

Bartelsen, Jan; Brauer, Johannes

**Working Paper**

## Kooperatives Lernen mit moodle in der Programmierausbildung

Arbeitspapiere der Nordakademie, No. 2008-06

**Provided in Cooperation with:**

Nordakademie - Hochschule der Wirtschaft, Elmshorn

*Suggested Citation:* Bartelsen, Jan; Brauer, Johannes (2008) : Kooperatives Lernen mit moodle in der Programmierausbildung, Arbeitspapiere der Nordakademie, No. 2008-06, Nordakademie - Hochschule der Wirtschaft, Elmshorn

This Version is available at:

<https://hdl.handle.net/10419/38596>

**Standard-Nutzungsbedingungen:**

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

**Terms of use:**

*Documents in EconStor may be saved and copied for your personal and scholarly purposes.*

*You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.*

*If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.*



**NORDAKADEMIE**  
HOCHSCHULE DER WIRTSCHAFT

**ARBEITSPAPIERE DER NORDAKADEMIE**  
**ISSN 1860-0360**

**Nr. 2008-06**

# **Kooperatives Lernen mit moodle in der Programmierausbildung**

**Jan Bartelsen**  
**Johannes Brauer**

**Oktober 2008**

Dieses Arbeitspapier ist als PDF verfügbar: <http://www.nordakademie.de/arbeitspapier.html>



**NORDAKADEMIE**  
HOCHSCHULE DER WIRTSCHAFT



Köllner Chaussee 11  
25337 Elmshorn  
<http://www.nordakademie.de>

# Inhaltsverzeichnis

<b>Abstract</b>	<b>1</b>
<b>1 Einleitung und Rahmenbedingungen</b>	<b>1</b>
<b>2 Kooperatives Lernen</b>	<b>2</b>
2.1 Der Begriff „CSCL“	2
2.2 Vorteile und Nachteile	3
2.3 Kommunikations-Werkzeuge	4
2.3.1 Grundlagen	4
2.3.2 Foren und newsgroups	4
2.3.3 Wikis	5
2.3.4 Weitere Werkzeuge	5
<b>3 Didaktische Mittel</b>	<b>6</b>
3.1 Tutorials und Screencasts	6
3.2 Projektmethode	6
3.3 Problemorientiertes Lernen	7
<b>4 Entwickelte Lösung</b>	<b>8</b>
4.1 Einleitung	8
4.2 elementare Lernaktivitäten	8
4.2.1 Wissen durch Screencast erwerben	8
4.2.2 Erläutern von Fachbegriffen	10
4.2.3 Report erstellen	13
4.2.4 Eigenen Lernerfolg feststellen	15
4.2.5 Gruppenprogrammierung	17
4.2.6 Beteiligte Akteure	18
<b>5 Ausblick</b>	<b>18</b>
<b>Literaturverzeichnis</b>	<b>19</b>

## Abstract

Innerhalb der NORDAKADEMIE-Vorlesung „Programmierung 1“ erhalten die Teilnehmer eine Grundausbildung in der Programmierung mithilfe der Programmiersprache Smalltalk. Diese Vorlesung soll nun um kooperative Elemente, die online über die Lernplattform *moodle* vermittelt werden, erweitert werden. Das vorliegende Arbeitspapier enthält zunächst eine Übersicht über Formen des kooperativen Lernens sowie eine Beschreibung gängiger didaktischer Mittel.

Daran anschließend folgt eine Beschreibung von fünf elementaren Lernaktivitäten, die dazu geeignet sind, die Vorlesung zu ergänzen und den Einsatz einer Lernplattform wie *moodle* systematisch zu gestalten.

## 1 Einleitung und Rahmenbedingungen

Vor einigen Jahren wurde an der NORDAKADEMIE die Programmiergrundausbildung im Studiengang Wirtschaftsinformatik unter das Motto „Objektorientierung von Anfang an“ gestellt. Dies führte zu einem zweistufigen Konzept: In den beiden ersten Semestern wird eine Einführung in die objektorientierte Programmierung (unter Verwendung der Programmiersprache Smalltalk) unterrichtet. In den beiden darauf folgenden Semestern geht es um die Übertragung erlernter Grundbegriffe der Objektorientierung auf eine zweite Programmiersprache (zurzeit Java). Mit diesem Ansatz sollte der zunehmenden Bedeutung der Objektorientierung auch in kaufmännisch-administrativen Anwendungsbereichen – zumindest bei Neuentwicklungen – Rechnung getragen werden. Der zuvor benutzte Einstieg über die prozedurale Programmierung, dem dann in höheren Semestern der „Umstieg“ auf die objektorientierte Programmierung folgte, hatte sich als schwierig und wenig erfolgreich herausgestellt.

Es soll hier nicht thematisiert werden, ob der Ansatz „Objektorientierung von Anfang an“ der richtige ist, welche Programmiersprachen gewählt und welche Entwicklungsumgebungen benutzt werden sollten. Das soll aber keineswegs bedeuten, dass diese Fragen etwa nicht mehr relevant wären. Man kann sich leicht vom Gegenteil überzeugen, indem man die Programme einschlägiger Konferenzen, wie z. B.

- *SIGCSE* Technical Symposium on Computer Science Education
- *ITiCSE* Innovation and Technology in Computer Science Education

betrachtet.

An dieser Stelle soll über Konsequenzen berichtet werden, die aus der Beobachtung gezogen wurden, dass die Studierenden zu einem großen Teil nach den beiden ersten Semestern das Paradigma der Objektorientierung nur unzureichend verinnerlicht haben. Es konnte beispielsweise beobachtet werden, dass die prozedurale Denkweise immer wieder „durchbricht“, Schwierigkeiten beim Klassenentwurf bestehen, die Konzepte Vererbung und Delegation nicht adäquat benutzt und Fallunterscheidungen anstelle der Nutzung von Polymorphie eingesetzt werden.

Ausgehend von diesen Erkenntnissen wurde im Jahre 2006 damit begonnen, die programmierpraktischen Übungen in den beiden ersten Semestern von Einzelaufgaben auf Semesteraufgaben umzustellen, da die Konzepte der Objektorientierung anhand kleiner Übungsaufgaben nur schwer motiviert und vermittelt werden können. Um die Studierenden in der Selbständigkeit des Wissenserwerbs zu stärken und zu motivieren, wurde gleichzeitig die „klassische“ Vorlesung durch einen web-gestützten Kurs auf der Grundlage der Lernplattform *moodle* ersetzt. Diese Umstellung wurde auch durch die Tatsache „beflügelt“, dass der gleichzeitige Wechsel

von Diplom- zu Bachelor-Studiengängen eine Reduzierung der zur Verfügung stehenden Unterrichtsstunden mit sich brachte und eine Überarbeitung des Konzepts der Lehrveranstaltung ohnehin notwendig machte.

Insbesondere die Umstellung auf *moodle* war eine „Ad-hoc-Aktivität“, die anfangs im Wesentlichen darin bestand, dort die Programmieraufgabe in Teilschritten zu beschreiben und Verknüpfungen auf Teile des existierenden Vorlesungsskripts (PDF-Dokumente) zu hinterlegen.

Den Einsatz der Lernplattform zu systematisieren, war die Hauptaufgabe, die im Rahmen einer Master-Thesis mit dem Titel „Entwurf einer kooperativen Lehrveranstaltung in *moodle* für die Grundausbildung in der Programmierung“ (angefertigt im Sommer 2008 im Rahmen des Studiengangs „Master of Computer Science“ an der FernUniversität Hagen) von Jan Bartelsen gelöst wurde<sup>1</sup>. Über einen Teil der Ergebnisse dieser Arbeit soll hier zusammenfassend berichtet werden.

Die Systematisierung des Einsatzes von *moodle* sollte sich unter Berücksichtigung der didaktischen Grundlagen kooperativer Lernformen und des Blended Learning insbesondere auf die adäquate Nutzung der von der Lernplattform gebotenen Lernaktivitäten und die Vereinheitlichung der Darstellung von Lerntexten ohne Medienbrüche beziehen. Darüber hinaus bestand das Ziel darin, den Kurs so zu konzipieren, dass er einerseits von Studierenden – z. B. in vorlesungsfreien Zeiten – selbständig durchgearbeitet werden kann, andererseits während der Präsenzveranstaltung dem Dozenten genügend Freiraum zur individuellen Betreuung von Studierenden verschafft.

Die Entwicklung eines Lernmoduls für den Kurs „Programmierung 1“ unter *moodle* ist Ziel des Projektes NORDPOL (NORDakademie Programmierunterricht OnLine), welches am 1. Januar 2008 an der NORDAKADEMIE für eine Dauer von zwei Jahren begonnen wurde. Innerhalb von NORDPOL sollen neben technischen auch fachdidaktische Aspekte beleuchtet werden. Ferner ist zu untersuchen, wie sich der Lernerfolg bei der objekt-orientierten Programmierausbildung messen lässt und ob der Lernerfolg durch den Einsatz einer Lernplattform und CSCL-Techniken<sup>2</sup> verbessert werden kann. Ziel ist es, dass der Dozent vorwiegend als Coach fungiert, der dadurch in die Lage versetzt wird, die Studierenden bei Problemen individuell zu beraten.

