

Von Häuptlingen und Indianern – Bachelor/Master als Chance

Karsten Weicker

Hochschule für Technik
Wirtschaft und Kultur Leipzig (FH)
Fachbereich Informatik, Mathematik und Naturwissenschaften
karsten@weicker.info

Nicole Weicker

Pädagogische Hochschule Heidelberg
Fakultät III, Institut für Datenverarbeitung/Informatik
nicole@weicker.info

Zusammenfassung

Angelehnt an Hierarchien in Projekten der realen Welt wird eine neue Organisationsform des Softwarepraktikums vorgestellt, in der Bachelor- und Masterstudenten gemeinsam Projekte bearbeiten. Das Ziel ist, Lernen im Projekt auf zwei Ebenen aufzuteilen. Der Ansatz wird anhand von Fragebögen analysiert und entsprechend bewertet.

1 Motivation

Die zentrale Bedeutung des Softwarepraktikums (Sopra) im Informatikstudium liegt in der Möglichkeit, die gelernten Inhalte der Softwaretechnik praktisch umzusetzen und dabei neben fachlichen und methodischen Kompetenzen durch die Teamarbeit auch eine Reihe von Sozial- und Selbstkompetenzen zu erwerben. Letzteres wird durch die Akkrediteure der neuen Bachelor- und Masterstudiengängen immer stärker gefordert [ASIIN 2005, ASIIN 2006].

Dabei kommt der Durchführung und der Organisation des Softwarepraktikums besonders viel Bedeutung zu. Die Arbeit von [Fincher et al. 2001] liefert einen umfassenden Überblick, auf wie viele unterschiedliche Arten Projektarbeit im Informatikstudium integriert werden kann. Insgesamt kann ein Trend von

eher kleinen Gruppen [Schlimmer et. al. 1994] zu größeren Gruppen mit bis zu 12 Studenten beobachtet werden [Ludewig/Reißing 1998]. So wurde auch an der HTWK Leipzig bis 2004 das Softwarepraktikum mit individuellen Projekten von Gruppen mit 2-4 Studierenden durchgeführt. Im Zuge der anstehenden Akkreditierung der Studiengänge wurde danach die Teamgröße angehoben, und die individuellen Projekte wurden durch ein für alle Gruppen identisches Thema ersetzt. Allerdings sind im Bachelor/Master-System zeitintensive Konzepte wie zwei einjährige, überlappende Studienprojekte [Ludewig/Reißing 1998] nicht umsetzbar.

Im Weiteren wollen wir ein Konzept für die Durchführung von Softwarepraktika skizzieren, wobei die folgenden Anforderungen die Rahmenbedingungen vorgeben: Die Dauer beträgt ein Semester, und die Teams sollen 8-12 Personen umfassen. Ferner muss das Projekt »realitätsnah« sein, d.h., ein echter Kunde (sprich: extern oder ein Hochschullehrer, in keinem Fall ein Student) definiert ein komplexes und evtl. risikoreiches Produkt, und innerhalb des Teams sollen projektübliche Hierarchien abgebildet werden. Insbesondere legen wir Wert auf die üblichen Teamprozesse in einem Projekt mit großem Zeitdruck sowie auf ein individuelles Projektmanagement.

In einer Befragung wurde an der HTWK Leipzig untersucht, welche der im Sopra angestrebten fachlichen, methodischen, persönlichen und sozialen Kompetenzen die Studierenden im Rahmen der Lehrveranstaltung gelernt haben.

2 Organisation des Softwarepraktikums und seine Probleme

Für die Organisation des Softwarepraktikums steht eine ganze Bandbreite möglicher Betreuungsmodelle zur Verfügung. Diese reichen von der verbreiteten, weil mit wenig bis keinem Aufwand realisierbaren, Laissez-faire-Betreuung bis zu Formen, in denen das Projektmanagement den Studenten quasi durch wöchentliche, von allen zu bearbeitende Aufgaben abgenommen wird. Letzteres verfehlt dabei häufig viele der Lernziele, die wir mit einem Softwarepraktikum verbinden, da die erwünschten fachlichen Auseinandersetzungen und gruppeninternen Konflikte unterbleiben.

Aber auch die Laissez-faire-Betreuung ist kritisch zu hinterfragen, da bereits in [Weicker et al. 2006] argumentiert wurde, dass unbetreute Kompetenzvermittlung nur in seltenen Fällen gelingt. Das deckt sich insbesondere auch mit unseren Erfahrungen mit Softwareprojekten, in denen die Studenten die Projektleitung selbst organisiert haben. Die Wahl eines Projektleiters existiert meist nur auf dem Papier, da die gewählte Person häufig weder das notwendige Charisma besitzt noch über die technischen und organisatorischen Fähigkeiten bzw. Erfahrungen verfügt. Und selbst wenn diese Voraussetzungen gegeben sind, wird der Leiter nicht als Primus inter Pares akzeptiert, sondern vom Team immer wieder in seiner Autorität in Frage gestellt. So läuft es in der Regel auf das reine Konsensprinzip ohne echte Führung hinaus.

