



NORDAKADEMIE
HOCHSCHULE DER WIRTSCHAFT

ARBEITSPAPIERE DER NORDAKADEMIE
ISSN 1860-0360

Nr. 2010-02

Abschlussbericht für das Projekt NORDPOL

**Jan Bartelsen
Johannes Brauer
Heiko Rehder**

Juni 2010

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



NORDAKADEMIE
HOCHSCHULE DER WIRTSCHAFT



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

Inhaltsverzeichnis

Abstract	1
1 Einleitung: Das Projekt NORDPOL	1
2 Die elementaren Lernaktivitäten	2
2.1 Überblick	2
2.2 Wissen durch Screencast erwerben	2
2.3 Erläutern von Fachbegriffen	2
2.4 Report erstellen	3
2.5 Eigenen Lernerfolg feststellen	4
2.6 Gruppenprogrammierung	5
3 Evaluation	5
3.1 Evaluationskonzept	5
3.2 Ergebnisse der ersten Befragung in Bezug auf die jeweiligen Lernaktivitäten	5
3.2.1 Screencasts	5
3.2.2 Fachbegriffe-Wiki	6
3.2.3 Report erstellen	6
3.2.4 Eigenen Lernerfolg feststellen	7
3.2.5 Gruppenprogrammierung	8
4 Erfahrungsbericht eines Lehrenden	9
5 Ausblick	10
Literaturverzeichnis	11

Abstract

Dieses Arbeitspapier gibt einen Überblick über die in den Jahren 2008 und 2009 an der NORDAKADEMIE durchgeführten Aktivitäten des Projektes NORDPOL (NORDakademie Programmierunterricht OnLine). Dies betrifft insbesondere fünf elementare Lernaktivitäten, die die Vorlesung ergänzen und den Einsatz der Lernplattform *moodle* gestaltet haben.

1 Einleitung: Das Projekt NORDPOL

In den Jahren 2008 und 2009 wurde an der NORDAKADEMIE - Hochschule der Wirtschaft in Elmshorn das Projekt NORDPOL (NORDakademie Programmierunterricht OnLine) von Professor Johannes Brauer und dem wissenschaftlichen Mitarbeiter Jan Bartelsen durchgeführt. Innerhalb dieses Projektes sollte unter Verwendung der Lernplattform *moodle*¹ ein Lernmodul für den Kurs *Programmierung 1* entwickelt werden, der an der NORDAKADEMIE im Bachelor-Studiengang *Wirtschaftsinformatik* in den ersten beiden Semestern durchgeführt wird. Thema dieses Kurses ist „Objektorientierung von Anfang an“ unter Verwendung der Programmiersprache *Smalltalk*.

Neben technischen Aspekten sollten bei der Entwicklung des Lernmoduls auch fachdidaktische Aspekte beleuchtet werden. Auf der Grundlage der Master-Thesis von Jan Bartelsen² wurden fünf elementare Lernaktivitäten entwickelt, die im Jahr 2008 in der Master-Thesis³ sowie im NORDAKADEMIE-Arbeitspapier „Kooperatives Lernen mit moodle in der Programmierausbildung“⁴ bereits ausführlich dargestellt wurden.

Durch die Gestaltung der Lehrveranstaltung als Blended-learning-Kurs sollte dem Dozenten die Möglichkeit gegeben werden, durch den verringerten Umfang von Frontalunterricht, die gewonnene Zeit für die Betreuung von Studierenden zu verwenden, die mit der Bearbeitung der Programmieraufgabe mehr Schwierigkeiten haben als andere.

Zwei Jahre später soll im vorliegenden Arbeitspapier nun auf den ersten Einsatz von NORDPOL zurückgeblickt werden, der im Studienjahrgang 2008 vom Wintersemester 2008/09 bis Sommersemester 2009 erfolgte. Die elementaren Lernaktivitäten, die in Kapitel 2 nochmals kurz dargestellt werden⁵, sondern rückblickend bewertet werden. Hierzu wurden zwei Evaluationen durchgeführt, die in Kapitel 3 beschrieben sind. Die Durchführung der Lehrveranstaltung *Programmierung 1* beim Jahrgang 2008 oblag neben Professor Brauer, der zwei von drei Zenturien⁶ übernommen hatte, dem Wissenschaftlichen Mitarbeiter Heiko Rehder, der NORDPOL aus der Sicht eines Dozenten in Kapitel 4 beschreiben wird.

¹<http://www.moodle.org>

²Bartelsen (2008)

³Bartelsen (2008)

⁴Bartelsen und Brauer (2008)

⁵für eine ausführliche Beschreibung sei auf Bartelsen (2008) oder Bartelsen und Brauer (2008) verwiesen

⁶Eine Zenturie ist an der NORDAKADEMIE eine klassenähnliche Gemeinschaft mit maximal 36 Studierenden.

