

# Just-in-Time Teaching für Software-Engineering

Timo Kamph, Peter Salden, Sibylle Schupp und Christian Kautz

Technische Universität Hamburg-Harburg

{timo.kamph,peter.salden,schupp,kautz}@tuhh.de

## Zusammenfassung

Ein verbreitetes Problem beim Unterrichten von Software-Engineering ist, dass den Studierenden die Erfahrung mit großen, realen Softwaresystemen fehlt. Dadurch wird die Sinnhaftigkeit der vorgestellten Methoden nicht ausreichend erkannt. Wir versuchen, die fehlende Erfahrung zu simulieren und haben dazu die Methode des Just-in-Time Teaching (JiTT), in Form von Quiz, eingeführt. Die Studierenden, die an den Quiz teilgenommen haben, konnten die Methoden des Software-Engineerings am Ende des Semesters besser einordnen. Die Einführung dieser JiTT-Variante ist mit einem nicht zu unterschätzenden Aufwand verbunden, der sich aus unserer und der Sicht der Studierenden aber lohnt.

## Einleitung

An der Technischen Universität Hamburg-Harburg (TUHH) ist die Veranstaltung Software-Engineering eine Pflichtvorlesung in zwei Bachelorprogrammen und eine Wahlpflichtveranstaltung in vier weiteren Studienprogrammen. Die Vorlesung wird von ca. 70 Studierenden im zweiten oder vierten Semester besucht. Die Vorkenntnisse und Erfahrungen der meisten Studierenden beschränken sich im Bereich Softwareentwicklung auf das Bearbeiten kleiner Übungsaufgaben in zwei oder drei einführenden Programmierkursen (Prozedurale Programmierung, Funktionale Programmierung, Objektorientierte Programmierung). Den Studierenden fehlt deshalb der Einblick in große Softwareprojekte und die damit einhergehenden Herausforderungen, z.B. dass viele Entwickler an einem Projekt beteiligt sind, keine Person alle Einzelheiten des Gesamtsystems im Detail kennt und das System auch nach Jahren noch zu warten sein muss. Ohne diesen Einblick fällt es vielen Studierenden schwer, den Sinn der in der Vorlesung behandelten Konzepte und Methoden zu verstehen.

Ein Ansatz, den Studierenden die fehlende Erfahrung zu vermitteln, ist, die Vorlesung um ein

begleitendes Softwareprojekt zu ergänzen. Dies erfordert allerdings die Änderung des Curriculums für die betroffenen Studiengänge und war studiengangstechnisch nicht möglich. Wir haben stattdessen versucht, die fehlende Erfahrung zu simulieren, indem wir den Studierenden vor jeder Vorlesung Aufgaben gestellt haben, die die Notwendigkeit des behandelten Themas verdeutlichen und hierzu reale (Open-Source) Softwareprojekte verwenden. Die Methode, den Studierenden bereits vor der Vorlesung den Einstieg in das Thema im Selbststudium zu ermöglichen, wird häufig als „Just-in-Time Teaching“ bezeichnet.

In diesem Artikel beschreiben wir unseren Einsatz von JiTT im Bereich der Veranstaltung Software-Engineering, ein Konzept, das in Zusammenarbeit des Instituts für Softwaresysteme mit dem hochschul- und fachdidaktischen „Zentrum für Lehre und Lernen“ an der TUHH entwickelt wurde. Der Artikel ist wie folgt gegliedert: Abschnitt 2 führt JiTT allgemein ein und erläutert den Hintergrund dieser Lehrmethode. Abschnitt 3 beschreibt unsere konkrete Umsetzung im Rahmen der Software-Engineering Veranstaltung. Die dabei gemachten Erfahrungen werden in Abschnitt 4 besprochen, bevor der Artikel mit unseren Schlussfolgerungen in Abschnitt 5 endet.

## „Just-in-Time Teaching“

Just-in-Time Teaching ist eine Lehr-Lern-Strategie, deren Kern eine spezielle Kombination von Präsenzlehre und online-gestütztem Selbststudium ist (grundlegend zu JiTT: Novak u.a. 1999).