Zum Abschluss dieses Kapitels sei darauf hingewiesen, dass im vorliegenden Arbeitspapier bei der Beschreibung von Rollen und Personen die männliche Form verwendet wird, also beispielsweise „der Student“ oder „der Dozent“. Damit sind stets Männer und Frauen gemeint, also auch die Studentin und die Dozentin. Aus Gründen der einfachen Lesbarkeit wurde auf die Nennung von männlichen und weiblichen Formen oder auf Mischformen (wie „Student/in“) verzichtet. Alle Frauen werden hierfür um Verständnis gebeten.

## 2 Kooperatives Lernen

### 2.1 Der Begriff „CSCL“

Kooperatives Lernen ist Gegenstand des Forschungsgebietes CSCL, wobei die Bedeutung der vier Buchstaben nicht eindeutig festgelegt ist. Je nach Autor wird „CSCL“ als „Computer Supported Cooperative Learning“<sup>3</sup> oder als „Computer Supported Collaborative Learning“<sup>4</sup> definiert<sup>5</sup>.

---

<sup>1</sup> siehe [Bar08]

<sup>2</sup> siehe Kapitel 2.1

<sup>3</sup> Computergestütztes kooperatives Lernen

<sup>4</sup> Computergestütztes kollaboratives Lernen

<sup>5</sup> Vgl. [Bar08], S. 9 und [HSW04], S. 1

Im vorliegenden Arbeitspapier soll zwischen den unterschiedlichen Übersetzungen des zweiten C im Begriff CSCL nicht weiter differenziert werden und frei nach der Definition von Haake et al. CSCL wie folgt verstanden werden:

CSCL bezeichnet den Einsatz von Informatiksystemen zur Unterstützung des gemeinsamen Lernens in der Gruppe, bei dem die Gruppenmitglieder gemeinsam Wissen erarbeiten und erwerben.<sup>6</sup>

Zum „klassischen E-Learning“ kommt bei CSCL der Gruppenaspekt hinzu, bei dem die Kursteilnehmer in einer Gruppe Aufgaben bearbeiten und miteinander kommunizieren. Hervorzuheben ist, dass es sich bei CSCL um ein interdisziplinäres Forschungsgebiet handelt, an dem neben der Informatik auch die Pädagogik, die Psychologie (Wie lernen Menschen?) sowie die Soziologie und Kommunikationswissenschaft (Wie kann Kommunikation und Kooperation in verteilten Lerngruppen gefördert werden?) beteiligt sind.<sup>7</sup>

## 2.2 Vorteile und Nachteile

Kooperatives Lernen wird als förderlich für den Lernerfolg angesehen und ist dem individuellen Lernen oft überlegen<sup>8</sup>, da sich die Lernenden beim kooperativen Lernen durch die Auseinandersetzung mit anderen Lernenden sowie die aktive Erarbeitung eines Wissensgebietes intensiver mit dem Lernstoff beschäftigen<sup>9</sup>. Details hierzu hat Slavin in seinem Buch „Cooperative Learning“<sup>10</sup> veröffentlicht.

Speziell bezogen auf CSCL werden von Hinze weitere Vorteile genannt<sup>11</sup>, von denen der Kompetenzerwerb und der individualisierte Lernweg näher erläutert werden sollen:

- Kompetenzerwerb: In dem Moment, wo sich verschiedene Lerner austauschen müssen, um gemeinsam zu einer Lösung zu kommen, können auch Konflikte auftreten, wenn unterschiedliche Meinungen vertreten werden. Da zu diesen Konflikten gemeinsam eine Lösung gefunden werden muss, schult ein CSCL-Modul „ganz nebenbei“ noch die Team-Fähigkeit der einzelnen Lernenden. Hinze betont sogar, dass die Förderung sozialer Kompetenz ein zentraler Lernaspekt und das Ziel „nicht allein der Erwerb von Faktenwissen“ sei.<sup>12</sup> Hinzu kommt die technische Kompetenz. Wenn die Studenten die Nutzung von Online-Kommunikationstechniken bereits im Studium kennenlernen und anwenden können, haben sie beim Berufseinstieg einen Vorteil gegenüber Bewerbern, die diese Kompetenzen nicht besitzen.

- Individualisierter Lernweg: In einer großen Gruppe von Lernenden kommt es immer wieder vor, dass einige Teilnehmer den Lernstoff schneller verstehen als andere. Hinzu kommt, dass bei einigen Teilnehmern bereits Vorwissen in unterschiedlichem Umfang existiert. Für den Dozenten besteht häufig die Schwierigkeit, die Anfänger nicht zu überfordern und gleichzeitig die Teilnehmer mit Vorkenntnissen nicht zu langweilen. Mit Hilfe von CSCL können hier verschiedene Aufgaben, bzw. Lektionen erstellt werden, so dass Teilnehmer mit Vorkenntnissen bereits

---

<sup>6</sup>Vgl. [HSW04], S. 2

<sup>7</sup>Vgl. [HSW04], S. 2

<sup>8</sup>Vgl. [WH03], S. 71

<sup>9</sup>Vgl. [Hin04], S. 37 und [WH03], S. 71

<sup>10</sup>[Sla95]

<sup>11</sup>Vgl. [Hin04], S. 37

<sup>12</sup>[Hin04], S. 37

in der ersten Vorlesung Aufgaben von späteren Lektionen oder knifflige Zusatzaufgaben bearbeiten, während der Dozent genügend Zeit hat, den Anfängern bei Problemen zu helfen.

Der Einsatz von kooperativen Lernmethoden und CSCL kann jedoch auch Nachteile erzeugen. Wie allgemein bei kooperativer Arbeit (und damit auch beim kooperativem Lernen) wird von allen Beteiligten eine erhöhte Teamfähigkeit gefordert, zusammen mit der Bereitschaft, sich aktiv einzubringen<sup>13</sup>. Die verminderte Anstrengung und Leistung einzelner Gruppenmitglieder kann im Extremfall dazu führen, dass ein Gruppenmitglied gar keine Leistung mehr erbringt, da die Überzeugung besteht, dass der eigene Beitrag ohnehin überflüssig sei und die Leistung des Einzelnen gar nicht beachtet werde. Dieses, auch als „soziales Faulenzen“ oder „Trittbrettfahren“<sup>14</sup> bezeichnete Vorgehen kann laut Janneck und Janneck jedoch vermieden werden, wenn die Anonymität bei der Kommunikation aufgehoben wird, so dass der Beitrag des Einzelnen jederzeit erkennbar ist<sup>15</sup>.

Ein weiterer Nachteil des kooperativen Arbeitens und Lernens ist die Gefahr von Prozessverlusten, die auftreten können, wenn die Leistung der Gruppe schlechter ist als die Summe der Einzelleistungen, was auch als „Ringelmann-Effekt“ bezeichnet wird<sup>16</sup>. Für weitere Vor- und Nachteile des CSCL sei auf Hinze<sup>17</sup> verwiesen.

## 2.3 Kommunikations-Werkzeuge

### 2.3.1 Grundlagen

Kommunikation bildet „die Grundlage für gemeinsame Lernerfahrungen“<sup>18</sup>. Und während beim Präsenz-Lernen die Kommunikation face-to-face erfolgt, müssen beim CSCL alternative Techniken zur Kommunikation gefunden werden.

Diese Techniken können nach verschiedenen Arten differenziert werden. Schümmer und Haake unterscheiden bspw. zwischen asynchroner (zeitversetzter) und synchroner (zeitgleicher) Kommunikation sowie nach der Anzahl der Kommunikationspartner und stellen Unterschiede zwischen bilateraler (zwischen zwei Partnern) und multilateraler (zwischen mehr als zwei Partnern) Kommunikation fest<sup>19</sup>.

Im Folgenden sollen einige Kommunikationswerkzeuge und ihr Nutzen für CSCL kurz vorgestellt werden.

### 2.3.2 Foren und newsgroups

Foren und newsgroups werden dazu genutzt, multilateral und asynchron in Textform miteinander zu kommunizieren. Wenn ein Teilnehmer einen Beitrag verfasst, wird dieser gespeichert und kann zu einem späteren Zeitpunkt von anderen gelesen werden. Obgleich es in der Technik einen Unterscheid zwischen einer newsgroup und einem Forum gibt, sollen diese Begriffe im Rahmen des vorliegenden Arbeitspapiers synonym verwendet werden, da sie im Bereich des kooperativen Lernens denselben Nutzen erzeugen.

---

<sup>13</sup>Vgl. [WH03], S. 71

<sup>14</sup>Vgl. [JJ04], S. 46 und [Hin04], S. 40

<sup>15</sup>[JJ04], S. 47

<sup>16</sup>Vgl. [Hin04], S. 39f und [Tüc05], S. 487ff

<sup>17</sup>Vgl. [Hin04]

<sup>18</sup>[SH04], S. 66

<sup>19</sup>Vgl. [SH04], S. 67

Foren werden im Rahmen von kooperativen Lehrveranstaltungen dazu verwendet, Fragen der Studierenden zu beantworten oder inhaltliche Diskussionen zum Thema des Kurses zu führen.