2 Die elementaren Lernaktivitäten

2.1 Überblick

Wie im Arbeitspapier „Kooperatives Lernen mit moodle in der Programmierausbildung“⁷ beschrieben, wurden im Rahmen von *NORDPOL* fünf elementare Lernaktivitäten entwickelt. Da diese im genannten Papier bereits beschrieben wurden, sollen sie an dieser Stelle nur kurz wiederholt werden⁸. Es kann, ergänzend zum ersten Arbeitspapier⁹, nun jedoch berichtet werden, wie und in welchem Umfang die elementaren Lernaktivitäten in den durchgeführten Kurs *Programmierung 1* integriert wurden.

2.2 Wissen durch Screencast erwerben

Bei dieser Lernaktivität geht es darum, dass dem Studierenden ein Screencast als Video-Aufzeichnung des Dozenten-Bildschirms zur Verfügung gestellt wird¹⁰. Hierbei hat der Dozent (vor der Lehrveranstaltung) auf seinem Computer Programmierfähigkeiten durchführt und diese auf seinem Bildschirm, zusammen mit gesprochenen Erklärungen, aufgezeichnet. Zum genauen Ablauf dieser Lernaktivität siehe Bartelsen und Brauer (2008), S. 8ff oder Bartelsen (2008), S37ff.

Obgleich Bartelsen erwähnt, dass sich für einen Screencast eine Dauer von maximal fünf Minuten als ideal erwiesen hat und größere Screencasts u. U. Nachteile haben¹¹, weisen die im Rahmen von *NORDPOL* entwickelten Screencasts aus thematischen Gründen teilweise eine längere Spieldauer von bis zu 10 Minuten auf. Insgesamt wurden 15 Screencasts angelegt, die vom lehrenden Professor, einem wissenschaftlichen Mitarbeiter oder einer studentischen Hilfskraft aufgezeichnet und gesprochen wurden. Hierfür wurde die Software „Screenflow“ unter Mac OS sowie „Camtasia“ unter Windows XP verwendet¹². 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.

2.3 Erläutern von Fachbegriffen

Beim Erlernen einer Programmiersprache tauchen eine Reihe von Fachbegriffen auf, die gelernt werden müssen. Wie in der Lerntheorie des Konstruktivismus beschrieben¹³, ist das Lernen deutlich effektiver, wenn der Lernstoff für andere aufbereitet wird. Aus diesem Grund sollen die Studenten die neuen Fachbegriffe selbst erklären, was zur Entwicklung des „Fachbegriffe-Wiki“ führte.

Entgegen der ursprünglichen Planung¹⁴ wurde ein zweistufiges Verfahren verwendet, welches wie folgt aufgebaut wurde:

⁷Bartelsen und Brauer (2008)

⁸Weitere Informationen zu den entwickelten Lernaktivitäten sind in Bartelsen und Brauer (2008) oder Bartelsen (2008) zu finden.

⁹Bartelsen und Brauer (2008)

¹⁰Vgl. Bartelsen und Brauer (2008), S. 8ff

¹¹siehe Bartelsen (2008), S. 37

¹²Für Details zur mit der Auswahl der Software verbundenen Problemen siehe Bartelsen (2008), S. 68

¹³Vgl. Bartelsen (2008), S. 26 und Stencel u. a. (2009)

¹⁴Vgl. Bartelsen und Brauer (2008), S.10

Das Unterrichtskonzept der NORDAKADEMIE sieht bzgl. des Kurses *Programmierung 1* vor, dass die Studierenden in Kleingruppen zu jeweils maximal 18 Studierenden eingeteilt werden und der Programmier-Unterricht in diesen Kleingruppen stattfindet. Daher war es naheliegend, diese bereits existierenden Kleingruppen auch für das Fachbegriffe-Wiki zu verwenden. So wurde die Stufe I des Fachbegriffe-Wiki innerhalb der Kleingruppen durchgeführt. Hierzu bekam jede Gruppe eine Liste aus 11 Fachbegriffen (wobei jede Gruppe dieselben Begriffe bekommen hat), wobei sich die Studierenden als 2er-Team einen Begriff aussuchen sollten, den sie erklären wollten. Da sich in jeder Gruppe maximal 18 Studierende befunden haben, wurden nicht alle Begriffe ausgewählt, was aber seitens der Dozenten nicht als bedeutsam erachtet wurde. Die Wahl der Fachbegriffe wurde mithilfe der *moodle*-Aktivität *Abstimmung* durchgeführt, bei der jeweils zwei Studierende einen Begriff gewählt haben. Die Studierenden haben dann im Verlauf des 1. Semesters zu zweit ihren jeweiligen Fachbegriff erklärt. Dies sollte immer in genau der Woche geschehen, in der der jeweilige Begriff erstmals in der Vorlesung genannt wurde, so dass am Ende des Semesters sechs Gruppen-Wikis zur Verfügung standen.