Das besondere Merkmal von JiTT ist es, dass den Studierenden stets noch vor Beginn einer Lehrveranstaltung Studienmaterial online zur Verfügung gestellt wird. Dies können Einstiegsfragen in ein neues Themengebiet sein, die bei der Einschätzung helfen, welche Vorstellungen und welches Wissen die Studierenden in diesem Bereich bereits haben. Es können aber auch anspruchsvollere Aufgaben sein, die bereits ein kor-

rektes Grundverständnis erfordern, aber weitergehende Schwierigkeiten aufzeigen. Ergänzend können für die Studierenden offene Fragen angefügt werden, in denen sie Unklarheiten oder Probleme formulieren, die sich bei der Bearbeitung ergeben haben.

Die Aufgaben – deren Umfang gering gehalten wird – bearbeiten die Studierenden vor Beginn der zugehörigen Präsenzveranstaltung. Ihre Antworten übermitteln sie elektronisch an die Lehrperson, idealerweise mittels einer Technik, die die studentischen Lösungen automatisch auswertet bzw. strukturiert. Lehrende haben dann die Möglichkeit, genau rechtzeitig („just-in-time“) einen Überblick über den Wissensstand ihrer Studierenden zu gewinnen und z.B. missverständliche intuitive Auffassungen eines Stoffes (Präkonzepte) zu erkennen. Sie können entsprechend in ihrer Veranstaltung bereits verstandenen Stoff verkürzt behandeln, dafür aber gezielt auf vorliegende Verständnisschwierigkeiten eingehen.

Wie genau dies geschieht, liegt frei in der Hand der Lehrenden. So können sie die aus den Einstiegsaufgaben gewonnenen Informationen lediglich zur Präzisierung ihrer eigenen Vortragsteile nutzen und z.B. kurzfristig in ihre Präsentationen einzelne Folien einbauen, die typische oder besonders interessante studentische Lösungen oder Probleme zeigen. Alternativ können die Informationen aus den Einstiegsaufgaben aber auch zum „roten Faden“ der zugehörigen Veranstaltung gemacht werden, indem beispielsweise gezielt an auffälligen Präkonzepten gearbeitet wird, strittige Fragen per Partnerdiskussion o.ä. in der Vorlesung weiter vertieft oder über Classroom-response-Systeme („Clicker“) nochmals für alle Studierenden zur Disposition gestellt werden (Luo 2008, 166).

Entscheidend ist in jedem Fall, dass die Lehrenden frühzeitig eine Rückmeldung zu den Lernbedürfnissen ihrer Studierenden erhalten bzw. die Studierenden eine Rückmeldung zu ihrem Lernstand. Auf diesem Weg soll die Stoffvermittlung effizienter werden, Interesse und Aktivität der Studierenden sollen steigen.

### **Didaktischer Hintergrund**

Betrachtet man JiTT in einem allgemeindidaktischen Kontext, so liegt sein Charme jenseits des Beitrags zu einer effizienten Lehrgestaltung zunächst in der Aktivierung der Studierenden zum selbstständigen Lernen – denn dass die studentische Eigenaktivität möglichst früh im Lehr-Lernprozess geweckt werden sollte, wird inzwischen gemeinhin als Voraussetzung für erfolgreiches Lernen aufgefasst (Novak u.a. 1999, 9).

Indem JiTT die Studierenden den Einstieg in

ein Thema selbst vollziehen lässt, nimmt es dabei auch die grundlegende Idee des didaktischen Konstruktivismus auf, dass jedes Individuum seine Wirklichkeit selbst erzeugt (Siebert 1997, 17). Für das Lernen bedeutet dies: Ein bestimmtes Verständnis lässt sich nicht standardisiert vermitteln, sondern Lernende müssen ihren eigenen Zugang zu einem Thema finden und zusätzliche Informationen in ihr eigenes Wissenskonstrukt (wiederum individuell) integrieren.

An den Hochschulen hatte dieser Ansatz die Proklamation eines „shift from teaching to learning“ zur Folge. Sein Ausdruck ist im Zuge des Bologna-Prozesses u.a. die Berechnung des studentischen Arbeitsaufwands (workload), die für einzelne Veranstaltungen unabhängig von der Präsenzzeit berechnet wird. Empirische Studien zu den neu konzipierten Studiengängen zeigen inzwischen, dass Selbststudium allerdings nur dann funktioniert, wenn es in ein didaktisches Gesamtkonzept eingebunden und in den Präsenzveranstaltungen wieder aufgenommen wird (Metzger 2011, 271 f.). An dieser Stelle setzt JiTT an, indem es die Ergebnisse des Selbststudiums zum Ausgangspunkt der Präsenzlehre macht.