Daher sind all die Zwischenformen in der Betreuung und Organisation interessant, die sowohl Teamdynamik und reale Projektsituationen als auch eine gewisse Betreuung miteinander vereinbaren. Ein Hilfsmittel ist der Einsatz von Tutoren, die über ein gewisses Maß an Erfahrung verfügen, da sie die Veranstaltung bereits erfolgreich absolviert haben. [Stoyan/Glinz 2005] und [Lindig/Zeller 2005] setzen studentische Hilfskräfte als »Coachs« ein, die den Prozess begleiten und den Teams mit Ratschlägen zur Seite stehen. [Demuth et al. 2002] gehen noch einen Schritt weiter, da dort der Tutor zum größten Teil die Projektleitung übernimmt. Der Tutoreneinsatz ist in jedem Fall positiv zu werten. Allerdings bringt dies auch einen gewissen Unsicherheitsfaktor mit sich, da Tutoren die Aufgabe mit unterschiedlich starker Motivation angehen – dies reicht von einem reinen »Job« bis zu Tutoren, die auch als Coach das Team bevormunden. (Dies haben wir konkret im geringfügig anderen Kontext eines projektbasierten, fortgeschrittenen Programmierkurses im zweiten Semester beobachtet.) Was dabei fehlt, ist Verbindlichkeit in vielerlei Hinsicht.

Zur Motivationssteigerung haben [Stoyan/Glinz 2005] damit gearbeitet, dass die Tutortätigkeit mit ECTS-Punkten belohnt wird. Dies ging uns bei der Neukonzeption des Softwarepraktikums noch nicht weit genug, sondern wir wollten neben einer verbindlichen Aufgabenteilung auch eine echte Benotung der betreuenden Studenten. Unser Ansatz wird im nächsten Abschnitt beschrieben und ist angelehnt an ein Szenario der University of Leeds [Fincher et al. 2001], bei dem Studenten im dritten Studienjahr vierköpfige Teams aus dem ersten Studienjahr anleiten.

3 Softwarepraktikum an der HTWK

Dem politischen Willen Rechnung tragend, dass der Bachelor-Abschluss berufsqualifizierend sein soll, wurden an der HTWK (wie an vielen anderen Hochschulen auch [BMBF 2005]) die Kapazitäten entsprechend gestuft: Auf 80 Informatik-Bachelors kommen 20 Informatik-Masters. Dazu kommen an der HTWK noch 50 Medieninformatik-Bachelors. Ausgehend von diesen Zahlen wurde eine gemeinsame Veranstaltung konzipiert, in der die Masterstudenten die Projektleitung in den Softwarepraktika übernehmen. Damit haben wir Teams bestehend aus ausschließlich gleichberechtigten Studenten ersetzt durch viele Indianer mit wenigen echten Häuptlingen. Im »worst case« leiten zwei Master in ihrem zweiten Master-Semester (Projektleitung und Qualitätssicherung) ein Team mit 13 Bachelors. Studienabbrecher berücksichtigend sind es häufig Teams mit neun Bachelorstudenten. In den ersten drei Jahren Übergangszeit vom Diplom zum gestuften System variiert die Zahl der Masterstudenten noch stark, weshalb hier flexibler organisiert wird. Die Vorbildung der Masterstudenten umfasst während des Bachelorstudiums zwei Softwaretechnikvorlesungen, das Softwarepraktikum als Indianer und ein 3-monatiges Praktikum in einem Betrieb; begleitend zur Pro-

jektleitertätigkeit gibt es einen wiederholenden Crashkurs in Projektmanagement sowie 14-tägige Supervisionssitzungen mit allen Masterstudenten.

Konkret investiert jeder Bachelor 240 Stunden (8 ECTS-Punkte) und jeder Masterstudent 120 Stunden (4 ECTS-Punkte). Die Teams werden vom Betreuer zufällig zusammengestellt, wobei darauf geachtet wird, dass jedes Team ähnliche Voraussetzungen hat und die Interessengebiete der Studenten alle Tätigkeiten abdecken. Die Aufgabe wird vom Kunden vorgegeben – in den letzten beiden Jahren waren dies Webanwendungen. Diese Aufgabe wird im Sommersemester in 16 Wochen konkurrierend bearbeitet. Jeder Bachelorstudent übernimmt eine Rolle im Team (Analytiker, Architekt, Chefprogrammierer, Technologie, Tester, Integrator, PR, GUI-Spezialist, Weiterbildungsbeauftragter, Handbuch etc.), was bei uns bedeutet, dass er für diesen Aufgabenbereich der Projektleitung zuarbeitet (Arbeitspakete formulieren etc.). Die Rolle entspricht damit nicht dem Arbeitsgebiet des Studenten – so muss z.B. jeder einen Anteil an der Programmierung beitragen. Das bedeutet auch, dass wichtige Aufgaben wie die Architektur – analog zu realen Projekten – nicht ein Aufgabenpaket eines Bachelors, sondern mit Chef-sache sind [Rainwater 2002]. Der Ablauf orientiert sich am Unified Process, und die vorgegebenen Meilensteine sind Anforderungsspezifikation, GUI-Prototyp, Entwurf, Unit-Test der Kernfunktionalität, 80 %-Systemtest und Abschlusspräsentation. Die Betreuung ist eher schlank und beschränkt sich auf Meilensteinabnahme, per Gutschein limitierte Kundengespräche [Schneider 2005], öffentliche Reviewsitzungen und 14-tägige Statusitzungen mit allen Masterstudenten. Neben der erstellten Software samt der dazugehörigen Dokumente wurden vom Team wöchentliche Projektberichte mit Zeitabrechnung abgegeben.