Zu Beginn des zweiten Semesters haben die Dozenten schließlich das Gerüst für ein finales Wiki (Stufe II) bereit gestellt, wobei dieses Wiki nicht mehr in sechs Teilgruppen unterteilt wurde. Es wurde nur ein einziges Wiki zur Verfügung gestellt, welches von den Studierenden aller sechs Teilgruppen mit Inhalt gefüllt wurde. Alle 11 Begriffe sollten in diesem Wiki erklärt werden und für jeden Begriff, der hier erklärt würde, würden die Dozenten eine Qualitätssicherung durchführen, so dass am Ende ein fertiges und gefülltes Wiki als Hilfe zur Klausurvorbereitung zur Verfügung steht. Dabei war den Dozenten wichtig, dass nicht einfach durch „Copy-and-Paste“ der Text aus Stufe I herüberkopiert wurde. Im besten Fall gab es schließlich von sechs Gruppen jeweils eine Erklärung zu diesem Begriff. Die Studierenden wurden explizit aufgefordert, aus ihren sechs Texten einen gemeinsamen Text zu erstellen und in kooperativer Wiki-Arbeit eine gemeinsame Erklärung (mit dem Besten aus den sechs Teilgruppen) zu erstellen.

Etwa vier Wochen vor der Klausur wurden die Erklärungen von den Dozenten überprüft. Doch erfreulicherweise mussten die Dozenten selten Texte verbessern oder korrigieren, da die Erklärungen der Studierenden von guter Qualität waren. So konnten die Studierenden eine selbsterstellte (und von den Dozenten geprüfte) Erklärung von Fachbegriffen als Vorbereitung auf die Klausur nutzen.

Ein grober Ablauf dieser Lernaktivität ist in Abbildung 1 skizziert. Für weitere Details sei auf das NORDAKADEMIE-Arbeitspapier Nr. 2010-01 „Kooperatives Lernen mit einem Wiki“¹⁵ verwiesen.

2.4 Report erstellen

Bei dieser Lernaktivität wurde eine an der FHTW Berlin¹⁶ bereits praktizierte Lösung¹⁷ aufgegriffen und modifiziert werden. Der Report soll als Einzelarbeit durch einen einzelnen Studenten parallel zur Bearbeitung einer Programmieraufgabe angefertigt werden, wobei der Student im Report (stichpunktartig) berichten und mit Screenshots belegen soll¹⁸:

¹⁵Bartelsen und Brauer (2010)

¹⁶heute HTW Berlin

¹⁷Vgl. Bartelsen (2008), S. 19f

¹⁸Vgl. Bartelsen und Brauer (2008), S. 13ff

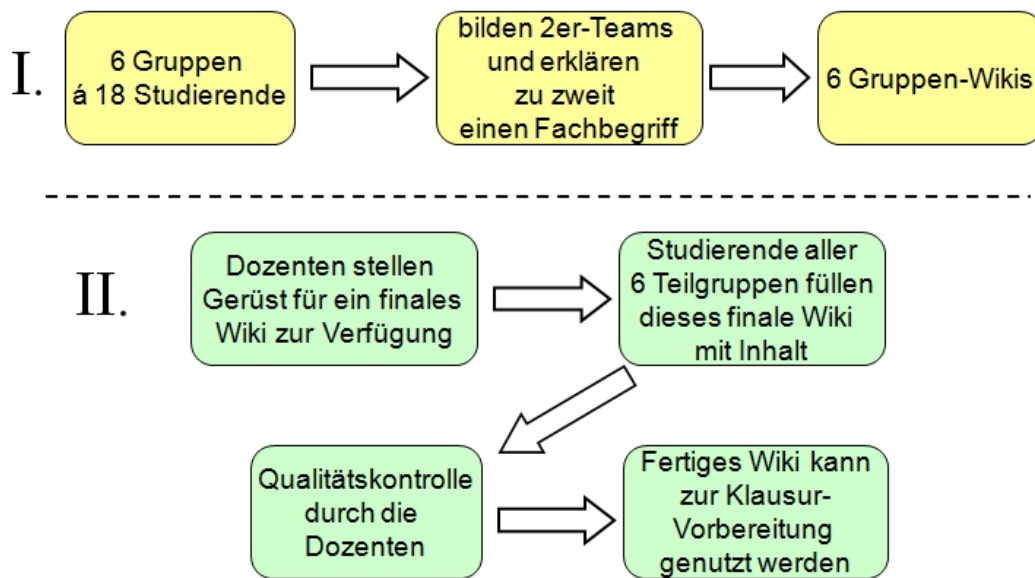


Abbildung 1: Das Fachbegriffe-Wiki in zwei Stufen

- wie er an die Aufgabe herangegangen ist
- auf welche Schwierigkeiten er gestoßen ist (Irrwege beschreiben, um daraus lernen zu können)
- welche Fehler wie behoben wurden
- wer geholfen hat (Kommilitone, Tutor, Lehrbücher, Internet-Quellen / Foren)
- benötigter Zeitaufwand

Nach der Erstellung des Reports wurde dieser in einem Wiki innerhalb des Kurses veröffentlicht, um so ggf. als Nachschlagewerk für Kommilitonen zu dienen, die bei der Programmierung auf ähnliche Fehler stoßen¹⁹. Zwei Übungsaufgaben des ersten Semesters wurden auf diese Weise begleitet und durch Reports beschrieben.