Möglich ist es dabei auch, das Selbststudium durch die Einbindung in das Prüfungskonzept einer Veranstaltung zusätzlich anzureizen. So kann durch JiTT der Forderung nach „formativem“, d.h. das Lernen begleitendem Prüfen nachgekommen werden (Dubs 2006, 2 f.). Die Bearbeitung der Aufgaben kann anteilig auf die Note angerechnet werden, oder bei Bestehen der Abschlussprüfung wird ein Bonus für die Aufgabebearbeitung gewährt, der die Abschlussnote verbessert. Häufig wird es in diesem Fall von den Lehrenden als ausreichend betrachtet, wenn die Aufgaben überhaupt bearbeitet werden, unabhängig davon, ob die Lösungen richtig sind (Luo 2008, 167; Slunt/Giancarlo 2004, 986).

### **Verbreitung und Wirksamkeit von JiTT**

„Just-in-Time Teaching“ wurde von Physik-Lehrenden der Indiana University-Purdue University at Indianapolis (IUPUI), der United States Air Force Academy (USAFA) und des Davidson College entwickelt (Novak u.a. 1999, 7). Inzwischen hat es sich sowohl über fachliche als auch über geographische Grenzen hinweg weit verbreitet (Poth 2009, 7). Auch in Deutschland ist Just-in-Time Teaching angekommen, allerdings mit noch recht geringer Verbreitung (zu nennen sind insbesondere die TU Hamburg-Harburg und die niedersächsische Ostfalia-Hochschule, siehe aber auch Hagel et al. in diesem Tagungsband). Eingesetzt wird es in mittleren und auch großen Veranstaltungen, so z.B. an der TU Hamburg-Harburg in Vorlesungen mit über 600 Studierenden.

Der vielversprechende Ansatz von JiTT hat auch das Interesse an der Frage geweckt, inwieweit sich ein Erfolg der Methode wissenschaftlich belegen lässt. Hierzu liegt inzwischen eine ganze Reihe kürzerer Studien vor, die überwiegend von Lehrenden mit Datenmaterial aus ihrem eigenen Unterricht durchgeführt worden sind (das in Amerika verbreitete „scholarship of teaching and learning“, vgl. Huber 2011, 118).

Luo (2008) beispielsweise weist nach, dass die regelmäßige Bearbeitung von JiTT-Aufgaben einen deutlich positiven Einfluss auf die Abschlussnoten seiner Studierenden hatte. Slunt und Giancarlo (2004) wiederum vergleichen JiTT mit dem Erfolg sog. „Concept Checks“, d.h. strukturierter Feedbackfragen im laufenden Unterricht, die Vorwissen oder Verständnis von Studierenden prüfen. Gemessen anhand von durchschnittlichen Endnoten fällt das Ergebnis deutlich zugunsten von JiTT aus (Slunt/Giancarlo 2004, 986 f.).

Diese Nachweise der Effektivität von JiTT sind nicht durchgängig auf Zustimmung getroffen. So zweifelt etwa Poth (2009, 2) an der statistischen Belastbarkeit vieler der vorliegenden Studien und kommt in seiner eigenen Analyse von JiTT zu durchaus skeptischen Einschätzungen. Auch er verwirft die Methode aber nicht, sondern fordert genauere Studien, unter welchen Bedingungen sie funktionieren kann (Poth 2009, 230 f.; vgl. Kautz 2006). Dies scheint auch insofern wünschenswert, als dass JiTT bei Evaluationen von den Studierenden regelmäßig als sehr gut und hilfreich bewertet wird.

Im Folgenden soll nun erläutert werden, welche Erfahrungen bei der Anwendung von JiTT im Rahmen der Veranstaltung „Software-Engineering“ an der Technischen Universität Hamburg-Harburg gemacht worden sind.

