

# Zielorientierte Didaktik der Informatik – Kompetenzvermittlung bei engen Zeitvorgaben

Nicole Weicker

Lehrbeauftragte für Didaktik der Informatik  
an der Universität Leipzig  
nicole@weicker.de

**Abstract:** Die Informatiklehre steht vor der Aufgabe, in kürzerer Zeit mehr Kompetenzen zu vermitteln. Die zielorientierte Didaktik der Informatik bestehend aus den didaktischen Prinzipien Lernzielorientierung, Anknüpfung an Vorwissen, enge Verzahnung von Input und Aktivität, Motivierung und frühzeitiges Feedback bzgl. aller Lernziele wird allgemein und in ihrem Einsatz in der Hochschullehre vorgestellt.

## 1 Einleitung

Die Rahmenbedingungen für die Vermittlung von Informatik an Hochschulen haben sich in den letzten Jahren stark verändert: Auf der einen Seite sollen mehr Kompetenzen vermittelt werden, da sich die Informatik von einer vorrangigen Erforschung der Möglichkeiten der Informationsverarbeitung zur Dienstleistungswissenschaft mit starkem Ingenieurscharakter für nahezu alle anderen Disziplinen entwickelt hat. Auf der anderen Seite sind die zeitlichen Vorgaben, in denen die Lernenden diese Kompetenzen erwerben sollen, verschärft worden [GI05]. Durch die politisch gewollte Vorgabe, 70% der Studierenden nach dem Bachelorabschluss ins Berufsleben zu schicken, entsteht die Notwendigkeit, die entsprechenden berufsbefähigenden Kompetenzen in wenigen Semestern zu vermitteln. Diesen Herausforderungen hat sich die Informatiklehre ohne größere Personalaufstockungen zu stellen.

Um der Anforderung – Vermittlung von mehr Kompetenzen in Informatik in kürzerer Zeit – gerecht werden zu können, ist die Didaktik der Informatik gefragt, neue passende Antworten zu finden. Die in dieser Arbeit vorgestellte zielorientierte Informatiklehre ist eine mögliche Reaktion auf diese Herausforderungen. Sie beinhaltet die enge Kombination der didaktischen Prinzipien: Lernzielorientierung, Anknüpfung an Vorwissen, enge Verzahnung von Input und Aktivität, Motivierung und frühzeitiges Feedback bzgl. aller Lernziele. Die Umsetzung der zielorientierten Informatiklehre wird an fünf Beispielen aus der Hochschullehre aufgezeigt: Workshopseminar, kooperatives Lernen im Übungsbetrieb des Grundstudium, aktivierende Methoden in Großveranstaltungen, Kombination von Spezialvorlesung und Übung sowie ideenorientiertes Spiralkonzept für Vorlesungen. Allen Beispielen ist gemein, dass sie ohne zusätzliches Personal durchgeführt wurden und sich der zeitliche Aufwand für die Lehrenden im Rahmen hielt.

## 2 Didaktische Antworten und ihre Grenzen

Zwei wichtige Lerntheorien, die kombiniert vielen didaktischen Aspekten der zielorientierten Lehre zugrundeliegen, sind der Kognitivismus und der Konstruktivismus [KW01].

### **Kognitivistisches Lernen**

Dem kognitivistischen Lernmodell liegen Modelle der neuronalen Verarbeitung von Wissen zugrunde. Lernen wird im Kognitivismus als streng regelhaft verlaufender Prozess des Wissenserwerbs verstanden, der erfolgreich von Außen gesteuert werden kann. Die Lerninhalte werden als fertiges System vom Lehrenden vermittelt. Die Lernenden sind während dieser Vermittlung passiv. Beim kognitivistischen Ansatz steht der Wissensinput durch den Lehrenden im Mittelpunkt. Das eigentliche Lernen findet dabei in der Regel nicht während der Lehrveranstaltung sondern in Eigenarbeit statt. Ein hauptsächlich kognitivistisches Vorgehen impliziert, dass die Lernenden unabdingbar viel Zeit zum Reifen ihres kognitiven Wissens benötigen. Kompetenzen in affektiver und pragmatischer Hinsicht werden vom Kognitivismus nicht explizit gefördert. Ihr impliziter Erwerb kann damit nicht von allen Lernenden im gleichen Maße vorausgesetzt werden.

Für die Informatiklehre ist die strukturierte Wissensvermittlung in Form von Frontalunterricht in Vorlesungen insbesondere im üblichen Massenbetrieb in den ersten Semestern unverzichtbar. Um jedoch in kürzerer Zeit (sprich mit weniger individueller benötigter Reifungszeit) kognitive Kompetenzen erreichen zu können, sind die Prinzipien des Kognitivismus um weitere didaktische Elemente zu erweitern.

### **Konstruktivistisches Lernen**

Im konstruktivistischen Ansatz wird Wissen als individuelle Konstruktion aufgefasst. Lernen ist vor diesem Hintergrund ein aktiver, konstruktiver Prozess in einem bestimmten Handlungskontext. Dazu hat die Lernumgebung vor allem Situationen anzubieten, in denen der Lernende eigene Konstruktionsleistungen erbringen kann. Unterrichten bedeutet in diesem Kontext mehr Unterstützen, Anregen und Beraten. Der Lehrende tritt in einer reaktiven Rolle hinter der aktiven Rolle des Lernenden zurück.