2.5 Eigenen Lernerfolg feststellen

Um den Studenten die Möglichkeit zur Selbstkontrolle zu geben, stehen innerhalb des *moodle*-Kurses *Programmierung 1* verschiedene Selbsttests zur Verfügung. Im Rahmen des NORDPOL-Konzeptes sollen diese Tests rein zur Selbstkontrolle der Studierenden dienen, d.h. dass die Studierenden direkt nach der Bearbeitung ein Feedback erhalten und den Test beliebig oft wiederholen können. Um zu verhindern, dass die Reihenfolge der richtigen Antworten auswendig gelernt wird, werden die Fragen (und ggf. auch die Antwortmöglichkeiten) mit jedem Test-Durchgang neu gemischt²⁰.

Um die Fragen möglichst automatisch auswerten zu können, wurden hauptsächlich Wahr/Falsch-Fragen verwendet, teilweise auch der *moodle*-Fragetyp *Kurzantwort*.

Für weitere Informationen siehe Bartelsen und Brauer (2008), S. 15ff oder Bartelsen (2008), S. 46ff..

¹⁹Vgl. Bartelsen und Brauer (2008), S. 14

²⁰Vgl. Höbarth (2007), S. 94

2.6 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 wechseln²¹. Sowohl Beck als auch Mellis sprechen davon, dass der entwickelte Code von deutlich besserer Qualität sei²². Jeder kann seine Stärken einbringen, wodurch Produktivität, Qualität und Zufriedenheit steigen²³.

3 Evaluation

3.1 Evaluationskonzept

Die 100 Studierenden des Jahrgangs 2008 im Studiengang *Wirtschaftsinformatik* wurden an zwei Stellen zum NORDPOL-Konzept befragt. Das erste Mal nach dem ersten Semester des Kurses, an dieser Befragung nahmen 54 Studierende teil (die Rücklaufquote beträgt folglich 54%²⁴). Die zweite Befragung wurde nach dem zweiten Semester durchgeführt. Die Befragung erfolgt jeweils mithilfe des *moodle*-Feedback-Moduls.

3.2 Ergebnisse der ersten Befragung in Bezug auf die jeweiligen Lernaktivitäten

3.2.1 Screencasts

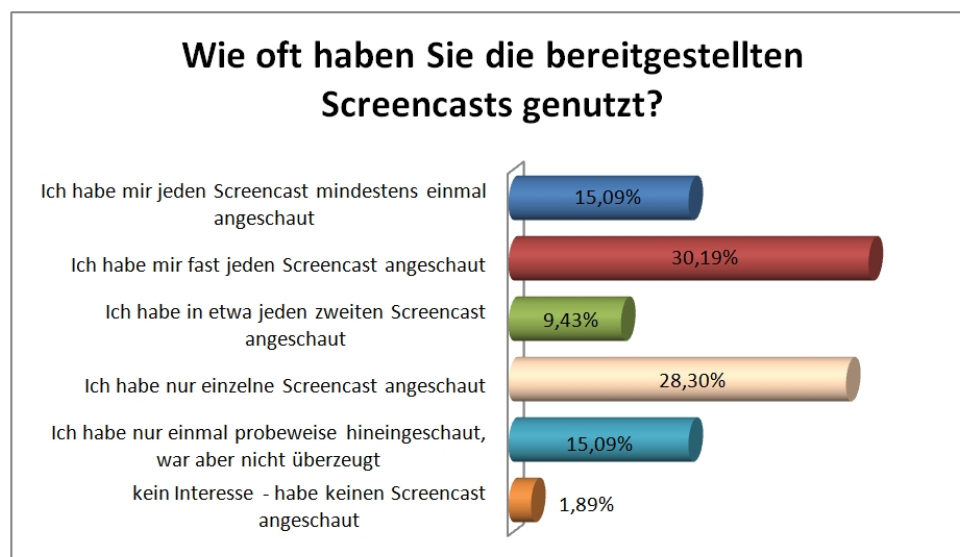


Abbildung 2: Auswertung über die Nutzungshäufigkeit der Screencasts

²¹Vgl. Beck (2000), S. 58f und S. 100ff

²²Vgl. Beck (2000), S. 101 und Mellis (2004), S. 389

²³Vgl. Beck (2000), S. 100

²⁴es gab im 1. Semester exakt 100 Studierende

Eine Auswertung über die Nutzungshäufigkeit zeigt Abbildung 2. Zusammengefasst kann gesagt werden, dass 83% der Teilnehmer von den Screencasts überzeugt waren. In der offenen Frage „Was fanden Sie an den Screencasts besonders gut oder besonders schlecht?“ wurden verschiedene Punkte genannt. Als negativ wurde die Sprechqualität genannt, da sich einige der Befragten gerne einen professionellen Sprecher gewünscht hätten. Da dies jedoch zum einen schwerer umsetzbar ist und zum anderen die Screencasts authentischer wirken, wenn der Professor (oder Mitarbeiter des Fachbereichs) die Screencasts vertonen, wurde sich gegen diese Möglichkeit entschieden. Als positive Aussagen wurde (unter anderem) genannt: „Screencasts sind eine geniale Möglichkeit Inhalte nachzubereiten, sei es wegen Abwesenheit oder einfach nur um das Wissen zu vertiefen.“ oder „Ich finde die Screencast eine sehr sinnvolle Sache. Insbesondere wenn man Dinge wiederholen möchte vor dem 2. Semester.“.