Ein Problem von Foren, die als Begleitung von Hochschul-Präsenz-Veranstaltungen angeboten werden, ist jedoch, dass sich kaum Studenten an den Diskussionen beteiligen. Diese Erfahrung konnte in Veranstaltungen der NORDAKADEMIE gemacht werden. So gab es beispielsweise im Forum zum Kurs „Praxis der Software-Entwicklung“ im Wintersemester 2007/2008 (ca. 100 Teilnehmer) innerhalb des Vorlesungszeitraums keinen einzigen ernstgemeinten Beitrag. Erst zum Ende des Semesters, als keine Vorlesung mehr angeboten wurde und die Studenten eine Hausarbeit als Prüfungsleistung anzufertigen hatten, wurden einige wenige Fragen gestellt.

Die Nutzung des Forums im Rahmen des Kurses ‘Praxis der Software-Entwicklung‘ änderte sich jedoch sehr stark, als im Sommersemester 2008 eine zweite Gruppe ihre Hausarbeit angefertigt hat. Diese zweite Gruppe unterschied sich dadurch, dass der Termin außerhalb der Semesterzeit lag, die Studenten sich also nicht regelmäßig gesehen haben. Hierbei wurde festgestellt, dass das Forum plötzlich sehr stark genutzt wurde. Viele Fragen und Diskussionen, die bei der ersten Gruppe (die die Hausarbeit während der Semesterzeit angefertigt hat) offenbar im persönlichen Gespräch geführt wurden, wurden nun ins Forum verlagert. Eine Umfrage unter 171 Studierenden der NORDAKADEMIE (wobei 51 den Fragebogen ausgefüllt haben) hat ergeben, dass die Studierenden stets das persönliche Gespräch der schriftlichen Kommunikation über ein Forum vorziehen<sup>20</sup>. Von daher ist ein Forum vor allem für Fern-Kurse und weniger als Begleitung von Präsenz-Veranstaltungen zu empfehlen.

Im Forum des NORDAKADEMIE-Kurses „Betriebssysteme“ im Sommersemester 2007 (ca. 60 Teilnehmer) sowie an anderen Hochschulen wurden ähnliche Erfahrungen gemacht, wobei festgestellt wurde, dass eine hohe Foren-Beteiligung vereinzelt auch durch eine Art Zwang zur Forum-Teilnahme „erkauft“ wurde. Details hierzu sind u.a. bei Bartelsen<sup>21</sup> zu finden.

### 2.3.3 Wikis

Ein Wiki bezeichnet eine Sammlung von Webseiten, die, ähnlich einem Content Management System, von den Benutzern geändert werden können. Dadurch werden Wikis auch für kooperatives Lernen interessant, da sie die gemeinsame Bearbeitung und Erstellung von Lerninhalten erlauben<sup>22</sup>. Innerhalb der Lernplattform *moodle* steht die Lernaktivität „Wiki“ standardmäßig zur Verfügung. Eine mögliche Verwendung innerhalb einer kooperativen Lehrveranstaltung besteht darin, die Studenten aufzufordern, neue Fachbegriffe im Wiki zu erklären und gleichzeitig die Ergebnisse der Kommilitonen korrekturlesen. Auf diese Weise entsteht ein großes Nachschlagewerk an Wissen, welches von den Teilnehmern selbst zusammengetragen wurde.

### 2.3.4 Weitere Werkzeuge

Die oben beschriebenen Werkzeuge werden derzeit an der NORDAKADEMIE eingesetzt. Darüber hinaus gibt es noch weitere Werkzeuge, die vor allem für Fernkurse geeignet sind. Hierzu zählen Chat, Instant-Messenger-Systeme, (Video-)Telefonie sowie das Lernen in virtuellen Welten wie beispielsweise „Second Life“. Weitere Informationen zu diesen Werkzeugen sind u.a. bei Schümmer und Haake<sup>23</sup> sowie bei Bartelsen<sup>24</sup> zu finden. Hinweise zum Lernen in virtuellen

---

<sup>20</sup>Vgl. [Bar08], S.34f

<sup>21</sup>[Bar08], S.15

<sup>22</sup>Vgl. [Ger07], S. 385

<sup>23</sup>[SH04]

<sup>24</sup>Vgl. [Bar08], S. 16f

Welten sind auf der Webseite des Projektes „E-Learning 3D“ der Universität Bielefeld zu finden, welches den Einsatz virtueller Welten in Lehr-Lern-Situationen erforscht<sup>25</sup>.

## 3 Didaktische Mittel

### 3.1 Tutorials und Screencasts

Der Bereich der Tutorials (auch „E-Tutorial“ oder „direkte Instruktion“ genannt) ist vor allem in „klassischen“ E-Learning-Anwendungen wie CBT<sup>26</sup> oder WBT<sup>27</sup> zu finden. Aber auch in Lernplattformen wie *moodle* können Tutorials eingesetzt werden. Bei Tutorials handelt es sich um Texte, die, angereichert durch Bilder oder Videos, vom Lernenden durchgelesen und bearbeitet werden. Ein Wissenserwerb erfolgt durch das Lesen und anschließendes Umsetzen des Inhaltes dieser Texte.

Neuere Formen von Tutorials sind Videoaufzeichnungen, beispielsweise von einer Vorlesung. Hierbei unterscheidet man zwischen „Screencasts“ (auch „Video-Tutorial“ genannt), die lediglich den Bildschirm des Dozenten aufzeichnen (einschließlich einer gesprochenen Erklärung), sowie „Video-Podcasts“, die die gesamte Vorlesung per Video zeigen. Es existieren auch Mischformen, bei denen hauptsächlich ein Screencast gezeigt wird, aber der Dozent auf einem kleinen Bereich des Videos zu sehen ist.

Durch die Bearbeitung des Tutorials oder Screencasts erfolgt eine Aneignung von Wissen, die am Ende des Tutorials durch einen Test abgefragt werden kann. Der Schwerpunkt von Tutorials liegt allerdings in der Bearbeitung durch eine einzelne Person, eine Gruppenarbeit findet hier in der Regel nicht statt.

Gegenüber einer Präsenzveranstaltung bietet ein Screencast einige Vorteile: So kann der Lernende die Lerngeschwindigkeit an sein eigenes Lerntempo anpassen. Weiterhin besteht die Möglichkeit, einen Screencast beliebig oft zu wiederholen, wenn man sich ein bestimmtes Thema zu einem späteren Zeitpunkt, z.B. vor einer Klausur, erneut anhören möchte. Demgegenüber ist als Nachteil zu nennen, dass keine direkte Interaktion mehr mit dem Dozenten stattfinden kann, z.B. durch Unterbrechen während der Vorlesung. Die Interaktion beschränkt sich auf die Kommunikation während Sprechstunden oder über webbasierte Foren.

### 3.2 Projektmethode

Innerhalb einer Lehrveranstaltung ist ein Projekt eine Unterrichtseinheit, „die durch Selbstorganisation der Lerngruppe gekennzeichnet ist und bei der der Arbeits- und Lernprozess ebenso wichtig ist wie das Endergebnis oder Produkt, das am Ende des Projektes steht.“<sup>28</sup> Im schulischen Unterricht hat ferner der Aspekt des fächerübergreifenden Unterrichtes eine große Bedeutung<sup>29</sup>. Die Abgrenzung der Projektmethode zu anderen didaktischen Methoden (z.B. dem problemorientierten Lernen) ist jedoch oft unscharf, so dass das Projekt häufig den Oberbegriff für verwandte Konzepte bildet<sup>30</sup>.

Schubert und Schwill weisen darauf hin, dass insbesondere im Informatik-Unterricht darauf zu achten ist, den pädagogischen Projektbegriff und den informatischen Projektbegriff nicht

<sup>25</sup> siehe <http://www.e-learning3d.de/>

<sup>26</sup> Computer Based Training

<sup>27</sup> Web Based Training

<sup>28</sup> [SS04], S. 297

<sup>29</sup> Vgl. [SS04], S. 297

<sup>30</sup> Vgl. [Jan04a], S. 244

miteinander zu vermischen. Software-Entwicklung findet heute üblicherweise in Projekten statt. Die Software-Engineering-Methoden in Informatik-Projekten haben dabei als ein Ziel, die zu erledigende Aufgabe in kleine Arbeitspakete zu zerteilen, die von den Projektmitgliedern weitestgehend in Einzelarbeit erledigt werden. Pädagogische Projekte haben jedoch genau das entgegengesetzte Ziel: Die Projektmitglieder sollen für die gesamte Projektlaufzeit konsequent kooperieren und Generalisten mit einem Überblick über das Gesamtprojekt bleiben<sup>31</sup>.

Die pädagogische Projektmethode geht auf John Dewey zurück. Dessen Idee des „learning by doing“ besagt, dass nicht das Erreichen vorher definierter Ziele zählt, sondern der während des Lernens erzielte Erfahrungsgewinn<sup>32</sup>. Ein pädagogisches Projekt im Rahmen einer Lehrveranstaltung besitzt mindestens folgende Merkmale und Eigenschaften<sup>33</sup>:

- Das Projekt muss einen praktisch-nützlichen Wert besitzen und auf die Interessen der Lernenden abgestimmt sein.
- Planung und Ausführung sind durch die Selbstständigkeit der Lernenden bestimmt.

Janneck nennt noch weitere Eigenschaften<sup>34</sup>, jedoch sind die oben genannten Eigenschaften diejenigen, die in der Literatur als Minimum von pädagogischen Projekten genannt werden. Weitere Eigenschaften werden nur von einzelnen Autoren genannt.

