolution of an exemplary problem which can then be transferred to other applicable problem areas“) sticht hier besonders hervor, er ist auch einzige Methode des "Premiumclusters 1", der praktisch in allen Lernphasen gut funktioniert. Der Ausspruch "machen hilft" bestätigt sich – egal in welcher Phase – hier auch wieder empirisch.

Auch zeigt sich in dieser Auswertung empirischer Daten, dass die seit langem irrational kritisch gesehene, geradezu verpönte – und damit jede praktische Lehrerfahrung gänzlich ignorierende und leugnende – 'direkte Instruktion', also das lehrkraftzentrierte Erklären („an instructional method with a central focus on the teacher. The teacher assumes the central role in directing the activities associated with the instruction and does not relinquish this role until the end of the learning process.“), als äußerst geeignet für die Phasen des Wissensaufbaus und der Vertiefung/Verarbeitung/Anwendung angegeben wird. Auch das *fragend-entwickelnde Unterrichtsgespräch* kann hierunter wohl verstanden werden. In der Praxis hat es sich bewährt, neue Inhalte zu Beginn kompakt zu erklären, damit der Lernende orientiert ist und dann in Übungs- und Übertragungsphasen vertiefend lernen kann. Keine Überraschung hier, aber ein empirischer Anker für eine bekanntermaßen bewährte Methode.

Es ist zu erkennen, dass die ersten drei Wissensprozesse (Aufbauen, Durcharbeiten, Anwenden) von mehr Methoden unterstützt werden als die anderen drei. Damit bietet sich auch direkt ein Ankerpunkt für eine Methodenerweiterung an. Welche Methoden könnten sinnvoll insbesondere für die Wissensprozesse Übertragen/Bewerten/Integrieren noch in den Methodenkanon aufgenommen werden? Der zentrale Aspekt der Aufgabenkultur kommt hier sicherlich gewinnbringend ins Rennen.

Insgesamt zeigt sich, dass bestimmte Unterrichtsmethoden mit Blick für den Informatikunterricht prädestiniert sind:

- problemorientierter Unterricht (problem-based learning),
- Lernaufgaben (learning tasks),
- Leitprogramm (programmed instruction),
- entdeckendes Lernen (discovery learning),
- Computersimulation (computer simulation),
- Projektmethode (project work) und
- direkte Instruktion (direct instruction), fragend-entwickelndes Unterrichtsgespräch.

Durch die breite Abdeckung der Wissensprozesse können diese Methoden auch als Leitmethoden, in anderen Begriffsverständnissen auch als Unterrichtsverfahren aufgefasst werden.

6

Problem-Based Learning

Problem-based learning (Abell and Lederman, 2007) is an instructional method enabling learners to acquire skills in the resolution of an exemplary problem which can then be transferred to other applicable problem areas.

Programmed Instruction

Programmed instruction is an instructional method (Canton, 2007) focusing on individualized material for the person learning to study on their own.

Computer Simulation

Computer simulation (Aldrich, 2009) comprises the application of simulation software for the virtual solution of (time-related) problems.

Learning Tasks

Learning tasks (Flewelling and Higginson, 2003) as an instructional method serve in initiating and guiding learning and thinking processes.

Discovery Learning

Discovery learning (Petty, 2009) is an instructional method with a central focus on the pupils in which learning recommendations are the focal point in order to motivate self-learning.

Project Work

The project work method (Branom, 1918) is an activities-oriented instructional method allowing learners to work on a defined objective in an organized, independent manner.

⁶ [Ze15], S. 295

Direct Instruction

Direct instruction (Petty, 2009) is an instructional method with a central focus on the teacher. The teacher assumes the central role in directing the activities associated with the instruction and does not relinquish this role until the end of the learning process.

Nicht jede ist für jede Phase des Lernens und bei jedem gleich gut, man wähle hier bedacht.