In diesem lerntheoretischen Modell findet das Lernen während der Veranstaltungen statt. Dadurch können die Lernenden neues Wissen direkt in ihrem individuellen Vorwissen verankern. Durch die eigene Auseinandersetzung wird Praxis und Wissensbildung miteinander verbunden. Zusätzlich hat jeder Lernende die Möglichkeit, eigene, dem eigenen Lerntyp angepasste Lernwege zu beschreiten. Allerdings bedeutet diese Art des Vorgehens, dass im Zweifel jeder Lernende „das Rad neu erfinden“ muss. Die Lerngeschwindigkeiten der Lernenden sind in aller Regel sehr unterschiedlich und die inhaltlichen Ziele sind im Vorherein nicht abschätzbar. Zusammenfassend ist der Konstruktivismus sehr zeitaufwändig. Durch zusätzliche Wissensinputphasen (wissensbasierter Konstruktivismus) ist es möglich, dass die Lernenden zumindestens auf einer Wissensbasis aufbauend lernen können, statt sich alles selbst erarbeiten zu müssen. Dennoch bleiben die anderen Nachteile des Konstruktivismus bestehen.

Im Rahmen der Informatiklehre ist der Konstruktivismus trotz seiner unbestreitbaren Nachteile von wichtiger Bedeutung, da durch konstruktivistische Lernphasen insbesondere pragmatische und affektive Lernziele angestrebt werden können, die im Kognitivismus der individuellen Auseinandersetzung des Einzelnen überlassen bleiben.

### 3 Zielorientierte Didaktik

Die im weiteren beschriebene zielorientierte Didaktik für Informatik folgt der Anforderung, in kürzerer Zeit fundierte Kompetenzen in allen Bereichen der Informatik zu vermitteln. Während der wissensbasierte Konstruktivismus kognitivistische Ansätze (Wissensinput) mit der Theorie des Konstruktivismus verbindet, verknüpft die zielorientierte Didaktik kognitivistisches Vorgehen, wann immer kognitive Lernziele angestrebt werden, mit konstruktivistischen Aspekten für die pragmatischen und affektiven Lernziele.

#### Lernzielorientierung

Ein wichtiges Standbein der zielorientierten Didaktik ist die Bestimmung der jeweils angestrebten Lernziele bzgl. aller Lernbereiche, damit ausgehend von den Lernzielen die Inhalte und deren Strukturierung festgelegt und geeignete Lehrmethoden bestimmt werden können.

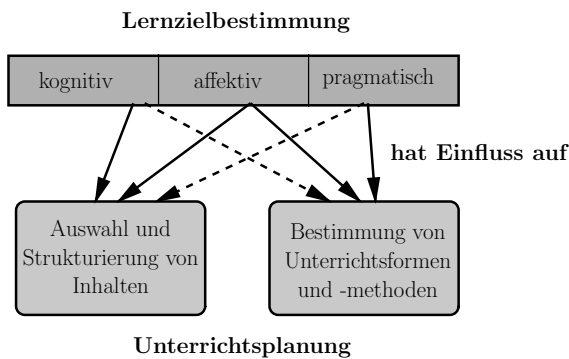


Abbildung 1: Zusammenhang zwischen Lernzielbestimmung und Unterrichtsplanung

Wie in Abbildung 1 dargestellt, beeinflussen vorwiegend die kognitiven und affektiven Lernziele die Auswahl der Inhalte und deren Strukturierung, während die affektiven und pragmatischen Lernziele bei der Bestimmung geeigneter Unterrichtsformen und -methoden berücksichtigt werden sollten. Didaktische Anhaltspunkte für die Auswahl von Inhalten bietet das Konzept des exemplarischen Lernens. Mögliche Strukturierungen sind ideen- oder problemorientierte Spiralgliederungen von Vorlesungen. Unterrichtsformen können neben Frontalunterricht und Übungsgruppen auch die Kombination von Vorlesung und Übung sein. Eine weitere Unterrichtsform, die in der Informatiklehre von besonderer Bedeutung ist, ist die Organisation von Lern- oder Arbeitsgruppen über kooperative Kon-

zepte. Unterrichtsmethoden sind beispielsweise der Einbau von Lernstopps in den Vorlesungsfluss oder Gruppenarbeitsphasen im kombinierten Vorlesungs-/Übungskonstrukt.