Die Benotung setzt sich bei den Bachelorstudenten zu gleichen Teilen zusammen aus der Bewertung der Dokumente (v.a. bzgl. Konsistenz) und Vorträge, der Produktbewertung durch den Kunden, einer Note für das persönliche Engagement (Arbeitsstunden, Arbeitspakete und einer Peer-Punktebewertung), einer Reflexion über das eigene Lernen und einer Zusammenfassung der Vorgänge im eigenen Rollenbereich. Bei den Masterstudenten gehen Dokumente, Produkt und Engagement zu je 20 %, der Projekt-/Qualitätsplan zu 30 % und eine Sesam-Simulation [Schneider 1994] zu 10 % ein.

Aus didaktischer Sicht sollen Studenten im ersten Durchgang als Bachelorstudent eigene Projekterfahrungen sammeln, sich in ihren Kompetenzen realistisch einschätzen lernen und durch Beobachten und Interaktion mit einem übergeordneten Projektleiter auch eine erste, eher passive Einschätzung des Projektmanagements entwickeln. Im zweiten Durchgang als Masterstudenten hingegen können sie auf diesen Erfahrungen aufbauen und durch aktive Betätigung Projektkompetenzen auf einer höheren Ebene erlangen. Dies entspricht dem klassischen didaktischen Spiralkonzept.

4 Bewertung der Lehrveranstaltung

Zur Beantwortung des Konzepts der Lehrveranstaltung wurden Fragebögen am Anfang und am Ende des Semesters ausgegeben. Durch einen individuellen Schlüssel können jeweils die beiden Fragebögen eines Studenten einander zugeordnet werden.

Konkret wurden die Befragungen in den Sommersemestern 2007 und 2008 durchgeführt. Von 2007 liegen 36 Bachelor- und 12 Master-Fragebögen vor. 2008 sind dies 50 bzw. 10 Fragebögen. Am Anfang wurden die Studenten gebeten, die Vorkompetenz bzgl. verschiedener Sozial-/Selbst- sowie fachlich-methodischer Kompetenzen auf einer Skala von 1 (sehr) bis 4 (kaum) einzuschätzen. Am Ende mussten sie nochmals die Vorkompetenz vor dem Projekt sowie den Lernerfolg beurteilen. Ferner wurden verschiedene Fragen zum Verlauf des Projekts gestellt.

Die Ergebnisse der beiden Vorher-Nachher-Befragungen sind vollständig in der am Ende stehenden Tabelle 3 enthalten. Im Weiteren werden die Ergebnisse dieser Befragungen vorgestellt und exemplarisch auch mit den Ergebnissen einer ähnlichen Befragung des Sopras 2006 ohne Masterprojektleiter verglichen.

4.1 Veränderte Wahrnehmung der Vorkompetenz

Zunächst interessiert uns an dieser Stelle, ob sich die Selbsteinschätzung der Vorkompetenz durch die Erfahrung im Softwarepraktikum ändert. Dies ist ein Indikator dafür, bei welchen Kompetenzen die Studierenden sich zunächst überschätzt haben bzw. welche Aspekte durch das Softwarepraktikum besonders gut gefordert werden, so dass die Vorkenntnisse über gemachte Erfahrungen verifiziert werden können. Die Umfrageergebnisse sind in Tabelle 3 dargestellt – Werte, auf die sich der folgende Text bezieht, sind grau hinterlegt.

Wenn man die beiden Bachelor-Jahrgänge gemeinsam auswertet, dann gibt es mit <5 % Fehlerwahrscheinlichkeit eine negativere Neueinschätzung bei Kostenschätzung, Schnittstellengestaltung, Begeisterungsfähigkeit, dem Zuhörenkönnen, Kommunikation mit Fachfremden und der Visualisierungskompetenz. Sogar mit <1 % Fehlerwahrscheinlichkeit gilt dies für die Oberflächengestaltung, Kundengespräche, andere Programmiersprachen als Java, Programmierrichtlinien, Unit-Tests, Abstraktionsvermögen, sichere Programmierkenntnisse, die Bereitschaft, Werkzeuge zu nutzen, selbstständiges Arbeiten, Neugier und die Bereitschaft, sich in Neues einzuarbeiten.

Bei den Masterstudenten liegen die Informationen nur von den zehn Studenten der 2008-Befragung vor. Mit <5 % Fehlerwahrscheinlichkeit erweisen sich hier Projektmanagement, Programmierrichtlinien und Einfühlungsvermögen als Kompetenzen mit realistischerer Korrektur der Selbsteinschätzung. Mit <10 % Fehler kann man noch die Fähigkeiten, selbstständig zu arbeiten und sich mitteilen zu können, hinzuzählen.

Deutlich erkennt man einen anderen Schwerpunkt bei den Masterstudenten. Interessant ist allerdings, dass alle Studenten ihre Fähigkeit, selbstständig arbeiten zu können, vor dem Projekt besser eingeschätzt haben, als sie dies rückblickend nach dem Projekt machen. Zusammen mit der guten Lerneinschätzung verdeutlichen die Ergebnisse, dass gerade die Teamarbeit auch das organisierte, zielstrebige Arbeiten des Einzelnen besonders gut schult.

4.2 Vergleich der Lernschwerpunkte

Beide Jahrgänge berücksichtigend wurden von den Bachelors die Oberflächengestaltung, Java, die Bereitschaft, Werkzeuge zu benutzen, die Fähigkeit selbstständig zu arbeiten, die Bereitschaft, sich in Neues einzuarbeiten, Hilfsbereitschaft und Verantwortungsbewusstsein besonders gut gelernt. (Diese Werte sind in Tabelle 3 ebenfalls grau markiert.)