## JiTT für Software-Engineering

Bei den bisherigen Umsetzungen von JiTT in unterschiedlichen Fächern ging es vorrangig darum, den Studierenden Grundwissen vor Beginn der Lehrveranstaltung zu vermitteln und den Lehrenden einen Überblick über den Wissenstand der Studierenden zu verschaffen. Bei unserem Einsatz von JiTT geht es hingegen in erster Linie darum, die Studierenden Erfahrungen mit der Komplexität von realen Softwareprojekten machen zu lassen. Diese Erfahrungen können nur gemacht werden, wenn die Studierenden auch an realen Softwaresystemen arbeiten. Wir haben uns daher entschieden, den Studierenden, im Rahmen eines Online-Quiz, Fragen zu Open-Source-Software zu stellen, deren Beantwortung sie an das Thema der nächsten Vorlesung heranzuführt und nebenbei den

Umfang des betrachteten Systems aufzeigt. Der Einsatz von JiTT besteht aus den folgenden Komponenten: einem wöchentlich wechselnden Softwaresystem, einem zweiteiligen Quiz und einer kurzen Einheit in der folgenden Vorlesung. Die in den jeweiligen Vorlesungen behandelten Themen und die dafür verwendeten Softwaresysteme sind in Tabelle 1 angegeben. Im Folgenden soll nun auf die Quiz im Detail eingegangen werden.

Vorlesungsthema	Softwaresystem
Software Prozesse	Firefox (Bugtracker)
Nichtfunktionale Anforderungen	LibreOffice (Release Notes)
Spezifikation	Aristotle Analysis System
Spezifikation nebenläufiger Systeme	Eclipse (Deadlock)
Design	STL Iteratoren
Architekturen	Diverse Systeme
Objektorientiertes Design	ADT Queue
Testen	Mozilla Test Case Writing Primer
Testüberdeckung	GCC libstdc++ Test Suite
Wartung	Diverse Programme

Tabelle 1: Vorlesungsthemen und verwendete Softwaresysteme

## Quiz

Jedes Quiz besteht aus zwei Teilen: der Einleitung, die die ganze Zeit verfügbar ist, und dem Fragenteil, für den es eine begrenzte Bearbeitungszeit gibt. In der Einleitung wird das betrachtete System kurz vorgestellt und es wird beschrieben, wo man das benötigte Material findet. Dabei kann es sich um einen bestimmten Eintrag im Bugtracker des Projekts handeln oder um ein Archiv mit dem Quelltext der Software. Darüber hinaus erhalten die Studierenden Hinweise, welche Stelle des Systems relevant ist. So ist sichergestellt, dass den Teilnehmenden beim Fragenteil, der in begrenzter Zeit zu bearbeiten ist, das benötigte Material zur Verfügung steht und sie sich schon etwas mit dem System vertraut machen können. Darüber hinaus werden die Studierenden motiviert, sich Gedanken zu den möglicherweise im Fragenteil gestellten Fragen zu machen und

sich ein wenig im System „umzusehen“.

Das Ziel ist, den Studierenden die Komplexität der Entwicklung realer Softwaresysteme zu verdeutlichen und sie anzuregen, sich eigene Gedanken zum Thema der nächsten Vorlesung zu machen. Dabei kommt es nicht darauf an, die gestellte Frage exakt zu beantworten. Oftmals ist das auch gar nicht möglich, da die Fragen bewusst so gestellt werden, dass es mehrere rationale Antworten gibt, die nur entsprechend begründet werden müssen. Eine solche Frage ist, zum Beispiel, ob bei Kenntnis eines bestimmten Fehlers im Softwaresystem die Veröffentlichung verschoben werden soll. Das Lernziel wird also schon durch das Bearbeiten der Aufgabe erreicht.

Die Aufgaben sind so gestellt, dass das Bearbeiten etwa 30 Minuten dauern sollte. Durch die zeitlich begrenzte Freischaltung der Bearbeitung wird verhindert, dass die Studierenden einen wesentlich höheren Aufwand betreiben.

### **Durchführung**

Es ist wichtig, dass die Studierenden den Zeitpunkt der Bearbeitung flexibel wählen können, um Konflikte mit anderen Veranstaltungen zu vermeiden. Da die Aufgaben von jedem Studierenden individuell zu bearbeiten sind, haben wir als Rahmen nur vorgegeben, dass das Quiz nach dem Ende der vorherigen Vorlesung veröffentlicht wird und bis zum Tag vor der nächsten Vorlesung bearbeitet sein muss. Das lässt dem Dozenten bzw. der Dozentin dann noch Zeit, die Antworten auszuwerten und in der Vorlesung entsprechend darauf zu reagieren.