Obig sind die Methodenbeschreibungen aus dem Forschungsartikel ([Ze15]) abgebildet, um möglichst klare gemeinsame Begriffsvorstellungen zu sichern. Im entsprechenden Buch ([Ze18]) sind selbige nochmals ausführlicher und auf Deutsch formuliert. Auch hier sei darauf verwiesen, dass es primär die Definition des Autors ist und man nicht mit jedem Punkt oder festgelegtem diffizilen Unterschied konform gehen muss. So liest sich bspw. auf S. 41: „Unter Direkte Instruktion versteht man ein mehrstufiges Unterrichtsvorgehen. Häufig wird dies mit dem so genannten Frontalunterricht gleichgesetzt, was jedoch nicht richtig ist. (*) Direkte Instruktion ist geprägt durch Klarheit sowohl auf Seiten der Lehrperson als auch der Schüler im Hinblick auf die Ziele, Medien, Inhalte und Methoden. Dabei versucht die Lehrperson an das Vorwissen der Schüler anzuknüpfen.“

Der Text nach dem nachträglich eingefügten (*) hat natürlich nichts mit der Methode der direkten Instruktion zu tun sondern ist allgemeingültig für alle Methoden und Unterricht überhaupt. ‘Klarheit’ hat stets zu herrschen und möglichst an Vorwissen anzuknüpfen ist auch ebenfalls eine methodenunabhängige Grundtatsache jedes Lehrvorganges. Lesen Sie also mit Verstand, ein ebenfalls allgemeingültiger – und sogar methodenunabhängiger – Satz.

Genauer lässt sich aus den Daten der Studie ableiten, dass sich für die Informatik die *direkte Instruktion* – das fragend-entwickelnde Unterrichtsgespräch sei hier mit inkludiert – und das *problembasierte Lernen* als besonders geeignet erweisen, wenn es um den Aufbau von Wissen geht. Ergänzt um *Lernaufgaben* und *Leitprogramme* als Brücke zur Vertiefung, Anwendung (“scharf schalten“) und Festigung zeigt sich nun auch empirisch wie effektiv und effizient diese bewährten klassischen Methoden sind. Die *Projektmethode* kann dann für die “höheren“ Lernziele erfolgreich eingesetzt werden, nachdem die fachliche Basis gelegt ist. Nicht überraschend, denn sobald ein gewisses Fundament an Kenntnissen und Fähigkeiten vorhanden ist, kann der Lernende dieses aktiv anwenden, miteinander verknüpfen und damit sein Verständnis insbesondere der Zusammenhänge erheblich festigen, vertiefen und schärfen. Das gelingt erfahrungsgemäß, in dem an größeren Projekten gearbeitet wird, in dem – durch gutes Projektdesign seitens der Lehrkraft – die vorher erlernten Kenntnisse und Fähigkeiten nun in einem Gesamtzusammenhang anzuwenden sind.

Nachdem nun ein Kanon an bewährten Methoden informatikspezifisch zur Verfügung steht, sei nochmals betont, dass Methoden Werkzeuge sind, welche vom Handwerker (Lehrkraft) Zweckgebunden einzusetzen sind. Dies geschieht insbesondere vor dem Hintergrund

- der eigenen Zielsetzung
- der jeweiligen Lernphase
- der Lerngruppe
- der Rahmenbedingungen
- der eigenen Präferenzen

Der eigenen Präferenzen? Ja, der eigenen Präferenzen. Ein Aspekt, der so gut wie nie genannt wird. Methoden werden üblicher Weise mit Fokus auf die Lernenden betrachtet, nicht mit Blick auf die Lehrenden. Diese haben die Methode aber zum Leben zu erwecken und müssen von ihr überzeugt sein, wenn Unterricht – gegen die im realen Leben vorkommenden Widrigkeiten – tatsächlich die Chance haben soll zu gelingen. Das wiederum ist authentisch nur möglich, wenn die Methode zur Lehrkraft passt.⁷ Tatsächlich. Wie alles andere auch. Methoden sind weder gut noch schlecht, sie sind Werkzeuge, die fachgerecht angewendet werden müssen. Erinnern Sie sich regelmäßig daran.

⁷ Das heißt also insbesondere, dass Sie das Referendariat wirklich dazu nutzen sollten sich auszuprobieren. Aktivieren Sie Ihr theoretisches Wissen für die Konzeption von Stunden und Reihen, variieren diese Planung hinsichtlich der Methoden und probieren diese einfach aus. Nur so finden Sie heraus, was wann wie und warum in Ihrer Praxis funktioniert und was nicht. Jemand der von einer Methode nicht überzeugt