Bei den Masterstudenten wurden besonders gute Werte bei den folgenden Kompetenzen erzielt: Projektmanagement, das Zuhörenkönnen, die Fähigkeit, sich mitteilen zu können, Hilfsbereitschaft, Durchsetzungsvermögen, die Fähigkeit zu leiten, Entscheidungsfähigkeit, Verantwortungsbewusstsein und die Fähigkeit zu deligieren.

Interessant ist, dass es sich auf den beiden Ebenen tatsächlich um unterschiedliche Lernschwerpunkte handelt – insbesondere sind dies bei den Masterstudenten Kompetenzen, die für Projektleiter wichtig sind.

Vergleicht man die Ergebnisse der Bachelors mit den Umfrageergebnissen von 44 Studenten einer ähnlichen Befragung 2006 bezüglich eines Softwarepraktikums ohne Masterstudenten, erkennt man, dass die meisten der Kompetenzen ähnlich gelernt wurden. Signifikante Unterschiede (Fehlerrate <11 %) sind lediglich bei der UML-Modellierung (2006: Mittelwert 2,62/Standardwert 0,85), Schichtenarchitektur (2006: 2,38/0,85) und Versionskontrolle (2006: 2,5/0,90) erkennbar, die in den neuen Veranstaltungen besser gelernt wurden. Lediglich die Kommunikation mit Fachfremden wurde mit <5 % Fehler schlechter gelernt. Bezüglich der Projektmanagementkompetenzen konnte kein signifikanter Unterschied zwischen 2006 und den Bachelors 2007/2008 festgestellt werden. Dies zeigt, dass die Einbeziehung von Masterstudenten das Lernen der Bachelors nicht verschlechtert – bei der Anwendung von relevantem fachlichen Wissen können sie sogar nachweislich davon profitieren.

Und zuletzt noch ein kurzer Blick auf die Unterschiede zwischen den Praktika 2007 und 2008 (in Tab. 3 fett dargestellt). Im jüngeren Projekt waren erstmals auch Medieninformatiker beteiligt, die Kunden waren externe Auftraggeber, und die Vorgaben für Präsentationen, Meilensteine etc. waren strenger. Vermutlich der letzte Punkt hat zu signifikant (<10 % Fehler) besserem Lernen von Unit-Tests, Visualisierung und Präsentation bei den Bachelors geführt. Ein schlechterer Lerneffekt beim sicheren Programmieren kann ebenfalls wie gewisse Turbulenzen

bei den Programmiersprachen darauf zurückgeführt werden, dass 2008 einige Teams mit Grails statt Java gearbeitet haben. Warum von den Bachelors Projektmanagement 2007 besser gelernt wurde, ist unklar. Bei den Masterstudenten wurden 2008 Präsentation, Kommunikation mit Fachfremden, Flexibilität und Begeisterungsfähigkeit signifikant besser gelernt.

4.3 Wichtigste Probleme im Projekt

Grundsätzlich ist das Softwarepraktikum so angelegt, dass die Aufgabe ein wenig zu komplex und zu umfangreich für die Bearbeitung in 15–16 Wochen ist. Dies soll Probleme und Konflikte im Team forcieren, da wir dadurch den Lerneffekt höher einschätzen. Tabelle 1 zeigt nun die prozentualen Antworten der Studenten auf die Frage, wo sie die größten Probleme bei der Arbeit am Projekt sahen.

Die meisten Nennungen hat der Punkt des fehlenden technischen Know-how erhalten. Dies ist einerseits verständlich, andererseits wurden die Studenten schon beim Start der Projekte dazu angehalten, sich durch den Cheftechnologen damit auseinanderzusetzen. Dies fand in vielen Projekten nicht im gewünschten Maß statt. Zusätzlich wurde allerdings beklagt, dass in der Anfangszeit nicht alle Studenten ausgelastet sind. Die zweithäufigste Nennung sind Kommunikationsprobleme: 2007 innerhalb des Teams, was 2008 durch die Probleme mit externen Kunden überlagert wurde.

Hauptprobleme	Befragung SS07		Befragung SS08	
	Bachelor	Master	Bachelor	Master
Zu komplexe Aufgabe	33%	33%	14%	40%
Kommunikation mit dem Kunden	3%	8%	70%	60%
Kommunikation im Team	50%	67%	30%	20%
Schnittstellenprobleme	17%	33%	36%	40%
fehlendes technisches Know-how	75%	50%	50%	60%
fehlende Motivation im Team	33%	42%	22%	20%
mangelhaftes Projektmanagement	3%	33%	20%	20%

Tab. 1 Auswertung des Fragebogens bezüglich der größten Probleme in den einzelnen Projekten. Gezeigt wird die prozentuale Nennung durch die Teilnehmer.

Bei den Bachelorstudenten folgt mit Abstand die fehlende Motivation der Teammitglieder, während die Kritik am Projektmanagement den letzten Platz einnimmt. Dies unterstreicht an dieser Stelle, dass die Projektleitung durch Masterstudenten bei den Bachelors sehr gut angenommen wird.

Bei den Masterstudenten sind die weiteren Punkte nahezu gleich gewichtet. Auffallend ist, dass die Komplexität der Aufgabe und Schnittstellenprobleme von

Mastern eher wahrgenommen werden. Und auch die selbstkritische Sicht der eigenen Leitungstätigkeit ist interessant.

4.4 Rollenverhalten der Masterstudenten

Um festzustellen, wie die Masterstudenten ihr Selbstverständnis als Projektleiter definieren, ließen wir diese Studenten ihren Führungsstil anhand von Eigenschaften charakterisieren. Diese sind den drei Prototypen »freundlicher Kollege«, »Choleriker« und »formaler Prozessler« aus [Vigerschow/Schneider 2007] entnommen. Tabelle 2 zeigt die Ergebnisse.