Im Einzelnen sehen die Schritte wie folgt aus: Am Nachmittag nach der Vorlesung wird der erste Teil, die Einleitung, ins Netz gestellt. Sobald die Teilnehmenden ihre Vorbereitungen abgeschlossen haben und denken, dass sie sich im betroffenen Teilsystem ausreichend gut auskennen, starten sie mit dem zweiten, dem Fragenteil. In dem Moment, in dem sie auf den Start-Knopf klicken, werden ihnen die Fragen gezeigt und die Bearbeitungszeit beginnt zu laufen. Wir haben uns entschlossen, die Bearbeitungszeit auf 60 Minuten zu begrenzen, um zu verhindern, dass zu viel Zeit in die Beantwortung investiert wird. Die Fragen selbst sollten allerdings in unter 30 Minuten zu beantworten sein und wurden von den Studierenden in der Regel auch in dieser Zeit beantwortet.

Um die Studierenden zur Teilnahme anzuregen, haben wir die Quiz in ein Bonussystem integriert, das auch die (erfolgreiche) Teilnahme an den Übungen belohnt. Die durch alle Quiz insgesamt zu erreichenden Bonuspunkte entsprechen allerdings nur 2% der in der Klausur zu erreichenden Punkte. Der Anreiz ist also eher symboli-

scher Natur, hat aber trotzdem im Durchschnitt mehr als 70% der Vorlesungsteilnehmerinnen und Vorlesungsteilnehmer zum Mitmachen motiviert.

Wir halten die Bewertung einfach: Entweder die Antwort ist ausreichend für einen Teilnahmepunkt oder nicht. Entscheidend für die Bewertung ist hauptsächlich, ob die Antwort den Schluss zulässt, dass die Teilnehmerin oder der Teilnehmer sich mit der Fragestellung beschäftigt hat.

Nachdem die Studierenden die Fragen beantwortet haben, ist der letzte Schritt, dass die Fragen und insbesondere die Antworten in der nächsten Vorlesung aufgegriffen werden. Hierfür hat es sich als sinnvoll erwiesen, bei der internen Auswertung die Antworten in verschiedene Kategorien einzuteilen, z.B. „dafür“ und „dagegen“ oder „das ist immer ein Problem“, „das ist nur unter Voraussetzung A ein Problem“, „das ist nur unter Voraussetzung B ein Problem“ und „das ist nie ein Problem“. Für viele Studierende ist es interessant zu erfahren, wie viele ihrer Kommilitoninnen und Kommilitonen ihre Überlegungen teilen. Aber auch die Lehrenden können daraus ablesen, ob ein bestimmter Aspekt häufig genannt wird und ein anderer selten. Dann kann der selten genannte Aspekt in der Vorlesung nochmals besonders betont werden, falls er wichtig ist. Darüber hinaus werden aus jeder Kategorie besonders gute oder interessante, repräsentative Antworten gezeigt. Die Veröffentlichung erfolgt anonym, da das öffentliche Nennen des Namens von den betroffenen Studierenden oftmals als peinlich empfunden wird. Die Studierenden erkennen allerdings ihre eigene Antwort und erhalten dadurch eine extra Motivation.

### **Online-Plattform**

Zur Durchführung der Quiz verwenden wir die E-Learning Plattform ILIAS (<https://www.ilias.de>). Die Quiz werden dort als „Test“ eingestellt. Mit ILIAS ist es möglich, die maximale Bearbeitungszeit und den spätesten Abgabetermin automatisch zu setzen. Insbesondere die Bearbeitungszeit lässt sich nur mit einem Online-System genau ermitteln. Die Fragen in den Quiz sind überwiegend Freitextfragen, manchmal kombiniert mit einer Auswahlfrage, die das grobe Einteilen der Antworten in Kategorien automatisiert hat. Bei der oben schon erwähnten Frage nach einem Bug im Firefox-Webbrowser muss man sich beispielsweise zunächst in einer Auswahlfrage für oder gegen das Verschieben des Veröffentlichungstermins entscheiden, bevor man seine Wahl in einer Freitextfrage begründet. Da es bei Freitextfragen nicht möglich ist, die Antworten vom System automatisch bewerten zu lassen, muss die Bewertung vom Lehrenden beim Durchsehen der Antworten erfolgen. Das gleiche

Muster findet sich bei einer Frage zur Aufnahme eines neuen Testfalls in eine Testsuite (siehe Beispiel). In einem anderen Quiz soll ein Vorschlag für eine neue STL Iterator-Hierarchie begründet werden. Dieses Quiz besteht nur aus einer Freitextfrage zur möglichen Begründung.