### 3.3 Problemorientiertes Lernen

Ein Problem entsteht laut Klix dann, wenn es nicht oder nicht unmittelbar gelingt, einen gegebenen Anfangszustand in einen gegebenen Zielzustand zu überführen<sup>35</sup>. Beim problemorientierten Lernen erfolgt das Lernen nun anhand solcher Problemstellungen mit einem möglichst hohen Praxisbezug. Die Studenten werden dabei ermutigt, als Gruppe eine Lösung für das Problem zu finden<sup>36</sup> und ihre Erfahrungen zu reflektieren<sup>37</sup>. Das problemorientierte Lernen baut auf dem Konstruktivismus auf, dem zufolge Lernen als aktiver Konstruktionsprozess angesehen wird<sup>38</sup>. Hoffmann definiert problemorientiertes Lernen als eine kooperativ selbstorganisierte, schrittweise aufgebaute Lösung von ausgewählten authentischen Problemen, die von Seiten der Lehrenden moderierend und begleitend unterstützt wird<sup>39</sup>. Sie betont dabei die Lernerzentrierung, da die Suche nach Information und ihrer Strukturierung in die Hände der Lernenden gelegt wird<sup>40</sup>.

Der Ablauf des problemorientierten Lernens kann nach Hoffmann wie folgt beschrieben werden<sup>41</sup>:

- Zunächst wird das zu lösende Problem von der Lehrperson vorgestellt.
- Anschließend werden Verständnisschwierigkeiten ausgeräumt und erste Problemlösestrategien entwickelt. Die Rolle des Lehrenden wandelt sich zu einem Moderator und Koordinator.

---

<sup>31</sup> Vgl. [SS04], S. 301

<sup>32</sup> Vgl. [Sch01], S. 286 und [GdW04], S. 35 nach John Dewey

<sup>33</sup> Vgl. [Sch01], S. 286 und [Jan04a], S. 239f

<sup>34</sup> Vgl. [Jan04a], S. 239f

<sup>35</sup> Vgl. [Kli76], S. 640

<sup>36</sup> Vgl. [SLH05], S. 1

<sup>37</sup> Vgl. [CNHS05], S. 1

<sup>38</sup> Vgl. [RS02], S. 217

<sup>39</sup> Vgl. [Hof04], S. 245

<sup>40</sup> Vgl. [Hof04], S. 251

<sup>41</sup> Vgl. [Hof04], S. 247ff

- Im Folgenden klären die Lernenden, welche Wissensbestände sie haben und wo noch Wissensdefizite bestehen. Außerdem wird eine Problemlösestrategie entwickelt.
- Im nächsten Schritt wird die entwickelte Strategie verfolgt. Ggf. sind an dieser Stelle Änderungen in der Zielstellung oder der Strategie vorzunehmen, falls Probleme auftreten. Die Rolle des Lehrenden kann um die eines Lernbegleiters und Tutors, der bei Problemen beratend zur Seite steht, erweitert werden.
- Im letzten Schritt sind die Ergebnisse und die Verfahrensweise aufzuarbeiten und zu reflektieren.

Diese fünf genannten Phasen können sich sowohl auf reine Präsenz-Veranstaltungen als auch auf virtuelle Veranstaltungen im Rahmen von CSCL-Umgebungen beziehen. Wenn die Ergebnisse und die Reflexion in digitaler Form vorliegen, bietet sich der Vorteil, dass die Dokumente archiviert und leicht allen Teilnehmern des Kurses zur Verfügung gestellt werden können. Für die Reflexionsphase ist laut Hoffmann auch bei virtuellen Veranstaltungen eine Präsenzphase zu empfehlen<sup>42</sup>.

## 4 Entwickelte Lösung

### 4.1 Einleitung

Zur Durchführung einer kooperativen Lehrveranstaltung in *moodle* für die Programmierausbildung wurden fünf elementare Lernaktivitäten entwickelt, die im Folgenden vorgestellt werden. Diese Lernaktivitäten können unabhängig von einem konkreten Thema oder einem konkreten Kurs (teilweise sogar auch unabhängig vom Fachbereich „Informatik“) eingesetzt werden. Ab Januar 2009 werden diese Lernaktivitäten im Verlauf der Vorlesung „Programmierung 1“ mehrfach eingesetzt und verwendet. Ergebnisse über den Erfolg dieser Lernaktivitäten können somit noch nicht mitgeteilt werden. Es ist jedoch geplant, die Ergebnisse nach Durchführung des Kurses „Programmierung 1“ ebenfalls zu veröffentlichen.

In den folgenden Beschreibungen der elementaren Lernaktivitäten treten Rollen Dozent, Tutor und Student auf, die an dieser Stelle kurz erläutert werden sollen:

Der Dozent führt den Kurs durch. Er besitzt das Expertenwissen, welches die Teilnehmer erlernen sollen und hält die zum Kurs gehörenden Präsenzvorlesungen. Er beantwortet fachliche Rückfragen per E-Mail oder im Webforum und korrigiert eingesendete Übungsaufgaben. Außerdem stellt und korrigiert er am Ende des Kurses die Klausur.

Der Tutor unterstützt den Dozenten bei Präsenzveranstaltungen. Er hilft den Studenten bei Programmierübungen, wenn diese nicht weiter wissen oder Fehler auftreten. Auch der Tutor kann fachliche Rückfragen im Webforum beantworten. In einem Kurs können mehrere Tutoren tätig sein, wobei die Tutoren in der Regel Studenten aus einem höheren Semester sind.

Der Student ist Teilnehmer des Kurses und möchte die Programmiersprache Smalltalk erlernen.

### 4.2 elementare Lernaktivitäten

#### 4.2.1 Wissen durch Screencast erwerben

Zum Erlernen einer Programmiersprache gehört oft auch das Erlernen einer entsprechenden Entwicklungsumgebung. Bei Smalltalk wird stets eine integrierte Entwicklungsumgebung mit

<sup>42</sup>Vgl. [Hof04], S. 250

einer graphischen Oberfläche verwendet<sup>43</sup>, wobei im Rahmen des Kurses „Programmierung 1“ die Entwicklungsumgebung VisualWorks der Firma Cincom eingesetzt wird. Die Bedienung einer solchen Entwicklungsumgebung nur in Worten zu beschreiben ist oft schwierig. Gemäß dem Sprichwort „Ein Bild sagt mehr als tausend Worte“ ist es oft einfacher und verständlicher, die entsprechende Bedienung vorzuführen und am Bildschirm zu zeigen. Solche Vorführungen verlangen jedoch eine Präsenz-Veranstaltung, was sich in internetgestützten, virtuellen Lehrveranstaltungen als schwierig erweist. Abhilfe schaffen hier die in Kapitel 3.1 vorgestellten Screencasts.

Sobald ein Dozent die Entscheidung getroffen hat, dass den Studenten Video-Aufzeichnungen angeboten werden sollen, stellt sich die Frage, was für eine Art von Video-Aufzeichnung verwendet werden soll. Soll nur der Bildschirm des Dozenten aufgezeichnet werden (Screencast)? Oder auch der Dozent selbst, weil seine Gesten für das Verständnis des Vortrages wichtig sind (Video-Podcast)?

Für den Programmier-Unterricht sind Video-Aufzeichnungen, die auch den Dozenten zeigen, nicht erforderlich, da es hier vor allem darauf ankommt, eine detaillierte Sicht auf den Computer des Dozenten zu haben, um die von ihm vorgeführten Schritte und Handlungen in der Entwicklungsumgebung nachvollziehen zu können. Hier bietet sich ein Screencast an, bei dem der Dozent Programmier-Tätigkeiten auf seinem Computer durchführt und dessen Bildschirm, zusammen mit gesprochenen Erklärungen, aufgezeichnet wird. Dabei sollte darauf geachtet werden, die einzelnen Screencasts nicht zu groß werden zu lassen. Bartelsen erwähnt, dass sich für einen Screencast eine Dauer von maximal fünf Minuten als ideal erwiesen hat und nennt entsprechende Nachteile größerer Screencasts<sup>44</sup>.

## Lernziel

Die Studenten sollen den Umgang mit der Entwicklungsumgebung VisualWorks lernen. Jeder Screencast behandelt dabei ein eigenes Thema, z.B. „Einstieg in VisualWorks“ oder „Nutzung des Object Explorers“. Das Verständnis des jeweiligen Themas ist das Lernziel eines Screencasts.

## Ablauf

Der Ablauf dieser Lernaktivität ist wie folgt aufgebaut (vgl. auch Abbildung 1):

- Im Rahmen der Vorbereitung zum Kurs fertigt der Dozent einen Screencast an und speichert ihn in einem Format, welches für die Wiedergabe in einem Internet-Browser geeignet ist.
- Der Dozent stellt den Screencast in der Lernplattform zur Verfügung.
- Die Studenten schauen sich den Screencast an.
- Die Studenten versuchen, die im Screencast gezeigten Handlungen am eigenen PC nachzuvollziehen und zu wiederholen. Wie in der Lerntheorie des Konstruktivismus<sup>45</sup> beschrieben, stellt sich vor allem durch solche aktiven Handlungen ein Lerneffekt ein.