### **Anknüpfung an Vorwissen**

Eine weitere wichtige Komponente der zielorientierten Didaktik ist die explizite Thematisierung und Nutzung von vorhandenem Vorwissen. Aus der Gehirnforschung ist bekannt, dass sinnvolles Lernen nur unter Einbeziehung des individuellen Vorwissens möglich ist [Ve04]. Ein großes Problem gerade in großen Veranstaltungen wie z.B. die Vorlesung „Einführung in die Informatik“ im ersten Semester des Informatikstudiums besteht darin, dass die Vorkenntnisse der Studierenden unbekannt sind. Zusätzlich besuchen nicht alle Studierenden die gleichen Parallelveranstaltungen, so dass sich der Dozent nicht darauf verlassen darf, dass bestimmte Inhalte in anderen Vorlesungen wie z.B. „Mathematik für Informatiker“ behandelt werden. Die wesentlichen Aspekte zur Nutzung von Vorwissen ist die Feststellung des Vorwissens, eventuell die Schaffung von Vorerfahrungen und die tatsächliche Aktivierung des Vorwissens.

Um explizit Vorwissen aktivieren zu können, ist es notwendig, dass der Dozent die unterschiedlichen Vorwissens- und Vorerfahrungstiefen der Studierenden erfragt. In kleinen Veranstaltungen kann dies direkt abgefragt und beispielsweise an der Tafel festgehalten werden. In größeren Veranstaltungen kann Vorwissen über Zettelabfrage, sprechende Wand (Wandplakate, in die die Studierenden über Klebepunkte oder Kreuze ihre Vorerfahrungen in vorgegebene Kategorien eintragen) oder über die Nutzung von Online-Tools (z.B. OPAL an der HTWK Leipzig) erfragt werden. Interessant ist für die „Einführung in die Informatik“ unter anderem in welcher Programmiersprache in welchem Umfang Kenntnisse vorhanden sind und welche Methoden und Darstellungformen der objektorientierten Modellierung bei wie vielen Studierenden vorausgesetzt werden können.

In vielen Fällen ist auch ohne explizite Befragung Vorwissen z.B. aus früheren Veranstaltungen bekannt. Dennoch genügt es nicht, dass das Vorwissen theoretisch vorhanden ist. Vielmehr ist es notwendig, das vorhandene Vorwissen in Gedächtnis zu rufen und damit zu aktivieren. Nur dann können die neuen Inhalte und Methoden direkt mit dem Bekannten verknüpft werden. Anderenfalls findet diese Verknüpfung zu einem individuell unterschiedlich späteren Zeitpunkt statt. Im Sinne der zielorientierten Didaktik wird angestrebt, dass derartige Verknüpfungen so bald wie möglich hergestellt werden. Aktiviert werden kann Vorwissen z.B. über eine kurze Wiederholung. Je aktiver die Studierenden bei dieser Wiederholung sind, desto wahrscheinlicher liegt das Vorwissen danach präsent im Gedächtnis vor.

Eine andere Form der Nutzung von Vorwissen ist die explizite Schaffung von Vorerfahrung auf die im weiteren Verlauf der Veranstaltung direkt zurückgegriffen wird.

### **Enge Verzahnung von Input und Aktivität**

Für eine zielorientierte Didaktik ist es notwendig, den Abstand zwischen kognitivem Input zu bestimmten Inhalten und deren praktische Umsetzung so gering wie möglich zu halten. Durch die Praxisphasen können die kognitiven Inhalte besser im Gedächtnis verankert und

um eigene Erfahrungen vertieft werden. Zusätzlich tauchen dadurch Fragen früher auf und Verständnisprobleme können gezielt angegangen werden.

Aus diesem Grund sollten in jede Veranstaltung Praxismöglichkeiten integriert werden. In Vorlesungen kann dies über aktivierende Methoden (z.B. Lernstopp oder Bienenkorb) umgesetzt werden. Bei kleineren Veranstaltungen kann dies über eine Kombination von Vorlesung und Übung stattfinden ( $4(V+Ü)$  statt  $2V + 2Ü$ ).

Im Kontext der Informatiklehre besteht insbesondere bei theoretischen oder abstrakten Konzepten das Problem, dass die Studierende das Konzept in der Vorlesung zwar nachvollziehen können, zuhause bei einer eigenen Umsetzung in Form der Lösung von Übungsaufgaben daran scheitern. Der Abstand zwischen Vorlesung und der praktischen Übung durch Übungsaufgaben ist häufig so groß, dass kognitive Inhalte bzw. das in der Vorlesung vorhandene Verständnis bereits wieder vergessen wurden. Im Gegensatz zu anderen Fächern ist in der Informatik (fast) immer ein Praxisbezug vorhanden, der explizit herausgearbeitet werden sollte.

## Motivierung

Unter Motivierung wird in dieser Arbeit in erster Linie die Motivierung der Lernenden und nicht die Einführung eines Themas verstanden. Die wesentlichen Einflussfaktoren (siehe Abbildung 2), durch die Lehrende die Lernenden motivieren können sind

- Vermittlung von Begeisterung durchs eigene Vorbild
- ansprechende (interessante) Themen
- Alltagsbezug aus der Lebenswelt der Lernenden
- Übertragung von Verantwortung (beispielsweise innerhalb eines stark selbstbestimmten Projekts)
- sozialer Kontext (Gruppenarbeit oder Wettbewerbssituationen)
- Scheinbedingungen oder Prüfungsleistungen.