Auch hier sieht man, dass der Führungsstil sehr stark durch den freundlichen Kollegen bestimmt ist – eine Herangehensweise, die auch dem beobachteten Verhalten bei den früher durchgeführten selbstorganisierenden Softwarepraktika entspricht. Aber die Nennungen zu den anderen beiden Prototypen zeigen, dass darüber hinaus ein strengeres professionelles Leitungsverhalten angestrebt wird. Insbesondere eine genauere Auswertung der Arbeitszeiten und der Verteilung der Arbeitspakete zeigt, dass die Arbeit im Team großteils gleich verteilt wird – mit wenigen Ausreißern bei der Arbeitszeit nach oben und unten.

	SS07	SS08		SS07	SS08
Der freundliche Kollege			selbst aktiv mitmachen	17%	30%
verständnis	42%	70%	appellierend	25%	20%
geduldig	75%	70%	»alle im selben Boot«	33%	60%
Der Choleriker					
fordernd	17%	0%	zielorientiert	83%	40%
bestimmt	0%	40%	»Nichts ist unmöglich«	25%	50%
Der formale Prozessler			bewährte Vorgehensw.	8%	20%
klare Vorgaben	17%	30%	Konflikte versachlichen	33%	30%
rational	58%	30%	»Erfolg durch Disziplin«	17%	10%

Tab. 2 Auswertung des Fragebogens bezüglich des Selbstverständnisses der Projektleiter. Angaben sind in Prozent.

Eine genauere Analyse zeigt, dass neun der 22 Masterstudenten laut Selbsteinschätzung überwiegend als freundlicher Kollege agiert haben. Jeweils einer fällt klar in die beiden anderen Prototypen. Die anderen 11 Studenten stellen Mischungen dar, wobei einer als ausschließliche Mischung des Cholerikers mit dem formalen Prozessler auffällt. Die Zuordnung der Eigenschaften zu den Prototypen war – wie auch die Prototypen selbst – den Studenten nicht bekannt.

5 Diskussion

Die hier beschriebene Veranstaltung ist so konzipiert, dass übliche Hierarchien aus der Arbeitswelt durch das Softwarepraktikum abgebildet werden. Die Ergebnisse zeigen die Akzeptanz der Struktur durch die Studenten. Die Häuptlinge werden tatsächlich als solche gesehen. Aus Gesprächen mit den Bachelors geht eine klare Erwartungshaltung an die Projektleiter hervor. In einem Fall war eine Mediationssitzung notwendig, weil der Master-Projektleiter dieser Erwartungshaltung nicht gerecht wurde. In den meisten Fällen haben die Projektleiter ihre Aufgabe ernst genommen und sich um eine professionelle Haltung bemüht. Da sie für dieses Projekt ebenfalls eine Note erhielten, haben wir subjektiv Einsatzbereitschaft und Verantwortungsgefühl erlebt, das weit über unsere Erfahrungen mit Tutoren in der Projektleitung hinausgeht.

Aus organisatorischer Sicht ließ sich die Veranstaltung unkompliziert durchführen. In der Übergangszeit der vergangenen zwei Jahre musste flexibel mit der Ressource der Projektleiter umgegangen werden. So gab es 2007 Projekte, in denen einzelne Masterstudenten sowohl Projektleitung als auch Qualitätssicherung übernahmen. 2008 wurde ein Qualitätssicherer immer zwei Projekten zugeordnet.

Es hat sich klar gezeigt, dass innerhalb einer Veranstaltung Häuptlinge und Indianer auf unterschiedlichen Ebenen lernen. Im Vergleich zu den gleichberechtigten Teams aus dem Jahr 2006 wurden keine substanziellen negativen Auswirkungen ersichtlich. Im Gegenteil profitieren die Bachelors fachlich von den Master.

Neben diesem Lernen auf unterschiedlichen Stufen, bei dem sich sowohl Bachelors als auch Masters auf ihre Ebene konzentrieren können und davon gegenseitig profitieren, sehen wir die Chance der vorgestellten Bachelor/Master-Kombination vor allem in der Tatsache, dass Studenten auf dem Weg zum Masterabschluss zwei Projekte in unterschiedlicher Funktion durchlaufen und ihr Agieren als Häuptling auf dem eigenen Erleben als Indianer aufbauen können.

Es sind viele Varianten des vorgestellten Konzepts denkbar. Insbesondere bei der Aufgabenverteilung zwischen den Masterstudenten eines Teams sehen wir noch Verbesserungspotenzial und werden im kommenden Semester mit einem Master für die fachliche Planung (Arbeitspakete) und einem für die Personalplanung arbeiten.