---

<sup>43</sup> Vgl. [Bra04], S.10

<sup>44</sup> siehe [Bar08], S. 37

<sup>45</sup> Vgl. [Bar08], S. 26 sowie [Jan04b], S. 17 und [Höb07], S. 16

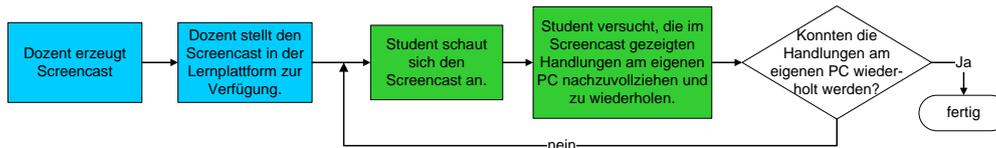


Abbildung 1: Flussdiagramm der Lernaktivität „Wissen durch Screencast erwerben“

## Beteiligte Akteure

Dozent und Studenten

Der Dozent erzeugt den Screencast. Die Studenten schauen sich den Screencast an und versuchen, die im Screencast gezeigten Handlungen im Selbststudium nachzuvollziehen.

## Implementierung

Zur Erzeugung der Screencasts können unterschiedliche Tools verwendet werden. Getestet wurden an der NORDAKDEMIE die Tools „QuickTime Pro“ unter Mac OS sowie „Camtasia“ unter Windows XP. Es hat sich als praktikabel erwiesen, die Screencasts als Flash-Datei zu speichern, da eine entsprechende SWF-Datei in jede Webseite (und damit auch in moodle) eingebettet werden kann. Unter <http://www.janbartelsen.de/nordakademie/nordpol/screencasts/systembrowser/> kann ein Beispiel-Screencast zum Thema „Systembrowser“ abgerufen werden.

### 4.2.2 Erläutern von Fachbegriffen

Beim Erlernen einer Programmiersprache tauchen eine Reihe von Fachbegriffen auf, die gelernt werden müssen. Wie in der Lerntheorie des Konstruktivismus beschrieben<sup>46</sup>, ist das Lernen deutlich effektiver, wenn der Lernstoff für andere aufbereitet wird. Aus diesem Grund sollen die Studenten die neuen Fachbegriffe selbst erklären. Damit bei der Erläuterung dieser Fachbegriffe bereits eine Qualitätskontrolle seitens der Studenten existiert, sollte jeder Fachbegriff durch eine Gruppe von zwei Teilnehmern erklärt werden. Hierfür eignet sich das Kommunikations-Werkzeug „Wiki“<sup>47</sup>. Bei der Erstellung der Wiki-Einträge sollte ein großer Wert auf wissenschaftliches Arbeiten und nachvollziehbare Quellen gelegt werden, um die Studenten schon früh mit dieser Arbeitsweise vertraut zu machen.

## Lernziel

Die Lernaktivität „Erläutern von Fachbegriffen“ hat drei Lernziele. Neben der Vermittlung von Fachkompetenz durch das Erlernen von Fachbegriffen soll als Nebeneffekt auch Medienkompetenz im Umgang von Shared-document-Systemen wie einem Wiki sowie die Teamfähigkeit über das Internet geschult werden. Dies sind zwei Kompetenzen, die im Zeitalter von globalen IT-Projekten von Bedeutung sein können. Ein weiteres Lernziel ist die Fähigkeit zur Offenlegung von nachvollziehbaren Quellen und wissenschaftlichem Arbeiten, was auf diese Weise bereits im ersten Semester geübt werden kann.

<sup>46</sup>Vgl. [Bar08], S. 26 und [SFD08]

<sup>47</sup>Vgl. Kapitel 2.3.3

## Ablauf

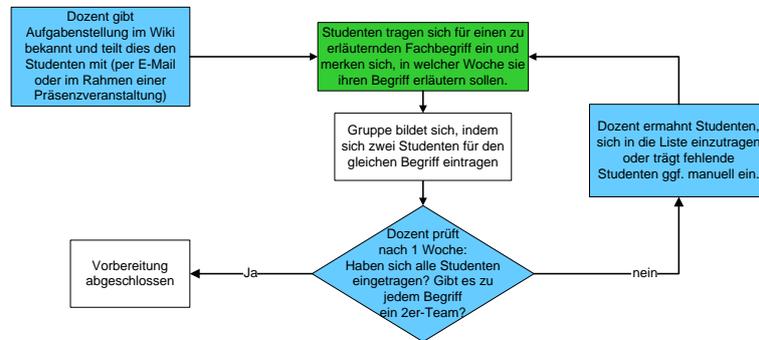
Der Ablauf dieser Lernaktivität ist wie folgt aufgebaut (vgl. auch Abbildung 2):

- Zunächst gibt der Dozent zu Beginn des Semesters die Aufgabenstellung sowie eine Liste von Fachbegriffen, die im Laufe des Semesters im Wiki erklärt werden sollen, im Wiki bekannt und informiert die Studenten per E-Mail oder im Rahmen einer Präsenzveranstaltung über die im Wiki stehende Aufgabe.
- Die Studenten wählen aus der Liste von Fachbegriffen ein Thema aus und melden sich für dieses Thema an. Die Zweier-Gruppe bildet sich dadurch, dass zwei Studenten den gleichen Fachbegriff gewählt haben.
- Der Dozent hat nach einer Woche zu prüfen, ob sich alle Studenten in die Liste eingetragen haben. Wenn nicht, sollte er die Studenten erinnern oder ggf. die noch offenen Begriffe manuell einem Zweier-Team zuweisen.
- Nun warten die Studenten, bis die Bearbeitungszeit für ihren Begriff beginnt. Die Bearbeitungszeiten sind so angelegt, dass die Begriffe immer in der Woche erklärt werden sollen, in der sie das erste Mal im Rahmen der Lehrveranstaltung genannt werden. Die Bearbeitungszeit beträgt jeweils eine Woche.
- Wenn die Bearbeitungszeit beginnt, entwirft ein Student aus dem Team (im Folgenden „Student1“ genannt) einen Text und schreibt einen Entwurf ins Wiki. Dieser Text muss nicht bereits die gesamte Fachbegriff-Erklärung sein. Es genügt eine Teil-Erklärung, wenn die beiden Gruppenmitglieder sich darauf verständigt haben, dass die Erklärung in zwei Teile aufgeteilt wurde.
- Sobald Student1 fertig ist, informiert er seinen Gruppenpartner (im Folgenden „Student2“ genannt), der sich den Text durchliest, begutachtet und ggf. Ergänzungen schreibt. Sofern die Studenten während der Bearbeitung der Aufgaben nicht nebeneinander sitzen (was an Präsenz-Hochschulen wie der NORDAKADEMIE durchaus üblich ist), kann die Kommunikation zwischen Student1 und Student2 asynchron über die in moodle integrierten „privaten Mitteilungen“ oder per E-Mail erfolgen, als auch synchron über Chat (wird von moodle zur Verfügung gestellt), Telefon oder Instant Messaging.
- Der zuletzt genannte Schritt wird solange wiederholt, bis alle Gruppenmitglieder mit dem Text zufrieden sind, wobei die Rollen zwischen Student1 und Student2 mit jeder Iteration getauscht werden. Außerdem sollte sichergestellt werden, dass der vorherige Schritt mindestens zweimal ausgeführt wird, so dass wirklich jedes Gruppenmitglied an dem Text mitgewirkt und Teile davon verfasst hat. Wie weiter unten erläutert wird, kann der Betreuer dies überprüfen, indem er sich anschaut, wer das Wiki-Dokument bearbeitet hat.
- Wenn alle beteiligten Studenten mit dem Text zufrieden sind, informieren sie den zuständigen Betreuer per E-Mail oder private Mitteilung.
- Der Betreuer begutachtet nun den Text und korrigiert kleinere Fehler. Bei größeren Fehlern oder Schwächen bittet er die Studenten, den Text zu überarbeiten.