Wenngleich keine Methode, so soll in diesem Rahmen ein ganz zentraler Punkt von hochwertigem Unterricht angesprochen werden, da dieser mit der Methodenfrage eng zusammenhängt. Es handelt sich um den Aspekt der *Aufgabenkultur*. Es steht ein festes Zeitkontingent zur Verfügung und die Art der Aufgaben sowie der Umgang damit hat großen Einfluss auf die Qualität der zu erreichenden Fähigkeiten. Insbesondere über einen längeren – und damit erst realistisch wirklich sinnvollen – Zeitraum betrachtet, etwa die gesamte Oberstufenzeit. Welche immer wiederkehrenden Grundfertigkeiten gibt es, welche Aspekte sind verschiedenen Themengebieten gemeinsam, was an Werkzeug wird ständig benötigt, welche Denkkonzepte durchziehen die Informatik usw. usf. Die Antworten auf diese Fragen versuchen Sie dann auch in Form von Übungsaufgaben in den Köpfen der Lernenden zum Leben zu erwecken.

Für eine vertiefende Beschreibung der Methoden und des Themenkreises des informatikspezifischen Bezuges sei auf das Buch [Ze18] verwiesen, das u.a. aus den Ergebnissen der Studie [Ze15] entstanden ist.

Im Buch wird zudem noch eine weitere Untersuchung der Methoden hinsichtlich der Lerntheorien (Behaviorismus, Kognitivismus, Konstruktivismus) vorgestellt, die weitere Einsichten in die 'Warum-Frage' von Methoden bieten kann. Insbesondere zeigen sich hier schön die Unterschiede in der subjektiven Einschätzung der Bedeutung von Methode und Lerntheorie durch die Befragten.

Zendler schreibt als Schlussfolgerung:

>>Die Ergebnisse machen deutlich, dass sich Unterrichtsmethoden für den Informatikunterricht im Zusammenhang mit bestimmten Lerntheorien positionieren lassen: direkte Instruktion und Leitprogramm im Kontext der behavioristischen Lerntheorie; Experimentmethode, Lernen durch Lehren und Referat im Zusammenhang mit der kognitivistischen Lerntheorie; entdeckendes Lernen, Experimentmethode, Planspiel, problemorientierter Unterricht im Umfeld mit der konstruktivistischen Lerntheorie.<<([Ze18], S. 154)

Ein weiterer Überblick – gerade im Bereich der Mathematik, welche als eine der Vorgängerwissenschaften der Informatik auch methodisch mit ihr Ähnlichkeiten hat, gibt es viel Literatur – kann aus [BBL07] und [Ma08] gewonnen werden.

ist, wird diese auch nicht erfolgreich umsetzen können. Natürlich sind der Individualität auch Grenzen gesetzt, ein Tafelbild bspw. hat gewissen Kriterien zu genügen – u.a. Lesbarkeit, Struktur und fachliche Korrektheit –, das versteht sich aber von selbst.

Praktische Beispiele

Für die Praxis gilt: Je einfacher, desto besser. Verlassen Sie sich nicht zu sehr auf funktionierende Technik und vermeiden Sie 'Schnickschnack'. Methodisch wie inhaltlich und organisatorisch. Occam's Rasiermesser (Ockhams Razor⁸) als übergreifendes Prinzip – in Kombination mit Murphies Gesetz – gilt es zu beachten.

Im Folgenden sollen einige typische Unterrichtsszenarien aus der Informatik mit Blick auf mögliche Methodenanwendungen betrachtet werden. Dass es sich dabei nur um Vorschläge handelt muss nicht noch erwähnt werden.

Szenario 1: Einführung in die Programmierung (Einführungsphase Oberstufe)

Situation: Sie sollen Ihre Lerngruppe in die Programmiersprache Java unter BlueJ einführen.

Mögliches Vorgehen: Es kann kein Vorwissen vorausgesetzt werden und es gilt zügig genug "Fahrt aufzunehmen", um möglichst bald sinnvolle und dann eigenständig zu bearbeitende Übungsaufgaben stellen zu können. Dafür ist die *direkte Instruktion* das Mittel der Wahl.

Methodische Kommentare: Im Bereich der Programmierung steht die Waage zwischen Theorie und Praxis klar auf Seiten der Praxis. Gerade im Anfängerbereich ist so viel Theorie nicht notwendig, um an die Programmierübungen zu kommen. Hier erst wird das theoretische Wissen zum Leben erweckt, wird an Hand verschiedener praktischer Aufgaben durchdrungen und verknüpft. Die Wissensvermittlung an sich, also der Theorieteil, kann dabei effektiv und effizient durch direkte Instruktion und fragend-entwickelndes Unterrichtsgespräch stattfinden. Nutzen Sie die so gewonnene Zeit für mehr Übungen. Zudem bietet diese lehrkraftzentriertere Phase auch eine gewünschte methodische Abwechslung⁹ und ist von überschaubarer zeitlicher Dauer. Sie ist inhaltsreich und bereitet auf ausgedehnte praktische, schüleraktive Phasen vor.