Kompetenz	Bachelorstudenten			Masterstudenten		
	Start Vorkmp	Ende Vorkmp	Ende Lerneff.	Start Vorkmp	Ende Vorkmp	Ende Lerneff.
Projektmanagement	3,31/0,82 2,94/0,77	3,44/0,65 3,22/0,74	2,08/0,65 2,40/0,76	2,50/0,85	3,25/0,97 3,40/0,70	2,08/0,90 1,80/0,63
Risiko-minimierung	3,53/0,70 3,12/0,66	3,50/0,56 3,21/0,73	2,75/0,87 2,94/0,76	3,40/0,52	3,67/0,65 3,20/0,91	2,75/0,87 2,78/0,44
Kostenschätzung	3,41/0,60 2,76/0,82	3,36/0,59 3,15/0,82	3,11/0,89 2,98/0,87	3,40/0,70	3,41/1,00 3,70/0,48	3,08/0,90 3,11/0,93
Prozessmodell	3,05/0,68 2,98/0,74	3,18/0,58 3,00/0,67	2,77/0,60 2,72/0,80	2,80/0,63	2,67/0,78 3,10/0,73	2,92/0,79 2,44/0,53
UML-Modellierung	2,61/0,73 2,36/0,75	2,72/0,70 2,5/0,66	2,33/0,89 2,31/0,83	2,50/0,53	1,92/0,79 2,40/1,07	2,75/0,87 2,67/0,87
GUI-Design	2,22/0,83 2,20/0,70	2,81/0,79 2,77/0,77	1,86/0,87 2,14/0,86	2,30/0,82	2,17/1,03 2,60/0,97	2,75/0,87 2,78/1,09
Kunden-gespräche	3,17/0,97 2,80/0,95	3,66/0,54 3,15/0,82	3,00/0,91 3,04/0,99	2,80/0,63	2,58/1,16 3,10/0,74	3,00/1,04 3,00/1,00
Schnittst.gest.	3,08/0,91 2,81/0,81	3,36/0,68 3,17/0,64	2,33/0,83 2,23/0,76	2,30/1,16	2,17/0,83 2,50/0,97	3,08/0,90 2,78/0,97
Schichtenar.	3,31/0,71 3,02/0,78	3,22/0,68 3,04/0,76	2,08/0,69 2,14/0,86	2,50/1,18	1,83/0,72 2,40/0,70	3,00/0,95 2,67/0,87
Entwurfsmuster	3,22/0,87 2,96/0,73	3,40/0,60 3,07/0,77	2,31/0,68 2,52/0,82	2,10/0,88	2,25/0,75 2,20/0,79	3,17/0,83 2,80/0,92
Java	3,11/0,75 2,39/0,86	3,22/0,80 2,66/0,79	1,58/0,73 2,04/0,73	2,10/0,99	1,83/0,83 2,30/0,95	3,00/1,04 2,89/1,17
andere Prog.sp.	2,27/0,74 2,24/0,82	2,69/0,82 2,87/0,82	2,92/0,87 2,29/0,87	1,50/0,71	1,92/0,51 1,90/0,74	3,17/1,03 2,89/1,17
Prog.richtlinien	2,86/0,76 2,55/0,84	3,11/0,75 2,98/0,74	2,31/0,75 2,39/0,73	1,80/0,42	1,92/0,67 2,70/0,67	2,75/1,14 2,78/1,09
Debugging	3,03/0,77 2,90/0,86	3,09/0,85 3,06/0,84	2,34/0,80 2,59/0,84	2,00/1,05	2,09/0,83 2,40/0,70	2,45/1,21 2,89/1,05
Unit-Tests	3,31/0,71 2,92/0,78	3,64/0,54 2,66/0,79	3,03/0,84 2,45/0,96	3,20/0,63	3,42/0,67 3,20/0,63	2,83/0,94 2,30/1,06
Versionskontr.	3,42/0,81 3,12/0,80	3,56/0,73 2,87/0,82	2,06/0,95 2,33/0,94	2,30/0,95	2,08/0,792, 60/0,70	2,83/0,94 2,40/1,07
Abstraktionsver.	2,53/0,70 2,16/0,79	2,83/0,77 2,55/0,65	2,44/0,69 2,29/0,65	1,60/0,52	1,92/0,67 1,67/0,71	2,92/0,79 2,50/0,85
Formalisierung	2,97/0,65 2,78/0,82	3,03/0,65 2,85/0,59	2,67/0,63 2,71/0,74	2,20/0,63	2,00/0,74 2,00/0,50	2,83/0,93 2,60/0,97
Form. Meth. anwenden	2,67/0,682 ,64/0,75	2,81/0,67 2,81/0,61	2,47/0,88 2,67/0,66	2,20/0,42	2,17/0,83 2,22/0,44	2,83/0,83 2,80/0,79