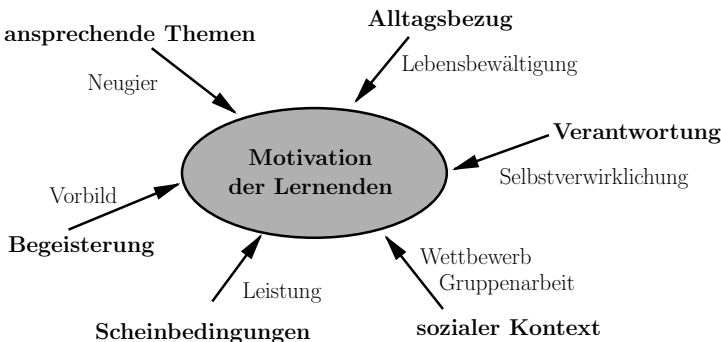


Abbildung 2: Einflussfaktoren zur Motivierung von Lernenden

Besonders die letztere Motivierungsmöglichkeit sollte nicht außer Acht gelassen werden. Auch wenn es sich dabei um eine nicht inhaltlich begründete Motivation handelt, kann diese hilfreich sein, so dass die Lernenden sich überhaupt auf die Aufgaben oder eine aktive Auseinandersetzung mit den Inhalten einlassen und die anderen Motivierungspunkte greifen können.

### Frühzeitiges Feedback bzgl. aller Lernbereiche

Ein wichtiger Bestandteil der zielorientierten Lehre ist die frühzeitige Rückmeldung an die Lernenden, in welchen Punkten sie wie gut sind und welche Verbesserungsmöglichkeiten es noch gibt. Ein derartiges Feedback auf den Leistungsstand erfolgt in der üblichen kognitiven Lehre durch die Rückmeldungen auf die Lösung von Übungsaufgaben und durch Prüfungsergebnisse. Diese Feedback-Praxis hat einige Nachteile. Zum einen gelingt es immer wieder Studierenden trotz z.T. scharfer Scheinbedingungen sich durch die Übungsaufgabenverpflichtungen zu „mogeln“, sei es durch die Mitarbeit in einer Gruppe, bei der sie nicht tatsächlich zur Lösung der Aufgaben beitragen oder sei es durch Abschreiben von fertigen Lösungen. Auf diese Art und Weise bringen sich diese Studierende selbst um eine Rückmeldung auf ihre tatsächliche Leistungen und bei den Prüfungen werden sie von ihrem eigenen Unvermögen überrascht. Auf der anderen Seite erhalten die Studierenden kaum oder keine Rückmeldung auf ihren Stand bzgl. affektiver oder pragmatischer Lernziele.

Die Basis zu einem funktionierendem Feedbackprozess besteht darin, dass der Lehrende alle Lernziele seiner Veranstaltung für sich kleinschrittig und bzgl. aller Lernebene formuliert. Dies sollte in aller Regel vor der eigentlichen Veranstaltung stattfinden. Während der Durchführung der Veranstaltung sollte der Lehrende die jeweils aktuellen Lernziele thematisieren und längerfristige Lernziele immer wieder wiederholen, so dass diese den Lernenden präsent sind. Zusätzlich sollte jeder Lernende regelmäßig alle paar Wochen ein Feedback auf seine Fähigkeiten (kognitiv), Fertigkeiten (pragmatisch) und Haltungen (affektiv) erhalten. In Abbildung 3 werden beispielhaft einige Möglichkeiten aufgezeigt, wie ein regelmäßiges Feedback auch in größeren Veranstaltungen möglich ist.

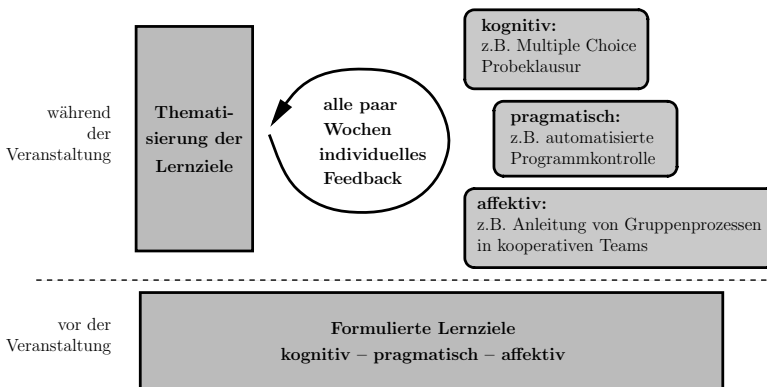


Abbildung 3: Funktionierender Feedbackprozess auch für große Vorlesungen mit Übungsbetrieb

## 4 Beispielhafte Umsetzungen in der Informatiklehre

In diesem Abschnitt wird anhand von einigen konkreten Beispielen vorgeführt, wie die sich zielorientierte Didaktik der Informatik konkret umsetzen lässt. Alle diese Beispiele sind an Universitäten und/oder Fachhochschulen exemplarisch umgesetzt worden. Für viele der Beispiele gibt es Evaluationen über Fragebögen.