Szenario 2: Einführung in die algorithmischen Grundstrukturen (Einführungsphase Oberstufe)

Situation: Sie haben die ersten Schritte in die Programmierung und die Programmierumgebung getan und Variablen und deren Datentypen vermittelt. Jetzt sollen die algorithmischen Grundstrukturen (Sequenz, Wiederholung, Entscheidung) eingeführt werden.

Mögliches Vorgehen: Eine *Impulsfrage* die Frage betreffend, was algorithmische Grundstrukturen sein könnten, bietet sich als Einstieg an. Ein kurzer *Lehrervortrag* nennt und erläutert diese dann. Dem könnte eine intensive *Einzelarbeitsphase* anhand eines *Leitprogramms* folgen, ergänzt um eine *Partnerarbeitsphase* zum Kenntnisabgleich. In das Leitprogramm integriert sind Fragen und kleine Übungsaufgaben, die in den Einzel-/Partnerarbeitsphasen zu bearbeiten sind. Musterlösungen werden explizit nach Bearbeitung zur Verfügung gestellt und sollen als Teil der Übungsaufgabe mit den eigenen Lösungen verglichen werden. Eine *Plenarphase* sichert die Endergebnisse, gefolgt von vielen Übungsaufgaben in der anschließenden, ausgedehnten *Übungsphase*.

Methodische Kommentare: Neben Variablen sind die algorithmischen Grundstrukturen das Fundament jedweder Programmierung. Daher ist größte Sorgfalt und Ausführlichkeit in dieser Phase notwendig, ebenso wie multiple Zugänge. Es darf hier keine Lücken im Verständnis geben, daher bietet sich methodische Vielfalt an, nicht als Alternative sondern als Pflichtprogramm. Dadurch soll die Chance erhöht werden, dass die Inhalte bei jedem auch ankommen und durch die leicht variierenden Zugänge wird gleichzeitig der geistig flexible Umgang mit den so zentralen Konzepten angestoßen. Eine

⁸ Beide Schreibweisen gibt es.

⁹ Im Gegensatz zum Eindruck, der manchmal aus didaktischer Literatur gewonnen werden kann, ist Methodenmonotonie auf alle Methoden bezogen. Auch eine ständige Gruppenarbeit oder ewig lange Projekte sind Methodenmonoton und werden erfahrungsgemäß eben auch von den Lernenden so empfunden, das gilt nicht nur für Frontalunterricht.

intensive Auseinandersetzung mit einem Grundlagentext/einem Leitprogramm in Einzelarbeit stellt Ruhe und Fokussierung sicher. Ferner wird Hochschulpropädeutisch gearbeitet und das verständige Lesen als tatsächlich fachunabhängige Fähigkeit weiter geschult. Die integrierten Übungen geben die Chance der Selbstkontrolle und der Austausch mit einem Partner sichert das nochmals ab. Übungen und Plenumsphase bieten weitere Gelegenheit dazu. Insgesamt also geplant mehrfach redundant.

Szenario 3: Integrierende Programmierübung (Einführungsphase Oberstufe)

Situation: Sie haben Datentypen, Variablen, algorithmische Grundstrukturen und Klasse/Objekt thematisiert und möchten diese nun durch Übung weiter vertiefen.

Mögliches Vorgehen: *Problemorientierter Unterricht* an Hand des Beispiels einer BMI-Waage. In kurzer Aufgabenstellung wird die Aufgabe beschrieben, eine BMI-Waage (body-mass-index) zu programmieren. Dabei wird die Konsolen-Ein-/Ausgabe als zweiter Teil der Aufgabe beschrieben, feste Werte ohne Benutzerinteraktion bilden den ersten Teil der Aufgabe.