**Legende:**

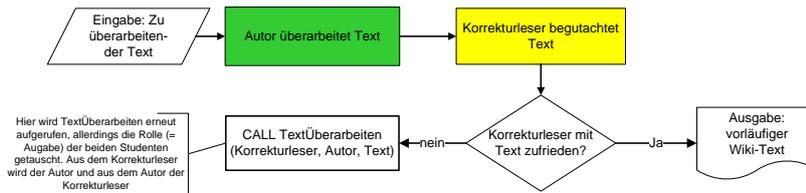


**Subroutine Vorbereitung**



**Subroutine TextÜberarbeiten (Autor, Korrekturleser, Text)**

**Rollen:**



**Hauptroutine**

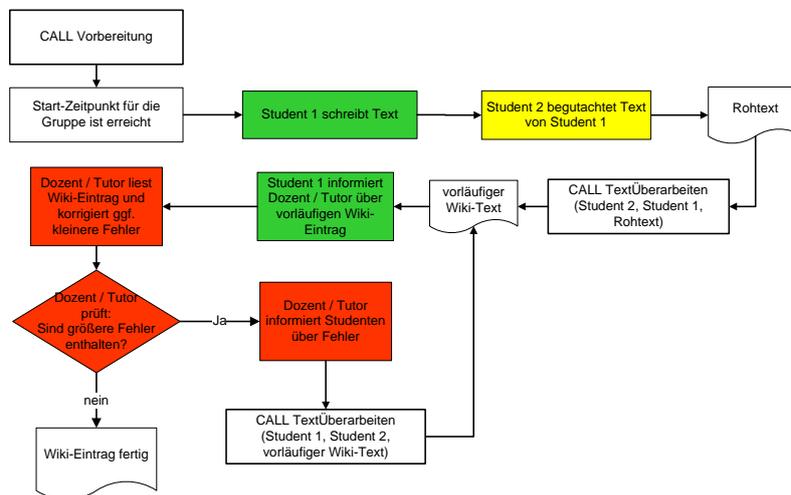


Abbildung 2: Flussdiagramm der Lernaktivität „Erläutern von Fachbegriffen“

Der Betreuer hat auch darauf zu achten, dass der Text wirklich von allen Gruppenmitgliedern gemeinsam erstellt wurde. Hierzu kann er im Wiki nachsehen, wer das Dokument bearbeitet hat und wer welche Änderungen durchgeführt hat. Taucht dort nur eine Person auf, gibt es zwei Möglichkeiten: Entweder die Gruppe hat sich offline getroffen, den Text gemeinsam erstellt und einer hat sich bereit erklärt, diesen zu posten. Es kann jedoch auch sein, dass der Text wirklich nur von einer Person erstellt wurde. Da der Dozent nicht überprüfen kann, welcher der beiden Fälle tatsächlich eingetreten ist, sollen die Studenten darauf achten, dass sie auch gemeinsam am Dokument arbeiten. Am Ende sollen alle Gruppenmitglieder zu gleichen Teilen als Autoren des Dokumentes erkennbar sein. Nur so kann das sekundäre Lernziel „Schulung der Teamfähigkeit über das Internet“ erreicht werden.

## **Beteiligte Akteure**

Dozent, Tutoren und Studenten

Der Dozent gibt die Aufgabenstellung und die zu erklärenden Fachbegriffe vor. Zusammen mit den Tutoren übt er als Betreuer-Team die Rolle der finalen Qualitätskontrolle aus, indem alle Wiki-Einträge gelesen und eventuell auftretende Fehler korrigiert werden, bzw. der Text zur Bearbeitung an die Studenten zurückgegeben wird.

Der Student ist Verfasser eines Wiki-Beitrages und gleichzeitig als „Peer-Reviewer“ auch Korrektur-Leser für Artikel oder Textabschnitte seiner Kommilitonen.

## **Implementierung**

Moodle stellt ein fertiges Wiki als Lernaktivität zur Verfügung, in das Texte und Bilder eingefügt werden können. Durch das moodle-Wiki besteht die Möglichkeit, jeden beliebigen Beitrag zu bearbeiten und ggf. zu korrigieren, wobei über den Menüpunkt „Änderungen“ verfolgt werden kann, wer an dem Dokument gearbeitet hat und was die Unterschiede zur vorherigen Version sind.

Die Korrektur des Textes kann sowohl durch andere Studenten als auch durch die Betreuer geschehen. Der Betreuer sollte hierbei jedoch bedenken, dass er nur kleinere Fehler korrigiert. Bei größeren Unstimmigkeiten sollte er die Studenten über eine private Mitteilung innerhalb von moodle auffordern, ihren Text zu überarbeiten und Fehler zu korrigieren.

Zur Auswahl des Fachbegriffes und der Bildung der Zweier-Gruppen eignet sich die moodle-Lernaktivität „Abstimmung“.

### **4.2.3 Report erstellen**

Bei größeren Programmierübungen ist es sinnvoll, dass die Studenten Reports anfertigen, in denen sie ihre Vorgehensweise bei der Programmierung ausführlich beschreiben. An dieser Stelle soll eine an der FHTW Berlin bereits praktizierte Lösung<sup>48</sup> aufgegriffen und modifiziert werden. Der Report soll als Einzelarbeit durch einen einzelnen Studenten angefertigt werden, wobei der Student im Report (stichpunktartig) berichten und mit Screenshots belegen soll:

- wie er an die Aufgabe herangegangen ist
- auf welche Schwierigkeiten er gestoßen ist (Irrwege beschreiben, um daraus lernen zu können)

---

<sup>48</sup>Vgl. [Bar08], S. 19f

- welche Fehler wie behoben wurden
- wer geholfen hat (Kommilitone, Tutor, Lehrbücher, Internet-Quellen / Foren)
- benötigter Zeitaufwand

Nach der Erstellung des Reports soll dieser in einem Wiki innerhalb des Kurses veröffentlicht werden. Um den Betreuungs- und Korrekturaufwand für den Dozenten bei über 100 Studenten nicht zu groß werden zu lassen, empfiehlt es sich allerdings, nicht mehr als zwei Übungsaufgaben pro Semester durch Reports beschreiben zu lassen.

## Lernziel

Lernziel des Reports ist es, dass die Studenten durch das Erstellen des Reports lernen, die eigenen Lernaktivitäten zu dokumentieren, um aus eventuell auftretenden Fehlern und Irrwegen lernen und später nachschlagen zu können, wie sie eigentlich vorgegangen sind. Für den Fall, dass die Studenten mit der Aufgabe überfordert sind oder deutlich mehr als die erwartete Zeit benötigen, bekommt der Dozent durch die Reports Hinweise zur Verbesserung der Lehrveranstaltung im kommenden Jahrgang.

Durch die Veröffentlichung der Reports innerhalb des Kurses, dienen sie auch als Nachschlagewerk für Kommilitonen. Ein Student kann so durch die Fehler und Lösungsansätze eines Kommilitonen lernen, falls er auf ähnliche Probleme gestoßen ist.

## Ablauf

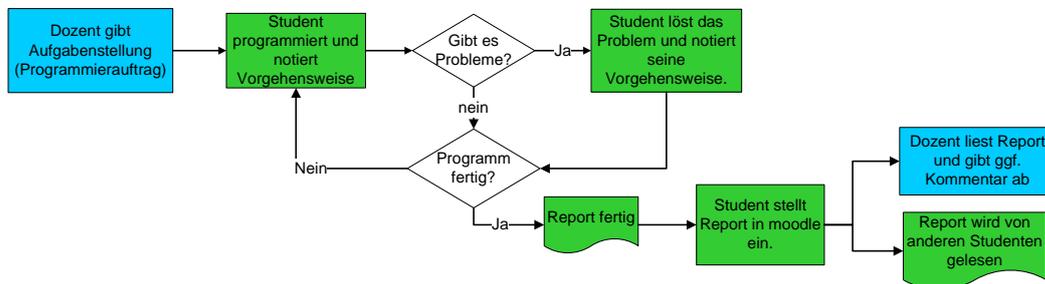


Abbildung 3: Flussdiagramm der Lernaktivität „Report erstellen“

Der Ablauf eines Reports sieht wie folgt aus (vgl. auch die Abbildung 3):

- Der Dozent gibt die Aufgabenstellung über die Lernplattform bekannt. Die Aufgabenstellung besteht darin, eine kleine Programmieraufgabe zu bearbeiten. Die Bearbeitung soll als Einzelarbeit erfolgen, der Student hat hierfür eine Woche Zeit.
- Der Student beginnt mit der Programmierübung und notiert parallel seine Aktivitäten im Report. Dazu fertigt er Screenshots von seinen Aktivitäten an.
- Sollte der Student auf Probleme stoßen, notiert er, wie er bei der Lösung des Problems vorgegangen ist. Um die Lösung später (auch vom Studenten selbst) nachvollziehen zu können, wird der genaue Lösungsweg einschließlich verwendeter Quellen und Hilfen (Kommilitonen, Internet-Foren, etc.) notiert.

- Wenn die Programmieraufgabe erfolgreich bearbeitet wurde und der Report fertig erstellt ist, lädt der Student seinen Report in die Lernplattform hoch, wo sie vom Dozenten oder einem Tutor gelesen und ggf. kommentiert werden kann, z.B. durch das Geben von Hinweisen für zukünftige Übungen, wenn der Student im Report einen umständlichen Lösungsweg beschrieben hat.
- Jeder Student kann nun die Reports der Kommilitonen einsehen und gezielt nach Stichwörtern durchsuchen.

## Beteiligte Akteure

Dozent, Tutoren und Studenten

Der Dozent gibt die Aufgabenstellung aus. Der Student hat die Rolle des Autors, der die eigenen Lernaktivitäten beschreibt. Der fertige Report kann sowohl vom Dozenten und von den Tutoren als auch von anderen Studenten gelesen und ggf. kommentiert werden.

## Implementierung

Für die Realisierung des Reports in moodle eignet sich die Lernaktivität „Wiki“. Hier können die Studenten ihren Report einstellen und zusammen mit entsprechenden Screenshots speichern. Ein kurzer Screencast<sup>49</sup> gibt eine Einführung in die Bedienung des Report-Wiki. Die Texte im Wiki können dann von jedem Kurs-Teilnehmer (Dozenten, Tutoren oder Kommilitonen) gelesen und ggf. kommentiert werden.

### 4.2.4 Eigenen Lernerfolg feststellen

Um den Studenten die Möglichkeit zur Selbstkontrolle zu geben, stehen innerhalb von Lernplattformen Tests mit unterschiedlichen Aufgabenstellungen zur Verfügung. Diese Tests können bezüglich ihrer Aufgabenstellung wie folgt unterschieden werden:

- Multiple-Choice, bzw. Single-Choice als Ankreuz-Test
- Wahr/Falsch-Test
- Zuordnungsaufgaben
- Lückentest
- Freifeld-Antwort

Bevor ein Test erstellt wird, ist zu überlegen, wie der Test im Rahmen des Unterrichtskonzeptes eingesetzt werden soll: Dient er als notenrelevante Überprüfung des aktuellen Lernerfolgs durch den Dozenten? Oder soll er der Selbstkontrolle durch die Studenten dienen? Aus pädagogischer Sicht ist es sinnvoll, den Test zur Selbstkontrolle als Wiederholung des Lerninhaltes anzubieten und dem Studenten die beliebige Wiederholung des Tests zu gestatten<sup>50</sup>. Wenn der Test bei einer falschen Antwort Hinweise zur richtigen Lösung bereitstellt, kann er dazu beitragen, den Lernstoff zu verinnerlichen und zu beherrschen<sup>51</sup>. Um zu verhindern, dass die Reihenfolge der richtigen Antworten auswendig gelernt wird, sollten die Fragen (und ggf. auch die Antwortmöglichkeiten) mit jedem Test-Durchgang neu gemischt werden<sup>52</sup>.

<sup>49</sup>Vgl. Kapitel 4.2.1

<sup>50</sup>Vgl. [Höb07], S. 94

<sup>51</sup>Vgl. [Höb07], S. 42

<sup>52</sup>Vgl. [Höb07], S. 94

Bei der Testerstellung ist darauf zu achten, dass vom Test-Ersteller mögliche falsche Antworten antizipiert werden, um hierzu ein gezieltes Feedback sowie Hinweise zur richtigen Lösung geben zu können. Folgendes Beispiel soll dies verdeutlichen:

Die Studenten sollen lernen, wie in Smalltalk Ausdrücke ausgewertet werden. Dazu wird Ihnen die Frage gestellt, was  $6 + 3 * 4$  ergibt. Wer einmal etwas von der Punkt-vor-Strich-Regel gehört hat, wird vermuten, dass die Lösung 18 ist. Bei Smalltalk ist dies jedoch nicht der Fall, da hier konsequent von links nach rechts ausgewertet wird. Das Ergebnis des genannten Ausdrucks ist folglich 36. Wenn ein Student als Ergebnis 18 wählt und nur den Hinweis erhält, dass die Antwort falsch sei, wird er verwundert sein, warum dies so ist und sich ggf. sogar beim Administrator beschweren, dass hier eine falsche Lösung hinterlegt ist. Besser ist es, als Test-Ersteller den möglichen Fehler „18“ zu antizipieren, um dem Studenten bei der Eingabe der 18 ein entsprechendes Feedback geben zu können. Dies könnte im konkreten Beispiel wie folgt aussehen: „Leider nicht richtig. Bedenken Sie bitte, dass bei Smalltalk von links nach rechts, ohne Berücksichtigung der Punkt-vor-Strich-Regel ausgewertet wird.“

## Lernziel

Ziel der Lernaktivität „eigenen Lernerfolg feststellen“ ist es, dass der Student überprüfen kann, ob er den Inhalt des aktuellen Themenblocks sicher verstanden hat und beherrscht.

## Ablauf

Der Ablauf der Lernaktivität „eigenen Lernerfolg feststellen“ sieht wie folgt aus (vgl. auch die Abbildung 4):

- Nachdem der Student den aktuellen Themenblock bearbeitet hat, kann er online einen Selbsttest durchführen. Dieser Selbsttest besteht in der Regel aus automatisch auswertbaren Fragen, so dass der Student ein sofortiges Feedback zu seinen Lösungsvorschlägen erhält.
- Wenn der Student eine Frage falsch beantwortet, bekommt er gezielte Hinweise, wie er zur richtigen Lösung findet.
- Der Student kann den Test anschließend erneut ausführen, wobei die Fragen (und bei Multiple-Choice, bzw. Single-Choice-Fragen auch die Antwortmöglichkeiten) gemischt werden.

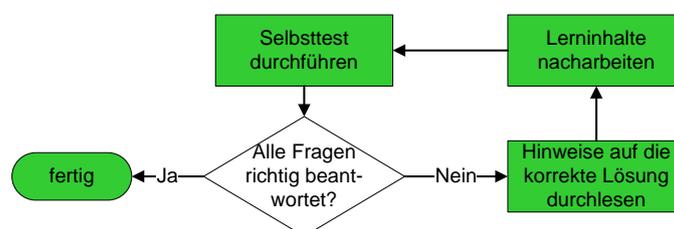


Abbildung 4: Flussdiagramm der Lernaktivität „eigenen Lernerfolg feststellen“

## Beteiligte Akteure

An dieser Lernaktivität ist nur der Student beteiligt. Er hat dabei jedoch nicht nur die Rolle des Lernenden, sondern auch die des Kontrolleurs, der den eigenen Lernerfolg überprüft. Wenn Fragen falsch beantwortet wurden, muss der Student in Eigenverantwortung den Lernstoff wiederholen und den Test erneut ausführen.

## Implementierung

Moodle stellt für jede der oben genannten Aufgabenstellungen eine Lernaktivität bereit, wobei die Reihenfolge der Fragen und Antworten gemischt werden kann. Außerdem kann die Zeit, die zur Ausführung eines Tests zur Verfügung steht, begrenzt werden.

### 4.2.5 Gruppenprogrammierung

Aus dem Bereich des Software-Engineering stammt der Begriff des „Pair Programming“, dem Programmieren in Paaren. Damit ist gemeint, dass immer zwei Programmierer gemeinsam an einem Rechner sitzen und dabei unterschiedliche Rollen einnehmen. Während einer der beiden eine Methode implementiert, denkt der zweite darüber nach, ob der Ansatz funktionieren kann und ob die gewählte Strategie richtig ist. Auch kontrolliert er, ob der Kollege eventuell einen Fehler gemacht hat. Dabei ist zu beachten, dass die Rollen zwischen beiden Partnern regelmäßig wechseln<sup>53</sup>. Sowohl Beck als auch Mellis sprechen davon, dass der entwickelte Code von deutlich besserer Qualität sei<sup>54</sup>. Jeder kann seine Stärken einbringen, wodurch Produktivität, Qualität und Zufriedenheit steigen<sup>55</sup>.

Daher soll auch im Rahmen einer Programmierausbildung das Programmieren in Paaren geübt werden. Übungsaufgaben, die vom Dozenten gestellt werden, sollen zu zweit programmiert werden. Dies kann entweder dadurch geschehen, dass die beiden Personen wie oben beschrieben an einem Rechner arbeiten oder ihre Gedanken über ein Kommunikations-Werkzeug zum kooperativen Arbeiten, wie z.B. Chat (in moodle integriert) oder Telefon-Konferenz, austauschen. Da moodle Kurs „Programmierung 1“ jedoch begleitend zu einer Präsenzveranstaltung eingesetzt wird, kann davon ausgegangen werden, dass die Studenten Gruppenaufgaben face-to-face bearbeiten, indem sie sich in einem Rechnerraum der Hochschule (oder zu Hause) treffen.

## Lernziel

Die Lernaktivität „Gruppenprogrammierung“ hat zwei Ziele: Zum einen sollen die Studenten zu zweit eine Programmieraufgabe lösen und so Praxiserfahrung im Umgang mit dem Lösen von Programmierproblemen sammeln. Dazu gehört auch das algorithmische Lösen und Formulieren von Problemen in Smalltalk-Code. Zum anderen soll gelernt werden, Lösungsalternativen miteinander oder mit dem Dozenten zu diskutieren.

## Ablauf

Nach dem Stellen der Übungsaufgabe durch den Dozenten bearbeiten die Studenten die Übungsaufgabe zu zweit. Bei Problemen können Sie den Dozenten oder einen Tutor fragen. Der Ablauf ist in Abbildung 5 dargestellt.

---

<sup>53</sup>Vgl. [Bec00], S. 58f und S. 100ff

<sup>54</sup>Vgl. [Bec00], S. 101 und [Mel04], S. 389

<sup>55</sup>Vgl. [Bec00], S. 100

## 4.2.6 Beteiligte Akteure

Dozent und Tutoren als Lernbegleiter.

Studenten als Lernende, die zu zweit eine Programmieraufgabe lösen. Innerhalb der Gruppenprogrammierung gibt es dabei, wie oben beschrieben, zwei Rollen, in denen sich die Studenten regelmäßig abwechseln.