Die Länge der Screencasts empfanden in einer weiteren Frage 79,55% der Befragten als „genau richtig“ und 20,45% als „zu lang“ („zu kurz“ wurde nicht angekreuzt).

Der Aussage „Die Screencasts haben mir geholfen, den Lernstoff zu verstehen“ stimmten 13,73% vollständig²⁵ zu, 33,33% stimmten überwiegend zu. Auch dies könnte als Indikator für den Erfolg dieser Lernaktivität gesehen werden.

3.2.2 Fachbegriffe-Wiki

Eine detaillierte Evaluation des Fachbegriffe-Wikis ist im NORDAKADEMIE-Arbeitspapier 2010-01 „Kooperatives Lernen mit einem Wiki“²⁶ ab Seite 8 enthalten, so dass an dieser Stelle darauf verwiesen werden soll. Zusammenfassend lässt sich sagen, dass zwar 25% der Befragten gesagt haben, dass ihnen das Erklären der Fachbegriffe beim Verständnis geholfen hat²⁷, dennoch ein Großteil der Studierenden der Meinung ist, dass eine solche Fachbegriffe-Übersicht vom Dozenten erstellt werden sollte (39,58% in der ersten Befragung und 55,56% in der zweiten Befragung²⁸). Dennoch erachten die Dozenten das Fachbegriffe-Wiki weiterhin als didaktisch sinnvoll. Ferner ist es aufgrund der Tatsache, dass das Fachbegriffe-Wiki eine zusätzliche Aufgabe war (neben den Programmieraufgaben) auch nachvollziehbar, dass die Studierenden die Möglichkeit, dass ein anderer die Aufgabe für sie erledigt, gerne in Betracht ziehen²⁹.

3.2.3 Report erstellen

Wie aus Abbildung 3 ersichtlich, war auch die Beurteilung der Lernaktivität „Report erstellen“ unterschiedlich. Während ein Drittel der Studierenden der Aussage zustimmen kann, aus den Fehlerbeschreibungen viel gelernt zu haben, empfinden 37,78% der Befragten diese Übung als reine Zeitverschwendung. Erschwerend kam dazu, dass im ersten Durchlauf von *NORDPOL* den Studierenden zu spät mitgeteilt wurde, dass zu einer Programmieraufgabe ein Report erstellt werden sollte und diese Übung von vielen nicht beachtet wurde. Aus der Beantwortung der Freitext-Frage zu den Reports war zu entnehmen, dass einige Studierende gar nicht wussten, dass es so etwas gibt. Hier ist für zukünftige Jahrgänge eine bessere Kommunikation und Erklärung dieser Aufgabenstellung erforderlich.

²⁵gefragt wurde in der Umfrage, ob diese Aussage „zu 100% zutrifft“

²⁶Bartelsen und Brauer (2010)

²⁷Vgl. Bartelsen und Brauer (2010), S. 9

²⁸Vgl. Bartelsen und Brauer (2010), S. 9

²⁹Vgl. Bartelsen und Brauer (2010), S. 10

Zur Aussage „Die Umsetzung in moodle könnte besser sein“ wurde u.a. der integrierte WYSIWYG-Editor vom *moodle*-Wiki bemängelt. Hier wird mit Spannung auf die Version *moodle 2.0* gewartet, die ein neues Wiki-System enthalten soll³⁰.