Methodische Kommentare: Diese Aufgabe ist schon recht komplex und kann diesbezüglich über den Detailgrad an zur Verfügung gestellten Informationen gesteuert werden. Ist dieser knapp gehalten, müssen die Lernenden zunächst eine gezielte Internetrecherche zum Thema BMI-Index durchführen, um die notwendigen Daten sowie die Berechnungsformel zu ermitteln. Anschließend sind Designentscheidungen zu treffen und das auch vor dem bekannten zweiten Teil der Aufgabe, der Benutzerinteraktion ergänzt und damit die Lösung des ersten Teils gezielt erweitert. Anschließend sind alle obig genannten programmiertechnischen Dinge anzuwenden. In natürlicher Weise wird – ohne Zutun der Lehrkraft – hier eine informatiktypische Form der Partnerarbeit stattfinden, man “schaut sich zwischendurch mal in der Nahumgebung um“. Die Lehrkraft hat hier primär – da Einführungsphase, also Anfängerbereich – kleinere Fehler zu beheben und ansonsten primär darauf zu achten, dass niemand strandet oder sich gänzlich verfängt. Nicht viel Redeanteil aber eine hohe notwendige Aufmerksamkeit. Der Übergang zwischen *problemorientiertem Unterricht* und der *Projektmethode* ist in der Informatik fließend. Für die leistungsschwächeren Lernenden fühlt sich so eine Aufgabe schon eher wie ein Projekt an, während die schnellen Leute damit ggf. zügig durch sind (Zusatzaufgaben immer im Hinterkopf oder besser gleich direkt als solche auf dem Übungsblatt explizit formuliert parat haben).

Diese Art der Übungsphase kann als **Typisch für den Informatikunterricht** bezeichnet werden und ein guter Aufgabenentwurf bestimmt hier in hohem Maße die Arbeitsatmosphäre, Effektivität, Effizienz und auch die (dauerhaftere) Zufriedenheit aller Beteiligten, inklusive der Lehrkraft.

Szenario 4: Entwicklung eines 2D-Shooters (Kursphase Oberstufe)

Situation: Als Abschlussprojekt der Programmierausbildung soll ein Spiel programmiert werden.

Mögliches Vorgehen: Der Musterfall für die *projektorientierte Methode*.

Methodische Kommentare: ‘Projekte’ hören sich zunächst gut an, sind methodisch aber sehr anspruchsvoll in der Durchführung. In solchen umfangreicheren Projekten kann unschätzbare Programmiererfahrung gewonnen werden, gerade durch die Effekte, die erst ab einem gewissen Komplexitätsgrad auftreten. Jedoch, behalten Sie das Ziel der Ausbildung (Abitur(prüfung)) im Auge. Solche Projekte bekommen ganz schnell ein Eigenleben, führen zu Frustration – was erstmal nichts Schlimmes ist, im Gegenteil, es ist hier realistisch und äußerst lehrreich – und sind vor allem nicht selten wenig Zielführend wenn es um die Abiturprüfung geht. Setzen Sie daher die Projektgröße realistisch an und definieren Sie vor allem viele und ganz klar definierte *Meilensteine*. Ein reines „macht mal“ funktioniert nicht.

Literatur

[BBL07]

Barzel, B.; Büchter, A.; Leuders, T.: Mathematik Methodik, Cornelsen Verlag, 2007

[Br77]

Bruner, J.S.: The process of education – A landmark in educational theory, Cambridge University Press, 1960, Zweidruck mit neuem Vorwort 1977

[CBH91]

Collins, A., Brown, J. S., Holum, A.: Cognitive apprenticeship: Making things visible, American Educator, 15(3), 6-11, S. 38-46, 1991

[CBN87]

Collins, A., Brown, J. S., Newman, S. E.: Cognitive apprenticeship: Teaching the craft of reading, writing and mathematics (Technical Report No. 403). BBN Laboratories, Cambridge, MA. Centre for the Study of Reading, University of Illinois. January, 1987

[HBZ13]

Hattie, J.; Beywl, W.; Zierer, K.: Lernen sichtbar machen, Schneider Verlag, 2013

[Ha09]

Hattie, J.: Visible learning. Routledge Verlag, 2009

[Ma08]

Mattes, W.: Methoden für den Unterricht, Schöningh Verlag, 2008

[Ze18]

Zendler, A.: Unterrichtsmethoden für den Informatikunterricht, Springer Vieweg Verlag, 2018

[Ze15]

Zendler, A.: Instructional methods to computer science education as investigated by computer science teachers, Journal of Computer Science, August 2015
DOI: 10.3844/jcssp.2015.915.927

Bildquellen und Internetquellen sind als Fußnote am jeweiligen Ort angegeben.

Übungen

Aufgabe 1:

Aufgabe 2:

Aufgabe 3: