

# Softwaretechnologiepraktikum im Grundstudium: Universitäres oder Reales Projekt?

---

Birgit Demuth<sup>+</sup>, Lothar Schmitz<sup>++</sup>, Barbara Wittek<sup>+</sup>

+ Fakultät Informatik, Institut für Software- und Multimediatechnik, Technische Universität Dresden  
01069 Dresden

{Birgit.Demuth, Barbara.Wittek}@inf.tu-dresden.de

++ Fakultät Informatik, Institut für Softwaretechnologie, Universität der Bundeswehr München  
85577 Neubiberg

lothar@informatik.unibw-muenchen.de

## Zusammenfassung

*Ein gemeinsam entwickeltes Softwaretechnologiepraktikum ist an der Fakultät Informatik der TU Dresden und an der Fakultät Informatik der UniBw München obligatorischer Bestandteil im Grundstudium. Das Praktikum wird an beiden Hochschulen in gleicher Weise als „internes Praktikum“ im universitären Rahmen durchgeführt. Seit einigen Jahren erhält an der TU Dresden ein Teil der Studenten die Möglichkeit, unter vergleichbaren Rahmenbedingungen ein reales Projekt als „externes Praktikum“ in regionalen Firmen oder wissenschaftlichen Institutionen durchzuführen. Über die Erfahrungen, die mit dem Praktikum generell, speziell aber mit den beiden alternativen Ausbildungskonzepten gesammelt wurden, wird im Folgenden berichtet.*

## 1 Einleitung

In vergangenen SEUH-Workshops wurden häufig die Konzeption sowie Erfahrungen bei der Durchführung von Software-Praktika vorgestellt; in [SEU 99] beispielsweise war dem Thema Software-Praktika eine eigene Sitzung gewidmet. Die geschilderten Erfahrungen beziehen sich auf sehr unterschiedliche Studiensituationen und Rahmenbedingungen: Praktika im Grundstudium oder im Hauptstudium, Praktika mit sehr vielen oder relativ wenigen Teilnehmern, Praktika, die „intern“ innerhalb der Hochschule oder „extern“ in Zusammenarbeit mit Software-Firmen durchgeführt wurden.

Das Softwaretechnologiepraktikum an der TU Dresden [Stp 04] ist für die Studiengänge Informatik, Medieninformatik und Informationssystemtechnik eine

Pflichtveranstaltung von zwei Semesterwochenstunden (SWS) im Grundstudium (auch im Ergänzungsstudiengang Softwaretechnik obligatorischer Bestandteil) und baut auf der Vorlesung/Übung "Softwaretechnologie I" auf. Analog ist die Situation in den Studiengängen Informatik und Wirtschaftsinformatik der UniBW München. Die aus dem Framework *SalesPoint* und einer umfangreichen Dokumentation bestehende Praktikumsinfrastruktur wurde von beiden Universitäten gemeinsam entwickelt, gepflegt und genutzt. Darüber haben wir in [Dem 99] bereits ausführlich berichtet.

Die Durchführung des Praktikums in einer frühen Phase des Studiums soll eine umfassende Ausbildung aller Studenten auf dem Gebiet der Softwaretechnologie gewährleisten, birgt aber auch Probleme in sich. Insbesondere die jährliche Teilnehmerzahl von ca. 500 Studenten an der TU Dresden (ca. 90 an der UniBw München) erfordert einen außerordentlich hohen organisatorischen und personellen Aufwand. Zeitweilige Engpässe bei den Hilfskraftmitteln vor etwa fünf Jahren gaben dann den Anstoß, an der TU Dresden die externe Form der Praktikumsdurchführung als mögliche Alternative zu erproben. Seitdem nehmen dort je nach Anzahl der verfügbaren Praktikumsplätze jährlich 15 - 20 % der Studenten am externen Praktikum teil. Die Studenten müssen sich explizit dafür bewerben und werden aufgrund ihrer Vorleistungen in der Grundlagenlehrveranstaltung Softwaretechnologie ausgewählt.

Für uns ergab sich die seltene Gelegenheit, die beiden unterschiedlichen Formen der Praktikumsdurchführung bei identischer Studiensituation über einige Jahre nebeneinander zu beobachten und zu vergleichen.

Im Folgenden beschreiben wir Ziele und Rahmenbedingungen des Softwarepraktikums innerhalb eines Massenstudiums (Abschnitt 2) anhand der internen Praktikumsform, die davon abweichenden Merkmale externer Projekte (Abschnitt 3) sowie Beobachtungen und Erfahrungen, die wir im Laufe der Jahre bei der Durchführung dieses Praktikums mit mehreren tausend Studenten gewinnen konnten (Abschnitt 4). Abschnitt 5 fasst aus unserer Sicht wichtige und praxisrelevante Lehren für unser Ausbildungskonzept zusammen.

## 2 Ziele und Rahmenbedingungen

Das Praktikum hat die Vermittlung objektorientierter Softwareentwicklungstechniken im Rahmen eines Softwareprojektes sowie das Erlernen arbeitsteiliger und teamorientierter Arbeitsweisen zum Ziel. Darüber hinaus wird die **Wiederverwendung** von Software vorgeschrieben, um die Arbeit mit Softwarebausteinen und deren Anpassung bewusst einzuüben [Dem 99]. Im internen Praktikum besteht daher die Aufgabe in der Erstellung einer Verkaufsanwendung auf Basis des Anwendungsframeworks *SalesPoint*. Es werden Teams von meist fünf Studenten gebildet, die eine grobe Beschreibung einer komplexen Aufgabenstellung als Basis für ihre Arbeit erhalten. Zunächst ist eine Anforderungsermittlung durchzuführen. Dann erfolgt in mehreren Schritten die Modellierung des zukünftigen Produktes mit Hilfe

verschiedener UML-basierter Beschreibungstechniken. Das Ziel der Modellierung ist ein auf die wiederzuverwendenden Softwarebausteine abgestimmter Entwurf, der dann mittels der Programmiersprache Java umgesetzt werden muss.

Zur **Teambildung** können die Studenten vorab Wünsche äußern, die nach Möglichkeit erfüllt werden. Für die Arbeitsteilung innerhalb der Teams wird das Chefprogrammiererprinzip empfohlen. Ein Student übernimmt die Rolle des Chefprogrammierers, der Entscheidungen trifft und als Ansprechpartner für Kunden und Betreuer fungiert. Die weiteren Rollen Assistent, Sekretär und Programmierer werden unter den restlichen Teammitgliedern aufgeteilt.

Für jedes Team im internen Praktikum wird ein studentischer **Tutor** engagiert, der den Softwareentwicklungsprozess kontrolliert, die Rolle des Kunden übernimmt und den Gruppen bei auftretenden technischen und kommunikativen Problemen hilfreich zur Seite steht. Die Tutoren haben dabei eine hohe Verantwortung, da Entscheidungen über Korrektheit von Dokumenten, Einhaltung von Zeitvorgaben und Beteiligung aller Teammitglieder am Projekt zu treffen sind. Die Teams müssen wöchentlich einmal an Pflichtkonsultationen mit ihrem Betreuer teilnehmen. Hierbei werden Ergebnisse besprochen, Probleme geklärt, die Einhaltung des Zeitplans überprüft etc.

Das interne Praktikum ist in **Phasen** eingeteilt, für die ein strenger Zeitplan vorgegeben wird. Nach zweiwöchiger Einarbeitungsphase, in der sich die Studenten die Grundlagen des wiederzuverwendenden Anwendungsframeworks aneignen sollen, entscheidet ein Zwischentest über die weitere Teilnahme am Praktikum. Die Bearbeitung der Aufgabe erfolgt danach in den Phasen Analyse, Entwurf, Implementation/Test und Wartung. Nach Abschluss dieser Phasen stellen die Teams ihre Ergebnisse in einer Endpräsentation vor. Für die Bearbeitung des Projektes stehen den Studierenden insgesamt 12 Wochen zur Verfügung. Außerdem ist strikt vorgegeben, welche Ergebnisse in jeder Phase entstehen sollen, wie diese anzufertigen sind. Dieses Vorgehen bietet zwar wenig Raum für eigene Experimente, hilft aber den meisten Teams, das Praktikum erfolgreich zu absolvieren. Sie besitzen bis zu diesem Zeitpunkt oft keine Erfahrung mit der Durchführung von Softwareprojekten und benötigen daher ausführliche Anleitung.