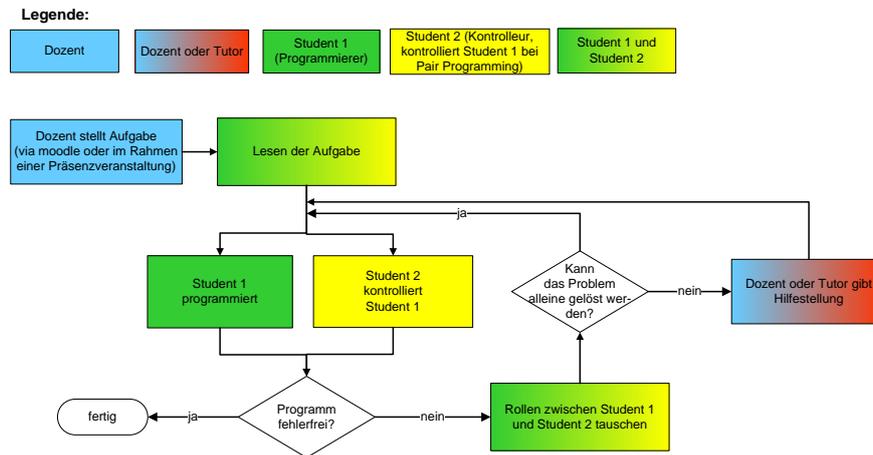


Abbildung 5: Flussdiagramm der Lernaktivität „Gruppenprogrammierung“

## 5 Ausblick

Die Wirksamkeit der in den voran gegangenen Kapiteln dargestellten didaktischen Mittel und Lernaktivitäten ist im weiteren Verlauf des Projekts noch gezielt und systematisch zu untersuchen. Dies gilt insbesondere für die in Kapitel 4.2 beschriebenen elementaren Lernaktivitäten, die in dem bisher verwendeten Online-Kurs entweder gar nicht oder nur sehr rudimentär verwendet wurden.

Bezüglich der didaktischen Mittel „Projektmethode“<sup>56</sup> bzw. „Problemorientiertes Lernen“<sup>57</sup> liegen hingegen erste Erfahrungen vor, die aus Gesprächen mit Studierenden während der Durchführung der Lehrveranstaltung und im Rahmen des regulären Lehrveranstaltungs-Feedbacks durch die Studierenden gewonnen wurden. Sowohl im ersten als auch im zweiten Semester arbeiten die Studierenden an jeweils einer einzigen größeren Programmieraufgabe. Die für die Erarbeitung der Lösung erforderlichen Teilaufgaben sind für das erste Semester in *moodle* sehr detailliert vorgegeben. Dies wird als notwendig erachtet, um die Variationsbreite der von den Studierenden erarbeiteten Lösungen einzugrenzen und damit für den Dozenten handhabbar zu machen. Die Studierenden schätzen sehr wohl die Möglichkeit, an einer durchgehenden Aufgabe zu arbeiten, dabei weitgehend selbständig voran schreiten und – wenn erforderlich – das „Coaching“ durch den Dozenten in Anspruch nehmen zu können.

Bei der Bearbeitung der Semesteraufgabe des zweiten Semesters wird den Studierenden bei ihrer Vorgehensweise – ihren nun schon vorhandenen Vorkenntnissen Rechnung tragend – größerer Freiraum eingeräumt, was insbesondere von den leistungsstärkeren Studierenden begrüßt

<sup>56</sup>Vgl. Kapitel 3.2

<sup>57</sup>Vgl. Kapitel 3.3

wird. Diese nehmen dann auch die Unterstützung durch den Dozenten in erheblich geringerem Umfang in Anspruch, was wiederum den nicht so Starken zugute kommt.

Die Wirksamkeit von didaktischen Maßnahmen exakt zu „messen“ ist jedoch kaum möglich, da keine Vergleichsgruppe zur Verfügung steht. Die Erkenntnisse werden letztlich auf die durch Befragung ermittelte „Kundenzufriedenheit“ zurück geführt werden müssen. Ob die eingangs geschilderten Defizite in der Verinnerlichung objektorientierter Konzepte verringert werden können, bleibt vorerst ungewiss. Geeignete Messmethoden hierfür müssen wohl erst noch erforscht werden.

## Literatur

- [Bar08] BARTELSEN, JAN-IVAR: *Entwurf einer kooperativen Lehrveranstaltung in moodle für die Grundausbildung in der Programmierung*. Master-Thesis an der FernUniversität Hagen, 2008.
- [Bec00] BECK, KENT: *Extreme Programming*. Addison Wesley, 2000.
- [Bra04] BRAUER, JOHANNES: *Grundkurs Smalltalk - Objektorientierung von Anfang an*. Vieweg Verlag, 2. Auflage, 2004.
- [CNHS05] CHERNOBILSKY, ELLINA, ANANDI NAGARAJAN und CINDY E. HMELO-SILVER: *Problem-based learning online: multiple perspectives on collaborative knowledge construction*. In: *CSCL '05: Proceedings of the 2005 conference on Computer support for collaborative learning*, Seiten 53–62. International Society of the Learning Sciences, 2005.
- [GdW04] GRUNE, CHRISTIAN und CLAUDIA DE WITT: *Pädagogische und didaktische Grundlagen*. In: HAAKE, JÖRG, GERHARD SCHWABE und MARTIN WESSER (Herausgeber): *CSCL-Kompodium*, Seiten 27–41. Oldenbourg Wissenschaftsverlag, 2004.
- [Ger07] GERTSCH, FREDI: *Das Moodle 1.8 Praxisbuch*. Verlag Addison-Wesley, 2007.
- [Hin04] HINZE, UDO: *Computergestütztes kooperatives Lernen: Einführung in Technik, Pädagogik und Organisation des CSCL*. Waxmann, 2004.
- [Höb07] HÖBARTH, ULRIKE: *Konstruktivistisches Lernen mit Moodle*. vwh Verlag, 2007.
- [Hof04] HOFFMANN, NICOLE: *Problemorientiertes Lernen*. In: HAAKE, JÖRG, GERHARD SCHWABE und MARTIN WESSER (Herausgeber): *CSCL-Kompodium*, Seiten 245 – 251. Oldenbourg Wissenschaftsverlag, 2004.
- [HSW04] HAAKE, JÖRG, GERHARD SCHWABE und MARTIN WESSER (Herausgeber): *CSCL-Kompodium*. Oldenbourg Verlag, 2004.
- [Jan04a] JANNECK, MICHAEL: *Projektorientierung*. In: HAAKE, JÖRG, GERHARD SCHWABE und MARTIN WESSER (Herausgeber): *CSCL-Kompodium*, Seiten 238–244. Oldenbourg Wissenschaftsverlag, 2004.
- [Jan04b] JANNECK, MONIQUE: *Lern- und kommunikationspsychologische Grundlagen*. In: HAAKE, JÖRG, GERHARD SCHWABE und MARTIN WESSER (Herausgeber): *CSCL-Kompodium*, Seiten 14 – 26. Oldenbourg Wissenschaftsverlag, 2004.

- [JJ04] JANNECK, MICHAEL und MONIQUE JANNECK: *Gruppen und Gruppenarbeit*. In: HAAKE, JÖRG, GERHARD SCHWABE und MARTIN WESSER (Herausgeber): *CSCL-Kompodium*, Seiten 42–53. Oldenbourg Wissenschaftsverlag, 2004.
- [Kli76] KLIX, FRIEDHART: *Information und Verhalten: kybernetische Aspekte der organis-mischen Informationsverarbeitung*. Verlag Hans Huber in Lizenz von Deutscher Verlag der Wissenschaften, Berlin, 3. Auflage, 1976.
- [Mel04] MELLIS, WERNER: *Projektmanagement der SW-Entwicklung*. Vieweg Verlag, 2004.
- [RS02] RIEKHOF, HANS-CHRISTIAN und HUBERT SCHÜLE: *E-Learning in der Praxis*. Gabler Verlag, 2002.
- [Sch01] SCHRÖDER, HARTWIG: *Didaktisches Wörterbuch*. Oldenbourg Wissenschaftsverlag, 3. Auflage, 2001.
- [SFD08] STENCEL, PRZEMYSŁAW, HELEN FOSTER und GUSTAV DELIUS: *moodle Docs: Philosophy*. <http://docs.moodle.org/en/Philosophy>, 2008. Zuletzt abgerufen am 12. Juli 2008.
- [SH04] SCHÜMMER, TILL und JÖRG HAAKE: *Kommunikation*. In: HAAKE, JÖRG, GERHARD SCHWABE und MARTIN WESSER (Herausgeber): *CSCL-Kompodium*, Seiten 66–79. Oldenbourg Wissenschaftsverlag, 2004.
- [Sla95] SLAVIN, ROBERT E.: *Cooperative Learning: theory, research and practice*. Allyn and Bacon, 1995.
- [SLH05] SCHÜMMER, TILL, STEPHAN LUKOSCH und JOERG M. HAAKE: *Teaching distributed software development with the project method*. In: *CSCL '05: Proceedings of th 2005 conference on Computer support for collaborative learning*, Seiten 577–586. International Society of the Learning Sciences, 2005.
- [SS04] SCHUBERT, SIGRID und ANDREAS SCHWILL: *Didaktik der Informatik*. Spektrum Verlag, 2004.
- [Tüc05] TÜCKE, MANFRED: *Psychologie in der Schule*. Lit-Verlag, 2005.
- [WH03] WESSNER, MARTIN und TORSTEN HOLMER: *Integration des kooperativen Lernens in die Didaktik von L3*. In: EHLERS, ULF-DANIEL, WOLFGANG GERTEIS, TORSTEN HOLMER und HELMUT W. JUNG (Herausgeber): *E-Learning-Services im Spannungsfeld von Pädagogik, Ökonomie und Technologie*, Seiten 70–82. Bertelsmann, 2003.