### Beispiel

Nachdem wir nun den allgemeinen Aufbau der Quiz beschrieben haben, wollen wir ein Quiz genauer vorstellen. Das Quiz bereitet die Vorlesung zum Thema Testüberdeckung vor und als Softwaresystem kommt die C++-Standardbibliothek (libstdc++) der Gnu Compiler Collection (GCC) zum Einsatz.

Die Einleitung zum Quiz stellt GCC und libstdc++ kurz vor und leitet die Studierenden dann zur Testsuite des „find“-Algorithmus der libstdc++. Die dazugehörige Frage stellt dann einen neuen Testfall vor und die Studierenden müssen sich in einer Auswahlfrage entscheiden, ob der neue Testfall in die Testsuite aufgenommen werden soll oder nicht. In einer zweiten Frage muss diese Entscheidung dann begründet werden. Um die Entscheidung zu treffen und zu begründen, ist es notwendig, sich zu überlegen, was die Kriterien für die Aufnahme eines neuen Testfalls sind, was von den vorhandenen Testfällen abgedeckt wird und was der neue Testfall testet. Somit machen sich die Studierenden zum einen Gedanken über die Kriterien für eine gute Testsuite und zum anderen erleben sie durch das Navigieren zur Testsuite und das Herausfinden, was genau getestet wird, wie groß und komplex die C++-Standardbibliothek ist.

Die im Durchschnitt zur Beantwortung der Fragen benötigte Zeit betrug bei diesem Quiz 16 Minuten und 20 Sekunden. Der Aufwand zur Vorbereitung kann vom System nicht erfasst werden, wir schätzen jedoch, dass er unter 30 Minuten lag.

Am Tag vor der Vorlesung ist das Quiz beendet und die Antworten müssen ausgewertet werden. Durch die Auswahlfrage erhält man schnell einen Überblick, wie viele Studierende den neuen Testfall aufnehmen wollen und wie viele nicht. In diesem Fall war das Ergebnis unentschieden. Zur Auswertung der Begründung in der Freitextfrage muss jede Antwort einzeln betrachtet werden. Da die Antworten in der Regel recht kurz ausfallen, werden hierfür circa ein bis zwei Minuten pro Antwort benötigt. Die meisten Begründungen, den Testfall aufzunehmen, lassen sich in eine Kategorie „der Testfall testet etwas, das bisher noch nicht getestet wurde“ einteilen und die meisten Begründungen ihn abzulehnen in eine Kategorie „der Testfall erhöht die Testabdeckung nicht“. Einige Antworten fallen in die Kategorien „jeder

neue Testfall ist gut“ und „der Test wird immer fehlschlagen“. Bei der Kategorisierung der Antworten werden auch drei besonders gute oder interessante Antworten ausgewählt und zusammen mit dem Ergebnis der Auswertung in der Vorlesung präsentiert.

### Evaluation

Die Bewertung des Einsatzes von JiTT in der Veranstaltung Software-Engineering erfolgt aus zwei Perspektiven: zuerst aus Sicht der Studierenden und dann aus Sicht der Lehrenden. Grundlage sind die Daten aus dem Sommersemester 2012.

#### Studentische Evaluation

Die Quiz wurden von den Studierenden gut angenommen. Von insgesamt 70 Teilnehmenden an der Veranstaltung haben durchschnittlich 51 an den Quiz teilgenommen. Das entspricht einer Quote von 72%. Allerdings hat die Beteiligung zum Ende des Vorlesungszeitraums abgenommen, nachdem die Studierenden die maximal erzielbaren Bonuspunkte erreicht hatten. Die durchschnittliche Bearbeitungszeit des Fragenteils lag bei 17:25 Minuten. Dazu kam noch die Zeit für die Vorbereitung, also das Lesen der Einleitung, das Herunterladen des Quelltextes und das erste Umsehen im behandelten System, die nicht vom Onlinesystem erfasst werden kann. Insgesamt erscheint der zusätzliche Zeitaufwand für die Quiz aber vertretbar. Erfreulich ist weiterhin, dass im Schnitt 90% der Teilnehmenden eine Antwort gegeben haben, die zeigt, dass sie sich ernsthaft mit dem Thema auseinandergesetzt haben.