### Seminar als Workshop

Die wesentlichen Lernziele eines Seminars sind neben der vertieften Beschäftigung mit fachlichen Inhalten die folgenden.

- wissenschaftliche Texte (zunehmender Komplexität) lesen und verstehen,
- fundierte Literaturlarbeit (Recherche) durchführen,
- wissenschaftliche Inhalte in eigenen Worten verständlich wiedergeben,
- die Fachsprache Informatik verwenden,
- eine wissenschaftliche Arbeit präsentieren,
- passende Präsentationsmedien einsetzen,
- die Verteidigung der eigenen Arbeit üben (für Studien-, Bachelor-, Master- oder Diplomarbeit im Studium, für eine eigene Arbeit gegenüber Kollegen, Geldgebern, Kunden, Vorgesetzten im späteren Arbeitsleben),
- eine Ausarbeitung erstellen, die die wissenschaftliche Arbeit, beschreibt, erklärt und kritisch hinterfragt,
- die eigene Ausarbeitung und Vortrag kritisch beurteilen.

Im Bestreben, das Erreichen dieser Lernziele innerhalb eines Seminars an der Hochschule so effektiv wie möglich zu fördern, wurde an jeweils zwei Universitäten und Fachhochschulen ein spezielles Workshopseminarkonzept durchgeführt [WDW06]. In Anlehnung an übliche wissenschaftliche Workshops und Konferenzen findet das eigentliche Vortragsseminar als Blockveranstaltung gegen Ende des Semesters statt. Zusätzlich zu den üblichen Aktivitäten wie der Einreichung eines Exposés, dem Peer-Review, dem Verfassen einer wissenschaftlichen Ausarbeitung für den Tagungsband und dem eigentlichen Vortrag finden eine Reihe von Vorträgen des Dozenten statt, in dem Wissen zu ausgewählten Kompetenzen vermittelt wird. Im Laufe des Semesters werden die Studierenden parallel zu ihrer eigenen Tätigkeit bzgl. der jeweils anfallenden Arbeitsaufgaben geschult. In Abbildung 4 sind die drei Ebenen dargestellt, die das Workshopseminar charakterisieren: Input, Praxis und Feedback. Die gestrichelten Elemente zeigen, welche Elemente nur bei einigen Workshopseminaren an den Universitäten Stuttgart und Leipzig, der FH Braunschweig-Wolfenbüttel und der HTWK Leipzig durchgeführt wurden. Benotet wurden jeweils die endgültige Ausarbeitung, die Gutachten, der Vortrag und die Teilnahmen an den Diskussionen im Verhältnis 30:30:30:10.

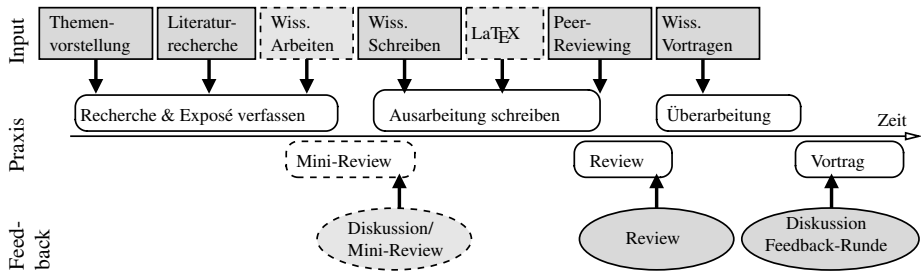


Abbildung 4: Ablauf des Seminars als Workshop

Inhaltlich erhalten die Studierenden je nach Titel des Seminars einen speziellen Auftrag. Im Seminar „Informatik und Gesellschaft“ an der Universität Leipzig sollen sie beispielsweise gegensätzliche Positionen zu ihren speziellen Vortragsthema gegenüberstellen. Im Seminar „Evolutionäre Algorithmen“ oder „Selbstreproduzierende Programme, Viren und Würmer“ an der HTWK Leipzig sollen sie die Techniken, die sie vorstellen, selbst umgesetzt haben und über diese Umsetzung berichten. Im Fachseminar „Didaktik der Informatik“ ist die Aufgabe der Studierenden Unterrichtsentwürfe aus der Literatur vorzustellen und anschließend eine eigene Position dazu zu beziehen bzw. einen alternativen Unterrichtsentwurf vorzulegen. Die Kernaufgabe wird thematisiert als die aktive Aufbereitung von Literatur und der Frage des Mehrwerts der Seminararbeit: „Warum sollte jemand die Seminararbeit statt der Originalliteratur lesen?“

Bereits früh kann im Unterschied zu anderen Seminaren festgestellt werden, dass sich die Studierenden aktiver mit ihren Themen und insbesondere mit der Literatur auseinandersetzen. Die Diskussion der Exposés wird von den Studierenden als anstrengend wahrgenommen. Auf der anderen Seite wird von vielen das Feedback begrüßt, dass ihnen hilft, ihre eigene Position zu schärfen. Die Studierenden sind trotz des Mehraufwands stark motiviert und die Ausarbeitungen und Vorträge sind qualitativ besser als übliche Seminarbeiträge. Zusätzlich fällt es leichter, inhaltliche Diskussionen anzuregen, da die verschiedenen Vortragsthemen durch die Vorstellung der Zusammenfassungen allen bekannt sind und jede Ausarbeitung von mehreren Studierenden begutachtet wurde. Im Feedback am Ende der Seminare wird einheitlich die inhaltliche Vermittlung von Grundlagen zur Wissenschaftsarbeit sowie das intensive Erleben am Workshop-Tag gelobt.