Ein großer Vorteil des internen Praktikums ist in der umfangreichen **Dokumentation** und den zahlreichen Hilfsangeboten seitens des Lehrstuhls zu sehen. Für die Arbeit mit dem zur Wiederverwendung im universitären Praktikumskonzept vorgeschriebenen Anwendungsframework *SalesPoint* steht vielfältige Unterstützung bereit: ein vollständig implementiertes einfaches Beispiel ("Videoautomat"), zwei „Kochbücher“ zu Implementierungsfragen („Hooks“ und „HowTos“) sowie das ständig verfügbare Expertenwissen der *SalesPoint*-Entwickler über ein Praktikumsforum. Darüber hinaus ist das Framework selbst gut dokumentiert (Technischer Überblick, API-Spezifikation).

Neben der eigentlichen Anwendung entstehen im Verlauf des Praktikums weitere Artefakte, die von den Gruppen ausführlich zu dokumentieren sind. Dazu ist von jedem Team eine **Webseite** zu entwerfen, auf der alle Ergebnisse des Praktikums zum

Zeitpunkt ihrer Entstehung zu veröffentlichen sind. Diese Maßnahme dient hauptsächlich der Kontrolle von Qualität und termingerechter Anfertigung der Dokumente, hält die Studierenden durch die Außenwirkung aber auch zu vollständiger und gewissenhafter Arbeit an.

Für alle Studierenden des Praktikums ist die Verwendung des **Versionsverwaltungssystems** CVS vorgeschrieben. Dies vermittelt den Nutzen eines solchen Systems in der arbeitsteiligen Softwareentwicklung. Auf der anderen Seite stellt das CVS ein Kontrollinstrument dar, das den Betreuern der Teams vor allem während der Implementierung erlaubt, den Projektstand zu überwachen und die Beteiligung aller Studenten am Projekt zu überprüfen. Sowohl für das interne als auch für das externe Praktikum wird die entsprechende Infrastruktur zentral bereitgestellt.

Als positiv für den Verlauf des internen Praktikums kann auch die **Homogenität und Vielzahl** der Projekte angesehen werden. Da alle Teams den gleichen zeitlichen Restriktionen und organisatorischen Rahmenbedingungen unterliegen und das *SalesPoint*-Praktikum auf jahrelange Erfahrung zurückgreifen kann, entstand eine ausführliche Praktikumsdokumentation und es gibt sehr viele Beispielprojekte im Web. Außerdem stoßen viele Teams während ihrer Arbeit auf die gleichen Probleme, so dass gegenseitige Hilfe (z.B. im Praktikumsforum) möglich ist. Und nicht zuletzt waren die studentischen Betreuer der Gruppen mit vielen anstehenden Schwierigkeiten auch während ihres eigenen Praktikums konfrontiert und können so kompetente Hilfe anbieten.

### 3 Das externe Praktikum

Bezüglich der Rahmenbedingungen und des Anforderungsniveaus unterscheiden sich internes und externes Praktikum nicht wesentlich, eher bei Aufgabenstellung, Arbeitsbedingungen und Kontrollmechanismen. Während im internen Praktikum die Aufgabe in der Erstellung einer Verkaufsanwendung auf Basis des am Lehrstuhl entwickelten Anwendungsframeworks *SalesPoint* besteht, hängt die **Aufgabenstellung** im externen Praktikum von den Anforderungen der jeweiligen Firma/Institution ab. Das Spektrum reicht dabei von hauseigenen Werkzeugen über Prototypen zur Ermittlung von Produktanforderungen und zur Nutzerakquisition bis zu realen (Web-)Anwendungen. Es wird jedoch darauf geachtet, dass der Umfang in etwa dem der universitären Projekte entspricht. Das setzt eine enge Zusammenarbeit zwischen dem Lehrstuhl und den am Praktikum beteiligten Firmen voraus: Damit die Rechte und Pflichten aller Beteiligten gewahrt werden können, wird bereits in der Vorbereitungsphase ein Kooperationsvertrag geschlossen. Auf der Basis dieser Vereinbarung können während der Bearbeitungszeit auftretende Probleme sowie berechnete Interessen der Firmen (etwa Geheimhaltung) angemessen berücksichtigt werden. Selbstverständlich hat dabei der Lernerfolg Priorität gegenüber Anforderungen an den Funktionsumfang des Produkts. Nicht zu unterschätzen ist der hohe Aufwand bei der Vorbereitung externer

Projekte. Das betrifft die Erstellung der Kooperationsverträge sowie die Einarbeitung in jeweils neue Themenstellungen und in die konkreten organisatorischen und technischen Rahmenbedingungen bei der jeweiligen Firma.