Vom Zentrum für Lehre und Lernen der TUHH (ZLL) wurde am Ende des Semesters ein Bewertungsbogen an die Studierenden verteilt, um ihre Einschätzung des JiTT-Ansatzes zu erfragen. Dabei sollten bestimmte Aspekte mit einer Note zwischen 1 (voll zutreffend) und 5 (nicht zutreffend) bewertet werden. Alle vorgegebenen positiven Äußerungen zu den Quiz wurden im Mittel als „weitgehend zutreffend“ bewertet. Erfreulich ist insbesondere die gute Bewertung der Aussagen, dass die Quiz die in der Vorlesung behandelten Konzepte mit realen Systemen verbinden und dass die Quiz geholfen haben, die Wichtigkeit des Vorlesungsinhalts zu verdeutlichen. Dies zeigt, dass das Ziel der Lehrinnovation weitgehend erreicht wurde. Das Ergebnis der studentischen Bewertung ist in Tabelle 2 dargestellt. Die Zeile unter der zu bewertenden Aussage gibt prozentual an wie oft die entsprechende Note von den Studierenden vergeben wurde.

#### Interne Evaluation

Betrachtet man die Einführung der Quiz aus Sicht

der Lehrenden, so fällt zunächst der große Aufwand zur Vorzubereitung der Quiz auf. Der größte Aufwand besteht darin, ein zum Inhalt der Vorlesung passendes Softwaresystem zu finden. Hat man das System gefunden, so ist noch eine Anleitung für die Studierenden zu schreiben, die sie zum betrachteten Teil des Systems führt. Zum Abschluss der Vorbereitung müssen noch Fragen zum System gestellt werden, die die Studierenden an das in der entsprechenden Vorlesung behandelte Thema heranführen. Dieser Aufwand fällt aber nur bei der Einführung an, da die gefundenen Systeme und Fragen in den folgenden Semestern wiederverwendet werden können.

Aussage					
1	2	3	4	5	k.A.
Der Aufwand für die Lehrinnovation lohnt sich.					
14%	38%	24%	14%	7%	3%
Die Innovation wurde erfolgreich umgesetzt.					
31%	34%	21%	7%	7%	0%
Insgesamt würde ich diesen Kurs weiterempfehlen.					
28%	45%	21%	3%	3%	0%
Die Quiz haben die in der Vorlesung präsentierten Konzepte mit realen Softwaresystemen in Verbindung gebracht.					
38%	48%	7%	7%	0%	0%
Die Quiz haben mir geholfen, die Bedeutung der Vorlesungsinhalte für reale Software-Engineering-Aufgaben zu verstehen.					
21%	45%	24%	3%	7%	0%
Die Auswahl der Softwaresysteme war angemessen.					
24%	55%	10%	7%	0%	0%
Die Aufgaben waren interessant und haben mich angeregt, mir das verfügbare Hintergrundmaterial anzusehen.					
21%	31%	31%	7%	10%	0%

Tabelle 2: Bewertung der Quiz durch die Studierenden

Nachdem die Quiz einmal vorbereitet wurden, müssen sie in die Online-Plattform eingepflegt werden. Gerade beim ersten Einsatz

eines Online-Systems sollten der Aufwand und eventuelle technische Probleme nicht unterschätzt werden. Vor jeder Vorlesung müssen die Antworten der Studierenden ausgewertet werden. Durch das Verwenden von Freitextfragen kann die Auswertung nicht automatisiert werden. Bei unserer Veranstaltung (mit durchschnittlich 51 Teilnehmern) hat die Auswertung etwa zwei Stunden in Anspruch genommen. Das Einbinden der Antworten in die Vorlesung ist hingegen kein großer Mehraufwand.