Der Mehraufwand für den Dozenten ist die Vorbereitung und das Halten der Inputsequenzen. Dieser Aufwand amortisiert sich bei Wiederholung dieses Konzepts schnell. Weitere zusätzliche Zeit ist für die Diskussion der Exposés sowie die Bewertung der Gutachten notwendig.

In diesem Beispiel der zielorientierten Didaktik wurden die Konzepte der Lernzielorientierung, der engen Verzahnung von Input und eigener Aktivität sowie einem frühzeitigen und umfassenden Feedback direkt durch organisatorische und strukturelle Vorgaben umgesetzt. Neben einer Motivation über den sozialen Kontext der Seminargruppe kann über ansprechende Themen und Alltagsbezug motiviert werden. Allerdings hat auch die extrinsische Motivation über die Benotung einen wesentlichen Anteil.



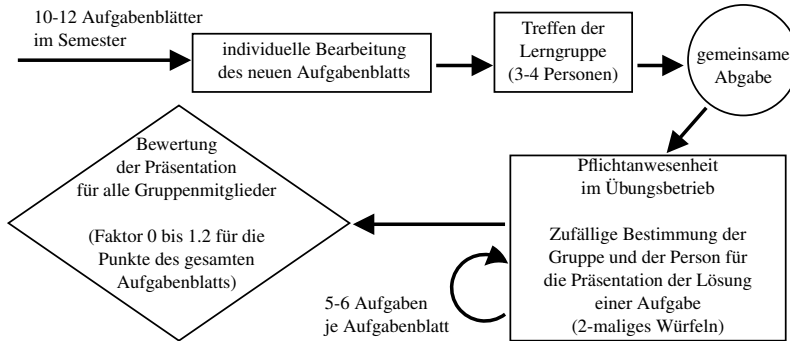


Abbildung 5: Ablauf des kooperativen Lernens im Übungsbetrieb

## Kooperatives Lernen im Grundstudium

In den ersten Semestern eines Informatikstudiums sind die Einführungsveranstaltungen häufig durch große Studierendenzahlen, verschiedene Studiengänge und z.T. stark heterogenen kulturellen Hintergrund der Studierenden gekennzeichnet. Je nach Studienort fällt es den Studierenden leichter oder schwerer, sich selbstorganisierend zu Arbeits- und Lerngruppen zusammenzufinden. Aus individuellen Befragungen von Studierenden in mündlichen Nachprüfungen im Studiengang Informatik hatten viele im Studium keine Kommilitonen, mit denen sie zusammen gearbeitet oder gelernt hätten. Neben der Übung fachlicher und methodischer Informatikinhalte im Übungsbetrieb sollen die Studierenden in den ersten Semestern lernen, sich im Studium selbst zu organisieren und individuell effektive Lernstrategien anzuwenden. Daneben ist es wichtig, dass sie sozialen Kontakt zu ihren Kommilitonen aufbauen und lernen, im Studium von einander zu profitieren (Entwicklung von Kommunikation- und ersten Teamkompetenzen).

An der Universität Stuttgart wurde im WiSe 2005/06, SoSe 2006 ein verpflichtend kooperativer Übungsbetrieb zur Veranstaltung „Einführung der Informatik I-II“ mit knapp 300 Studierenden eingeführt. Das Konzept (positive Abhängigkeit innerhalb von vorgegebenen Kleingruppen, individuelle Verantwortlichkeit für die eigene Arbeit und das Produkt der Kleingruppe, direkte Zusammenarbeit in der Kleingruppe, Notwendigkeit der Entwicklung von sozialen Kompetenzen und regelmäßige Gruppenreflexionen) [FB01], seine Durchführung (siehe Abbildung 5) und Ergebnisse von Akzeptanzbefragungen sind ausführlicher in Weicker et al. [WDW06] beschrieben. Den dort berichteten Erfahrungen ist noch hinzu zu fügen, dass die Studierenden im folgenden Semester in der Veranstaltung „Einführung in die Informatik III“ weiter kooperativ zusammen arbeiteten und ohne Scheinzwang die Übungsaufgaben weiterhin lösten. Im Vergleich zu früheren Jahren erwarben ca. 60% (sonst üblich 15–20%) der Studierenden den Schein zu dieser Veranstaltung.