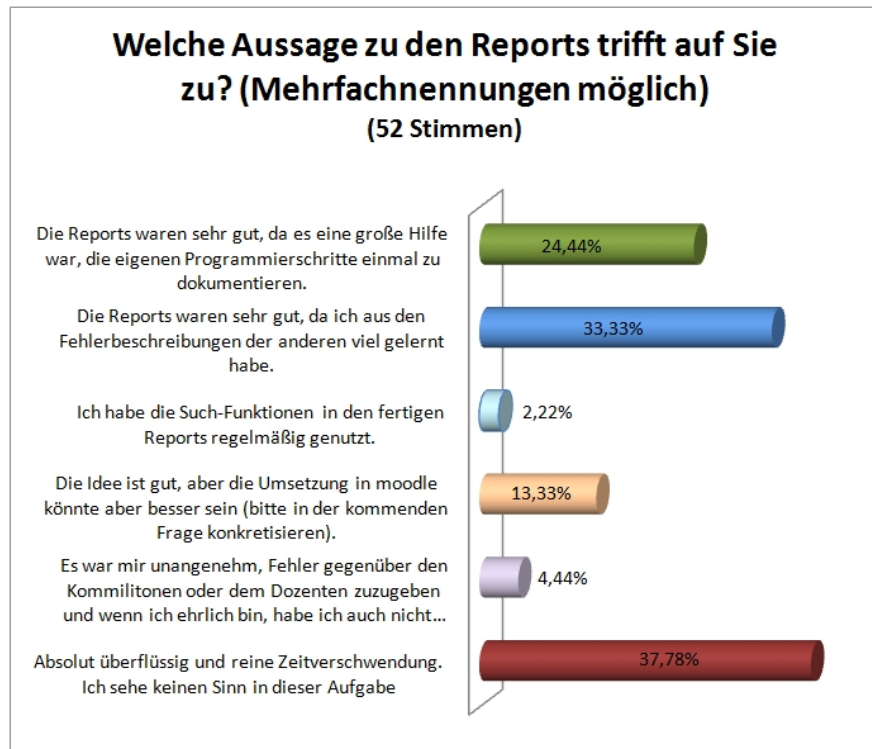


Abbildung 3: Auswertung zur Lernaktivität „Report erstellen“

3.2.4 Eigenen Lernerfolg feststellen

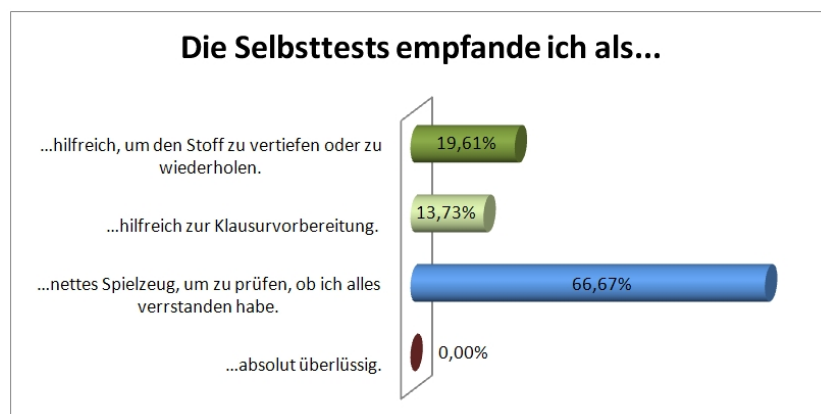


Abbildung 4: Auswertung 1 zur Lernaktivität „Eigenen Lernerfolg feststellen“

³⁰siehe hierzu http://docs.moodle.org/en/Development:Wiki_2.0, zuletzt abgerufen am 03.06.2010

Die Selbsttests zu Themen des Kurses *Programmierung 1* wurden von den Studierenden sehr gut angenommen und von keinem als „überflüssig“ empfunden (siehe Abbildung 4). Die starke Nutzung dieser Lernaktivität (siehe Abbildung 5) lässt die Idee aufkommen, die Selbsttests auszuweiten und weitere Tests anzubieten. Dies ist jedoch zu Beginn stets mit einem hohen Erstellungsaufwand seitens des Dozenten verbunden. Hier wäre es wünschenswert, sich mit anderen Hochschulen auszutauschen, um Synergieeffekte nutzen zu können.

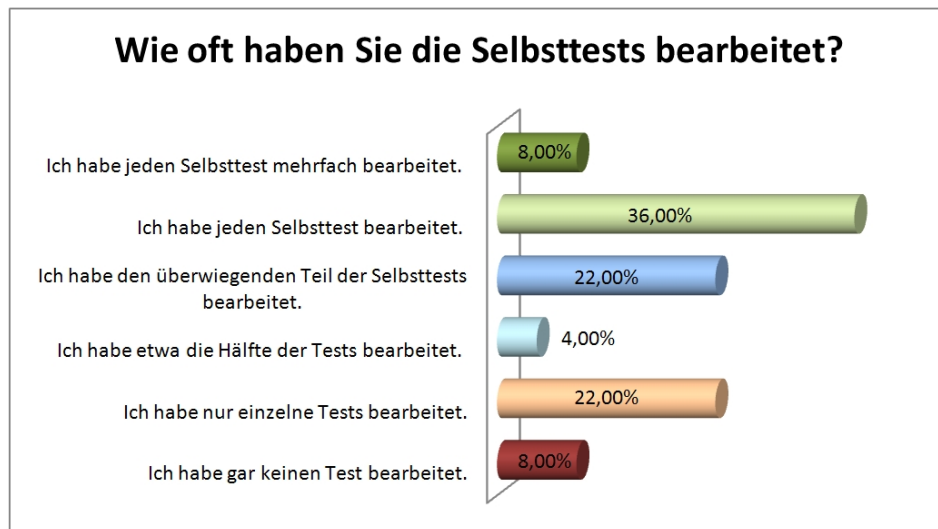


Abbildung 5: Auswertung 2 zur Lernaktivität „Eigenen Lernerfolg feststellen“

3.2.5 Gruppenprogrammierung

Zu diesem Thema wurden den Studierenden folgende offene Fragen gestellt:

1. Beurteilen Sie die Arbeit mit 2er-Teams als positiv?
2. Haben Sie das gemeinsame Programmieren praktiziert?
3. Halten Sie es für sinnvoll, dass Sie zu zweit gleichzeitig am Programm arbeiten?