Die Schwierigkeiten liegen hauptsächlich bei der Auswahl der Softwaresysteme und Fragen. Es ist nicht leicht, zu einem bestimmten Thema oder Problem ein Softwaresystem zu finden, das das Problem veranschaulicht und gleichzeitig nicht zu komplex ist, um die Studierenden nicht zu überfordern. Das spiegelt sich auch in der unterschiedlichen Qualität unserer Quiz wieder.

Positiv für die Lehrperson in der Vorlesung ist, dass die Studierenden ein reges Interesse an den Antworten zum Quiz haben. Gerade bei längeren Vorlesungen, wenn die Konzentration der Studierenden nachlässt, hat das Besprechen der Fragen und Antworten einen positiven Effekt auf die Aufmerksamkeit. Des Weiteren war es manchmal möglich, verbreitete, aber falsche Vorstellungen der Studierenden zu erkennen und entsprechend darauf einzugehen.

## Schlussfolgerungen

Es war unser Ziel, die Studierenden Erfahrungen mit realen Softwareprojekten machen zu lassen, damit sie die Sinnhaftigkeit der in der Vorlesung vorgestellten Methoden besser verstehen. Dazu haben wir die Methode des JiTT, in Form von Quiz, eingesetzt. Die größte Herausforderung, bei unserem Einsatz von JiTT, ist das Finden von Softwaresystemen, an denen man die in der Vorlesung vorgestellten Konzepte verdeutlichen kann.

Eine systematische empirische Überprüfung, dass die Studierenden mit Hilfe der Methode gegenüber einer rein klassischen Vorlesung bessere Lernergebnisse erzielt haben, steht für den gegebenen Fall aus. Die Aussagen der Studierenden am Ende des Semesters, unsere eigenen Beobachtungen und der Leistungsvergleich mit früheren Jahrgängen legen für uns jedoch nahe, dass wir unser Ziel erreicht haben. Wir sind überzeugt, dass die Quiz das Verständnis nachhaltig verbessert haben und werden JiTT weiterhin in der beschriebenen Weise einsetzen.

## Literatur

Dubs, R. (2006): Besser schriftlich prüfen. Prüfungs-

- gen valide und zuverlässig durchführen. In: Neues Handbuch Hochschullehre, Einzelbeitrag H 5.1.
- Huber, L. (2011): Forschen über (eigenes) Lehren und studentisches Lernen – Scholarship of Teaching and Learning (SoTL). In: Das Hochschulwesen, vol. 59, Heft 4, S. 118-124.
- Kautz, C. (2006): Aktives Lernen in Großen Vorlesungen: Einsatz von internet-gestützten Vortests und interaktiven Vorlesungsfragen. In: Schlattmann, J. (Hrsg.): Die Bedeutung der Ingenieurpädagogik. Hamburg, S. 1-8.
- Luo, W. (2008): Just-in-Time-Teaching (JiTT) improves student's performance in classes – adaptation of JiTT in four geography courses. *Journal of Geoscience Education*, v. 56, n.2, p. 166-171.
- Metzger, C. (2011): Studentisches Selbststudium. In: Schulmeister, R.; Metzger, C. (Hrsg.): Die Workload im Bachelor. Zeitbudget und Studierverhalten – eine empirische Studie. Münster u.a., S. 237-276.
- Novak, G.; Patterson, E.; Gavrin, A.; Christian, W. (1999): Just-in-Time Teaching. Blending active learning with web technology. Upper Saddle River, NJ.
- Poth, T. (2009): Adressatengerechtes Unterrichten mit dem Just-in-Time Teaching-Verfahren. Diss., Würzburg. <http://kola.opus.hbz-nrw.de/volltexte/2009/416/pdf/DissertationPoth.pdf>. Zugriff am 23.10.2012.
- Riegler, P. (2012): Just in Time Teaching. Wer liest und wer lehrt an der Hochschule? In: Waldherr, F.; Walter, C. (Hrsg.): Wissen, können, verantwortlich handeln. Forum der Lehre 2012. Ansbach, S. 89-95.
- Siebert, H. (1997): Didaktisches Handeln in der Erwachsenenbildung. Didaktik aus konstruktivistischer Sicht. 2. Auflage, Neuwied u.a.
- Slunt, K., Giancarlo, L. (2004): Student-centered learning: a comparison of two different methods of instruction. *Journal of Chemical Education*, vol. 81, No. 7, p. 985-988.