Kompetenz	Bachelorstudenten			Masterstudenten		
	Start Vorkmp	Ende Vorkmp	Ende Lerneff.	Start Vorkmp	Ende Vorkmp	Ende Lerneff.
Exaktheit	2,60/0,74 2,62/0,67	2,81/0,79 2,79/0,66	2,47/0,81 2,51/0,71	2,20/0,63	2,17/0,83 2,11/0,78	2,58/0,90 2,40/0,97
sichere Progr.	2,50/0,88 2,38/0,88	2,83/0,74 2,87/0,80	1,86/0,68 2,24/0,75	1,80/0,79	1,75/0,75 1,78/0,67	2,75/1,06 2,80/1,23
Werkzeuge nutzen	2,06/0,67 1,88/0,72	2,75/0,73 2,55/0,85	2,00/0,99 1,90/0,77	1,70/0,95	1,75/0,62 1,89/0,93	2,17/0,83 2,30/1,05
Problemlösen	2,58/0,65 2,10/0,74	2,40/0,74 2,57/0,71	1,97/0,70 2,24/0,72	1,90/0,88	2,41/0,67 1,78/0,67	2,50/0,80 2,10/0,74
Selbstständig arbeiten	1,94/0,79 1,70/0,54	2,08/0,73 2,24/0,77	1,92/0,87 2,02/0,70	1,40/0,84	1,50/0,67 2,22/0,67	2,83/1,19 3,00/0,94
Zeitmanagement	3,00/0,68 2,60/0,83	2,92/0,60 2,81/0,77	2,17/0,77 2,27/0,76	2,70/0,95	2,33/0,98 3,00/0,50	2,41/0,67 2,20/1,14
Geduld	2,36/0,932 ,24/0,82	2,19/0,58 2,34/0,94	2,19/0,98 2,08/0,81	1,60/0,97	2,58/0,67 2,33/0,87	2,67/0,89 2,20/0,92
Ausdauer	2,22/0,83 2,12/0,75	2,25/0,73 2,36/0,79	1,89/0,75 2,18/0,81	2,00/0,67	2,33/0,78 2,00/0,71	2,58/0,79 2,20/0,92
Neugier	1,47/0,56 1,84/0,79	1,97/0,94 2,15/0,86	2,17/0,91 2,20/0,97	1,50/0,53	1,92/0,67 1,67/0,87	2,75/0,87 2,10/0,99
Neues einarbeiten	1,75/0,60 1,94/0,89	2,03/0,74 2,42/0,83	1,67/0,59 1,88/0,78	1,40/0,52	1,50/0,52 1,89/0,93	2,25/0,97 2,30/0,82
Begeisterung	1,86/0,76 1,94/0,79	2,17/0,74 2,32/0,86	2,14/0,80 2,22/1,01	1,60/0,70	1,75/0,75 1,44/0,73	2,75/1,06 2,00/0,94
Kreativität	2,36/0,87 2,26/0,80	2,50/0,94 2,30/0,78	2,39/0,96 2,45/0,89	2,30/0,95	2,58/0,67 2,11/0,78	2,92/1,00 2,70/0,95
Flexibilität	2,17/0,77 2,30/0,76	2,37/0,73 2,32/0,73	2,00/0,69 2,08/0,76	1,80/0,63	2,17/0,72 1,89/0,33	2,67/0,89 1,90/0,99
zuhören können	1,77/0,68 1,80/0,72	2,14/0,88 2,09/0,69	2,11/0,76 2,10/0,78	1,60/0,70	2,00/0,74 2,22/0,97	2,17/0,58 1,90/0,99
sich mitteilen können	2,36/0,76 2,20/0,88	2,46/0,85 2,54/0,78	2,20/0,68 2,13/0,70	1,70/0,82	2,33/0,78 2,44/0,88	2,33/0,65 1,80/0,79
Kom. mit Fachfremden	2,39/0,90 2,34/0,87	2,69/0,68 2,61/0,74	2,91/0,85 2,58/0,92	1,90/0,88	2,42/0,90 2,33/0,71	2,92/0,79 2,20/0,92
Einfühlungsvermögen	2,44/0,65 2,46/0,97	2,43/0,65 2,50/0,69	2,60/0,88 2,42/0,85	1,70/0,67	2,42/0,79 2,56/0,88	2,67/1,07 2,10/0,99
Integration im Team	2,31/0,82 2,08/0,67	2,39/0,69 2,22/0,70	1,89/0,57 2,15/0,68	1,90/0,57	2,42/1,00 2,11/0,60	2,33/0,98 2,40/0,70
Kritikfähigkeit	2,28/0,70 2,16/0,65	2,33/0,76 2,20/0,65	2,14/0,59 2,19/0,82	2,10/0,74	2,42/1,00 2,22/0,97	2,67/0,78 2,10/0,88

Kompetenz	Bachelorstudenten			Masterstudenten		
	Start Vorkmp	Ende Vorkmp	Ende Lerneff.	Start Vorkmp	Ende Vorkmp	Ende Lerneff.
Hilfsbereitschaft	1,81/0,58 1,90/0,81	1,92/0,84 2,07/0,71	1,92/0,69 2,06/0,73	1,50/0,53	1,58/0,67 1,89/0,33	2,42/1,00 1,90/0,99
Durchsetzungsvermög.	2,56/0,65 2,46/0,84	2,64/0,83 2,59/0,80	2,28/0,85 2,29/0,74	1,70/0,67	2,58/0,90 2,33/0,87	2,17/0,94 2,20/0,92
Kompromissbereit.	2,14/0,64 2,18/0,69	2,22/0,59 2,28/0,58	1,92/0,69 2,17/0,72	2,10/0,57	2,25/0,75 2,11/0,60	2,25/0,62 2,50/0,85
Visualisierungskomp.	2,44/0,77 2,32/0,68	2,74/0,66 2,57/0,65	2,66/0,73 2,29/0,68	2,20/1,03	2,08/0,67 2,56/0,88	2,83/0,94 3,00/0,94
Präsentationskomp.	2,78/0,80 2,42/0,86	2,92/0,60 2,63/0,71	2,78/0,76 2,41/0,82	2,10/0,74	2,00/0,43 2,22/0,83	3,17/0,83 2,40/1,07
Arbeit verteidigen	2,44/0,77 2,16/0,65	2,44/0,69 2,35/0,67	2,22/0,72 2,17/0,83	1,80/0,63	2,00/0,95 2,11/0,78	2,83/1,11 2,30/1,06
Fähigkeit zu leiten	2,92/0,91 2,72/0,88	2,97/0,70 2,93/0,74	2,89/0,89 2,73/0,96	2,30/0,95	3,00/0,95 2,33/0,71	1,92/0,90 2,00/0,82
Entscheidungsfähigkeit	2,44/0,77 2,60/0,70	2,50/0,51 2,61/0,74	2,14/0,72 2,33/0,72	2,10/0,99	2,50/1,00 2,44/0,88	2,25/0,75 2,10/0,99
Deligieren können	2,89/0,82 2,66/0,82	3,03/0,51 2,78/0,66	2,56/0,81 2,60/0,84	2,30/0,95	2,67/0,78 2,78/0,83	2,17/0,58 1,90/0,99
Risikobereitschaft	2,89/0,67 2,56/0,70	2,89/0,71 2,72/0,66	2,67/0,89 2,53/0,83	2,70/0,82	2,92/0,51 3,00/0,87	2,58/0,90 2,40/0,70
Verantwortungsbewusstsein	2,22/0,72 2,05/0,67	2,33/0,63 2,28/0,69	2,03/0,77 1,98/0,67	2,00/0,94	2,17/0,72 2,11/0,93	2,08/0,90 2,30/0,95
Systemdenken	2,56/0,65 2,33/0,77	2,53/0,70 2,52/0,72	2,11/0,85 2,29/0,74	2,10/0,57	2,18/0,98 2,67/0,71	2,73/0,79 2,50/0,71
Konflikte managen				2,30/0,95	2,75/0,97 2,22/0,97	2,17/0,94 2,30/1,05
Teambildung				2,20/0,92	3,08/0,79 2,44/1,01	1,92/0,90 2,40/1,07