Den zahlreichen und teilweise recht ausführlich gegebenen Antworten ist zu entnehmen, dass die Studierenden die Gruppenarbeit überwiegend positiv beurteilen. Als wichtigste Gründe können dabei die Möglichkeit, sich gegenseitig über Probleme bei der Bewältigung der Aufgabenstellung hinweg helfen zu können, und der intensive Diskussionsprozess über Problemlösungen innerhalb der Teams identifiziert werden.

Da die Studierenden die Zweiergruppen selbständig bilden durften, kamen bei diesem Prozess naturgemäß unterschiedliche Strategien zum Zuge. Wenn man die Studierenden bezüglich ihrer Vorkenntnisse in der Programmierung vereinfachend in die Kategorien „Anfänger“ und „Experte“ einteilt³¹, waren in den Teams alle drei möglichen Kombinationen dieser Kategorien anzutreffen. Wollte man die Arbeit der Gruppenmitglieder an den „Idealen“ des Pair Programming messen, wäre dies sicher nur bei etwa „gleich starken“

³¹beide Kategorien können natürlich auch durch eine Studentin besetzt sein. Die männliche Form wurde hier lediglich aus Gründen der einfachen Lesbarkeit gewählt.

Partnern angemessen. Vereinzelt Äußerungen von Studierenden geben zu der Vermutung Anlass, dass Teams, deren Mitglieder stark divergierende Vorkenntnisse aufwiesen, weniger gut „funktioniert“ haben. Es gab offenbar Experten, die auf Anfänger-Partner eher herab geschaut haben. Umgekehrt beklagten sich Anfänger, dass ihr Experte sie kaum an der Arbeit beteiligt habe. In welchem Umfang diese Probleme tatsächlich auftraten, lies sich aus den Befragungsergebnissen aber nicht ermitteln. Im ersten Semester können die Vorkenntnisse nicht bei allen so weit nivelliert werden, so dass derartige Effekte ausgeschlossen werden könnten. Andererseits legt es den Entschluss nahe, der von einigen Studierenden erhobenen Forderung, Gruppenarbeit auch schon im ersten Semester zu ermöglichen, nicht nachzukommen.

In Einzelfällen haben Studierende ihren Unwillen, in einer Zweiergruppe zu arbeiten, dadurch zum Ausdruck gebracht, dass sie das Team zwar („zwangsweise“) gebildet aber anschließend weitgehend unabhängig voneinander die Programmieraufgabe bearbeitet haben.

Ebenfalls vereinzelt wurde beklagt, dass die Entwicklungsumgebung bzw. das Code-Repository wirkliche Parallelarbeit gar nicht unterstützte. Dabei handelt es sich zwar um eine klare Fehleinschätzung, aber sie beleuchtet die Schwierigkeit, im Rahmen der Programmier-Grundausbildung auch noch den effektiven Umgang mit einem komplexen Entwicklungssystem unterrichten zu müssen.

Der über eine Abstimmung in *moodle* abgewickelte Teambildungsprozess selbst wurde von der großen Mehrheit der Studierenden positiv beurteilt. Probleme wird es bei einigen aber schon deshalb gegeben haben, weil nicht jeder Teilnehmer seinen Wunschpartner bekommen kann.

Gruppenarbeit in der Programmiergrundausbildung ist an der NORDAKADEMIE teilweise auch schon vor der Durchführung von NORDPOL praktiziert worden. Die bisher „gefühlte“ positive Einschätzung ist durch die Umfrage im Rahmen von NORDPOL in vollem Umfang bestätigt worden.

4 Erfahrungsbericht eines Lehrenden

Der Kurs *Programmierung 1* wurde in drei Zenturien³² gelehrt, wobei zwei Zenturien von Professor Johannes Brauer und eine Zenturie von dem wissenschaftlichen Mitarbeiter Heiko Rehder unterrichtet wurde.

Herr Rehder schreibt hierzu:

„Für mich als Dozent mit erst geringer Lehrerfahrung war der Online-Kurs eine große Hilfe. Zum einen, da der Kurs einen Rahmen bot, an dem ich mich als Dozent orientieren konnte, zum anderen weil den Studenten die Möglichkeit geboten wurde, in Ihrem eigenen Tempo voranzuschreiten und bei Bedarf auch einzelne Abschnitte jederzeit wiederholen zu können. Den einzelnen Lernaktivitäten kamen dabei unterschiedlich starke Bedeutungen zu. Die Screencasts wurden stark frequentiert und auch außerhalb der Vorlesungszeit genutzt, bspw. in der Praxisphase zur Wiederholung und Vertiefung. Im ersten Jahr der Nutzung rief das Erstellen der Reports bei vielen Studenten Unmut hervor, weil sie die im Kurs zur Verfügung gestellten Aufgaben sequentiell abarbeiteten und die Aufgabe der Reporterstellung am Ende der einzelnen Aufgabenblöcke stand. Daher begannen die Studenten erst mit der Erstellung des Reports nachdem die vorhergehenden

³²Eine Zenturie ist an der NORDAKADEMIE eine klassenähnliche Gemeinschaft mit maximal 36 Studierenden.