## Aktivierende Methoden

Insbesondere in großen Vorlesungen, in denen Frontalunterricht zwangsläufig die bestimmende Unterrichtsform ist, kann über kurze „Denk- bzw. Diskussionspausen“ eine enge

Verzahnung von Input und Aktivität erreicht werden. Dabei stellt der Dozent eine kurze Frage oder Aufgabe, die die Studierenden entweder allein oder in Zweier- bis Dreiergruppen lösen sollen [FB03]. Dabei gibt es zwei wichtige Prinzipien, die bei der Durchführung derartiger aktivierender Methoden zu beachten sind, damit tatsächlich eine aktivierende Wirkung auf möglichst viele Studierende erreicht wird. Auf der einen Seite sollte einer solche Pause nicht länger als 2–3 Minuten angesetzt werden, damit die Diskussionen in den Kleingruppen sachbezogen bleiben. Wer in dieser Zeit keine Idee/Lösungsansatz o.ä. gefunden hat, wird auch nach 10 Minuten nicht weiter sein. Auf der anderen Seite ist es entscheidend, dass der Dozent direkt Studierende um ihre Lösung bittet. Wenn nur die Studierenden aufgerufen werden, die sich selbst melden, werden die übrigen beim nächsten Mal wenig Veranlassung haben, sich selbst Gedanken zu machen. Besonders gut ist es, mehrere Studierende auch aus den hinteren Reihen um ihre Antwort zu bitten, bevor diese kommentiert werden.

Die Beispiele in Abbildung 6 stammen aus der Vorlesung „Algorithmen und Datenstrukturen“ im SoSe 2007 an der HTWK Leipzig.



Abbildung 6: Aktivierende Fragen aus der Vorlesung „Algorithmen und Datenstrukturen“.

### Kombination von Vorlesung und Übung

An den Universitäten Stuttgart und Leipzig wurde in verschiedenen Kontexten eine Kombination Vorlesung und Übung umgesetzt, wie sie in Abschnitt 3 zur engen Verzahnung von Input und Aktivität vorgeschlagen wird.

„Formale Methoden“ für Wirtschaftsinformatiker (Stuttgart) mit  $2V+1Ü$  ( $= 3(V+Ü)$ ) fand im WiSe 2004/05 mit 24 Studierenden statt. Ein wichtiges Lernziel dieser Veranstaltung ist, pragmatische Fertigkeiten im Kontext mit formalen Methoden der Informatik zu fördern. Durch die direkten Übungsphasen mit Gruppen- und Diskussionsaufgaben war es möglich, direkten Einfluss auf die Motivation der Studierenden zu nehmen und über ein unmittelbares Feedback den Lernerfolg zu steigern.

„Didaktik der Informatik“ für Lehramtsstudierende (Leipzig) mit  $2V+2Ü$  ( $= 4(V+Ü)$ ) fand im WiSe 2005/06 mit 14 und im WiSe 2006/07 mit 11 Studierenden statt. Neben der Vermittlung fachdidaktischer Inhalte und Methoden sind wichtige Lernziele dieser Veranstaltung affektiver Natur. Informatikunterricht an Schulen ist in vielerlei Hinsicht

besonders (falsches Bild von Informatik bei Kollegen, Schülern und Schülerinnen, stark heterogene Vorkenntnisse/-erfahrungen und Motivation, überdurchschnittlich viel Einzelbetreuung im Unterricht durch ausgedehnte Praxisphasen). Deshalb ist es wichtig, dass die angehenden Lehrer und Lehrerinnen sich dieser Besonderheiten bewusst werden und für sich entsprechende Strategien zum Umgang damit entwickeln. Die Integration von Gruppenarbeiten, Rollenspielen und entwickelndem Unterrichtsgespräch, die jeweils einzelne Besonderheiten im Zusammenhang mit Unterrichtsplanung und -durchführung thematisieren, erleichtern die Schulung derartiger affektiver Lernziele.

### Spiralkonzept in „ADS“

Der übliche Aufbau der Vorlesung „Algorithmen und Datenstrukturen“ gliedert sich in die drei bzw. vier große Teilabschnitte 0. Grundlagen, (falls notwendig), 1. Suchen, 2. Sortieren und 3. Graphalgorithmen. Diese Gliederung ergibt sich logisch aus einer inhaltlichen Sortierung. Der Zusammenhang zwischen gewählten Datenstrukturen, darauf möglichen Algorithmen und dem jeweils notwendigen Zeit- bzw. Platzbedarf wird für jeden Aufgabenbereich getrennt betrachtet. Auf Gemeinsamkeiten im dahinterliegenden Entwurfsprinzip wird bei einem derartigen Aufbau der Vorlesung in der Regel nur mündlich verwiesen. Der Linearansatz entspricht der üblichen Vorgehensweise, bei der der Vorlesungsaufbau hauptsächlich durch die Inhalte festgelegt wird. Wenn z.B. alle Verfahren zur Aufgabenstellung Suchen nacheinander behandelt werden, geht für die Studierenden oft das Verständnis verloren, warum jetzt noch ein andere Suchverfahren behandelt wird. Die Querbezüge zwischen den Algorithmen werden nur innerhalb des Abschnittes Suchen gezogen und die grundlegende Entwurfsmuster werden nur implizit vermittelt.