Tab. 3 Auswertung des Fragebogens bzgl. der Kompetenzen.
 Start = Befragung vor dem Softwarepraktikum, Ende = Befragung nach dem Softwarepraktikum, Vorkmp = Vorkompetenz, Lerneff. = Lerneffekt. Es ist jeweils in der ersten Zeile der Mittelwert und die Standardabweichung für die Befragung 2007 und in der zweiten Zeile analog für 2008 dargestellt. Werteskala: 1 (sehr) bis 4 (kaum). Auf die grau hinterlegten bzw. fett gedruckten Werte wird im Text direkt Bezug genommen.

Literatur

- [ASIIN 2005] ASIIN Fachausschuss Informatik: Fachspezifisch erganzende Hinweise. (Akkreditierungsagentur fur Studiengange der Ingenieurwissenschaften, der Informatik, der Naturwissenschaften und der Mathematik e.V.), 2005.
- [ASIIN 2006] ASIIN: Antrag auf Reakkreditierung durch den Akkreditierungsrat. (Akkreditierungsagentur fur Studiengange der Ingenieurwissenschaften, der Informatik, der Naturwissenschaften und der Mathematik e.V.), 2006.
- [BMBF 2005] Bundesministerium fur Bildung und Forschung: Bachelor- und Master-Studiengange in ausgewahlten Landern Europas im Vergleich zu Deutschland, 2005.
- [Demuth et al. 2002] B. Demuth, M. Fischer, H. Hussmann: Experience in early and late software engineering project courses. In Proc. 5th Conference on Software Engineering Education Training (CSEE&T 2002), S. 241–248, IEEE Computer Society, 2002.
- [Fincher et al. 2001] S. Fincher, M. Petre, M. Clark (Herausgeber): Computer Science Project Work: Principles and Pragmatics. Springer, London, Berlin, 2001.
- [Lindig/Zeller 2005] C. Lindig, A. Zeller: Ein Softwaretechnik-Praktikum als Sommerkurs. In: K.-P. Lohr, H. Lichter (Herausgeber), Software Engineering im Unterricht der Hochschulen SEUH 2005, S. 68-80, Heidelberg, dpunkt.verlag, 2005.
- [Ludewig/Reiing 1998] J. Ludewig, R. Reiing: Teaching what they need instead of teaching what we like – the new software engineering curriculum at the University of Stuttgart. Information & Software Technology, 40(4):239–244, 1998.
- [Rainwater 2002] J.H. Rainwater: Herding Cats: A Primer for Programmers Who Lead Programmers. Apress, Berkeley, 2002.
- [Schlimmer et al.1994] J. C. Schlimmer, J. B. Fletcher, L. A. Hermens: Team-oriented software practicum. IEEE Trans. Educ., 37:212–220, 1994.
- [Schneider 1994] K. Schneider: Komm, wir spielen Projektleiter! – Ein Lehrspiel fur Software Engineering. In: H. Humann, B. Paech, Software Engineering im Unterricht der Hochschulen SEUH '94, S. 118-128, Stuttgart, Teubner, 1994.
- [Schneider 2005] K. Schneider: Vier Formen der Erfahrungsvermittlung im Studium. In: K.-P. Lohr, H. Lichter (Herausgeber), Software Engineering im Unterricht der Hochschulen SEUH 2005, S. 16-25, Heidelberg, dpunkt.verlag, 2005.
- [Stoyan/Glinz 2005] R. Stoyan, M. Glinz: Methoden und Techniken zum Erreichen didaktischer Ziele in Software-Engineering-Praktika. In: K.-P. Lohr, H. Lichter (Herausgeber), Software Engineering im Unterricht der Hochschulen SEUH 2005, S. 2-15, Heidelberg, dpunkt.verlag, 2005.
- [Vigenschow/Schneider 2007] U. Vigenschow and B. Schneider: Soft Skills fur Softwareentwickler. dpunkt.verlag, Heidelberg, 2007.
- [Weicker et al. 2006] N. Weicker, B. Draskoczy, K. Weicker: Fachintegrierte Vermittlung von Schlusselkompetenzen der Informatik . In: P. Forbrig, G. Siegel, M. Schneider (Herausgeber), HDI 2006: Hochschuldidaktik der Informatik, S. 51-62, Bonn, Gesellschaft fur Informatik, 2006 .