Aufgaben bearbeitet wurden. Sinnvoller wäre eine Bearbeitung der Aufgaben mit parallelem Reporting gewesen, so wie es ursprünglich (s. 2.4) angedacht war. Die Möglichkeit, den individuellen Lernerfolg festzustellen wurde von den Studenten gern aufgenommen und angewandt. Teilweise wurden Tests auch zur Prüfungsvorbereitung wiederholt.

Die Aktivität der Gruppenprogrammierung wurde von einzelnen Studenten ganz unterschiedlich genutzt. Während einige Gruppen sich gemäß dem Prinzip des Pair Programming abwechselnd die Aufgabe des Programmierens und Überprüfens teilten, nutzen andere Gruppen diese Möglichkeit nicht oder nur begrenzt, insbesondere bei Gruppen wo Programmierkenntnisse unterschiedlich stark verteilt waren. Hier war es notwendig, die Studierenden anzuhalten, die Prinzipien des Pair Programming anzuwenden, was insgesamt zu einem besseren Verständnis der objektorientierten Programmierung führte.

Zusammenfassend ist der Onlinekurs eine große Unterstützung sowohl für den Lehrenden als auch für die Studierenden.“

5 Ausblick

Im Projekt NORDPOL wurden erste Erfahrungen im systematischen Einsatz der Lernplattform *moodle* an der NORDAKADEMIE gesammelt, die, wie oben beschrieben, teils positiv, teils verbesserungswürdig sind. Die Arbeit an der Frage, wie man Studierenden im ersten Semester am Besten das Programmieren beibringt, ist damit jedoch noch nicht beantwortet (und war ja auch nicht Ziel des Projektes). Auch sind die Ergebnisse, die mit dem Studienjahrgang 08 gemacht wurden, nicht repräsentativ für die Menge aller Studierenden.

Es ergeben sich derzeit vor allem folgende Fragestellungen:

- Wird ein Einsatz des *NORDPOL-moodle*-Kurses in anderen Jahrgängen ähnliche Ergebnisse bringen? Hierzu wird der Kurs auch beim Jahrgang 09 (Wintersemester 2009/2010 sowie Sommersemester 2010) eingesetzt und am Ende ebenfalls evaluiert werden.
- Wie kann man beim Erstellen von Selbsttests Synergieeffekte, bspw. durch den Austausch mit anderen Hochschulen, nutzen?
- Sollte die Programmiergrundausbildung vielleicht dahingehend ausgebaut werden, dass eine stärkere Integration von Lern- und Entwicklungsumgebung erfolgt? Das NORDAKADEMIE-Arbeitspapier Nr. 2010-03 *Ideen für die Neugestaltung der Programmierausbildung durch Entwicklung von Webanwendungen*³³ wird sich mit dieser Frage beschäftigen.

Vor allem eine Fokussierung auf die letzte Fragestellung soll Gegenstand der zukünftigen Forschungstätigkeiten der NORDAKADEMIE im Themenfeld der Programmiergrundbildung sein.

³³Bartelsen u. a. (2010)

Literatur

- [Bartelsen und Brauer 2008] BARTELTSEN, Jan ; BRAUER, Johannes: *Kooperatives Lernen mit moodle in der Programmierausbildung*. Arbeitspapier der NORDAKADEMIE, Nr. 2008-06. 2008
- [Bartelsen und Brauer 2010] BARTELTSEN, Jan ; BRAUER, Johannes: *Kooperatives Lernen mit einem Wiki*. Arbeitspapier der NORDAKADEMIE, Nr. 2010-01. 2010
- [Bartelsen u. a. 2010] BARTELTSEN, Jan ; BRAUER, Johannes ; REHDER, Heiko: *Ideen für die Neugestaltung der Programmierausbildung durch Entwicklung von Webanwendungen*. Arbeitspapier der NORDAKADEMIE, Nr. 2010-03. 2010
- [Bartelsen 2008] BARTELTSEN, Jan-Ivar: *Entwurf einer kooperativen Lehrveranstaltung in moodle für die Grundausbildung in der Programmierung*. Master-Thesis an der FernUniversität Hagen. 2008
- [Beck 2000] BECK, Kent: *Extreme Programming*. Addison Wesley, 2000
- [Höbarch 2007] HÖBARTH, Ulrike: *Konstruktivistisches Lernen mit Moodle*. vwh Verlag, 2007
- [Mellis 2004] MELLIS, Werner: *Projektmanagement der SW-Entwicklung*. Vieweg Verlag, 2004
- [Stencel u. a. 2009] STENCEL, Przemyslaw ; FOSTER, Helen ; DOUGIAMAS, Martin ; DELIUS, Gustav ; HUNT, Tim: *moodle Docs: Philosophy*. <http://docs.moodle.org/en/Philosophy>. 2009. – Zuletzt geändert am 3. Mai 2009, Webseite abgerufen am 16. März 2010