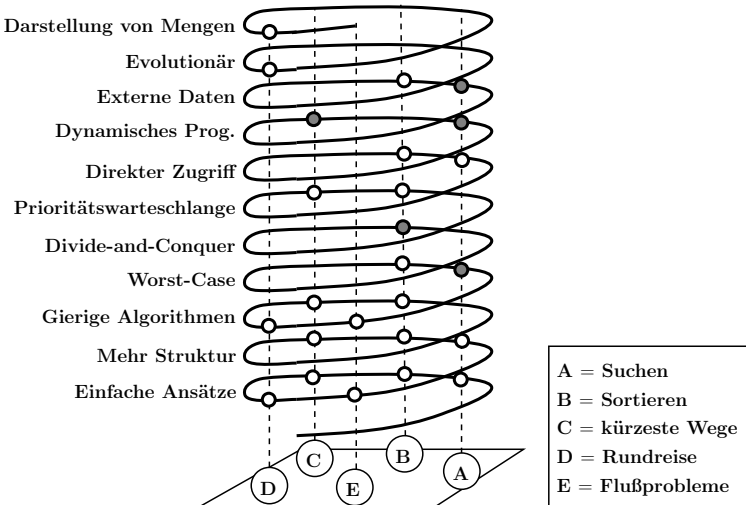


Abbildung 7: Spiralaufbau der Vorlesung „Algorithmen und Datenstrukturen“ an der HTWK Leipzig SoSe 2006 nach dem Spiralansatz. ○ bedeutet, dass das entsprechende Thema als Lösungsansatz für das jeweilige Problem A–E umgesetzt wird. ● kennzeichnet die Inhalte, die von den Studierenden erfahrungsgemäß als schwierig eingestuft werden.

Gerade für die Vorlesung „Algorithmen und Datenstrukturen“ bietet sich eine ideenorientierte Spiralstruktur für die Vorlesungsgliederung an, da anhand weniger grundsätzlicher Aufgabenstellungen (Suchen, Sortieren, etc.) verschiedene Algorithmenentwurfsmuster vorgestellt und analysiert werden (vgl. Abbildung 7). Dem Spiralentwurf geht die Überlegung voraus, welche wesentlichen Inhalte und Ideen die Studierenden erlernen sollen. Durch die Ideenorientierung wird ein besonderes Gewicht auf die Entwurfsmuster wie z.B. gierige Verfahren oder direkter Zugriff gelegt“. Die Studierenden sollen die Prinzipien und Denkweisen, die den Entwurfsmustern zugrundeliegen verstehen und anwenden lernen. Ein Nachschlagekatalog, welche Sortier- oder Suchverfahren es mit welchen Laufzeiten und welchen hauptsächlichen Einsatzgebieten gibt, wird durch die Zusammenfassung am Ende der Vorlesung geliefert.

## 5 Zusammenfassung und Ausblick

Die zielorientierte Didaktik der Informatik verwendet die didaktischen Prinzipien der Lernzielorientierung, der Anknüpfung an Vorwissen, die enge Verzahnung von Input und Aktivität, die Motivierung und frühzeitiges Feedback bzgl. aller Lernbereiche. Die in dieser Arbeit aufgeführten Beispiele der Umsetzung zielorientierter Informatiklehre zeigen, dass es möglich ist, auch unter der Einschränkung begrenzter Zeit- und Personalressourcen die Kompetenzvermittlung im Informatikstudium deutlich zu verbessern. Weitere Beispielen im Bereich der E-Learning-gestützte Programmierlehre, der Betreuung von Softwareprojekten, der Vermittlung von Projektleitungs Kompetenzen und der Betreuung von Abschlussarbeiten werden derzeit erarbeitet bzw. ausgewertet.

## Literaturverzeichnis

- [FB01] R. M. Felder und R. Brent. Effective strategies for cooperative learning. *J. Cooperation & Collaboration in College Teaching*, 10(2):69–75, 2001.
- [FB03] R. M. Felder und R. Brent. Learning by doing. *Chem. Engr. Education*, 37(4):282–283, 2003.
- [GI05] Gesellschaft für Informatik e. V., Hrsg. Empfehlungen für Bachelor- und Masterprogramme im Studiengang Informatik an Hochschulen. Gesellschaft für Informatik e. V., Bonn, 2005.  
<http://www.gi-ev.de/service/publikationen/empfehlungen/> (Stand 26.5.2007).
- [KW01] Andreas Krapp und Bernd Weidenmann. *Pädagogische Psychologie*. Beltz, Weinheim, 4. Auflage, 2001.
- [Ve04] Frederic Vester. *Denken, Lernen, Vergessen*. dtv, München, 30. Auflage, 2004.
- [WDW06] Nicole Weicker, Botond Draskoczy und Karsten Weicker. Fachintegrierte Vermittlung von Schlüsselkompetenzen der Informatik. In Peter Forbrig, Günter Siegel und Markus Schneider, Hrsg., *HDI 2006: Hochschuldidaktik der Informatik – Organisation, Curricula, Erfahrungen*, Seiten 51–62, Bonn, 2006. GI-Edition Lecture Notes in Informatics. 2. GI-Fachtagung, 7./8.12.2006 in München.