Für das externe Praktikum wird ein ähnlicher Ablauf angestrebt wie für das interne, jedoch wird hier von den Studierenden eine größere Eigenverantwortung erwartet. Insbesondere können sie in Absprache mit ihren **Betreuern** den im internen Praktikum vorgeschriebenen Softwareentwicklungsprozess an ihre Situation anpassen. Die Firmen des externen Praktikums betreuen ihre Teams autonom. Wenn es jedoch gewünscht wird, stellt der Lehrstuhl ebenfalls einen studentischen Betreuer zur Verfügung, der die softwaretechnische Seite des Praktikums überwacht, während die Firma die Kundenrolle übernimmt. Diese Praxis hat sich als günstig erwiesen, da so die Einheitlichkeit der Anforderungen im internen und externen Praktikum besser gewährleistet ist; künftig wollen wir daher regelmäßig so vorgehen.

Im externen Praktikum verwenden die Teams verschiedenste **Technologien**, Frameworks und Softwarekomponenten wieder, zum Teil Open Source-Software, zum Teil firmeninterne Entwicklungen, zum Teil kommerzielle Produkte. Die Dokumentation dazu ist oft unzureichend und Expertenwissen gar nicht oder nur über Umwege erhältlich. Die Studenten müssen stärker auf eigene Recherchen und Experimente setzen, was oft einen erhöhten Arbeitsaufwand bedeutet. Allerdings ist dadurch der persönliche Gewinn auch höher, da sich die Studenten im externen Praktikum oft selbstständig mit sehr aktuellen Technologien beschäftigen müssen, was ihnen im weiteren Verlauf ihres Studiums bzw. im späteren Berufsleben von Nutzen sein kann.

Im externen Praktikum ist die Forderung nach Veröffentlichung von Modellierungsergebnissen und Implementationsdetails meist nicht durchzusetzen, da hier die Geheimhaltung von Firmeninterna Vorrang hat. Das führt häufig dazu, dass die Studenten die **Dokumentation** generell vernachlässigen, was sich am Ende des Projektes auch in einer mangelhaften Anwender- und Entwicklerdokumentation widerspiegelt.

## 4 Beobachtungen und Erfahrungen

Die Planmäßigkeit des Softwareentwicklungsprozesses hat im Praktikum einen hohen Stellenwert. Während jedoch im internen Praktikum durch die lückenlose Betreuung die Einhaltung eines sauberen **Entwicklungsprozesses** durchgesetzt wird, neigen Studenten im externen Praktikum oft zu einer missverstandenen Form des Extreme Programming. Die Ursache dafür liegt darin, dass die Betreuung in den Firmen eher *ergebnisorientiert* verläuft, im universitären Rahmen dagegen eher *prozessorientiert*. Andererseits stellt die größere Freiheit im externen Praktikum auch eine Chance für einen größeren Lernerfolg dar. Im internen Praktikum erhalten die Studenten strenge Vorgaben, zu welchem Zeitpunkt welches **UML-Diagramm** anzufertigen ist. Dabei erschließt sich für viele Studierenden oft nicht der Zweck des jeweiligen Modellierungsschrittes für den weiteren Verlauf des Projektes. Das "Malen von

Diagrammen" wird nur als Belastung empfunden. Im externen Praktikum bekommen die Studenten zwar auch Hinweise auf die Verwendung von UML-Modellen, sie müssen die Entscheidung über deren sinnvollen Einsatz größtenteils jedoch selber treffen. Dabei erfolgt im positiven Fall eine stärkere Auseinandersetzung mit den Zielen und Möglichkeiten der UML-Modellierung und der Lernerfolg ist höher.

Beim Vergleich der **Ergebnisse** im externen und internen Praktikum lassen sich kaum qualitative oder quantitative Unterschiede feststellen. Der Arbeitsaufwand ist für die Studierenden sowohl im internen als auch im externen Praktikum wesentlich höher als die vorgesehene SWS. Da die Firmen die im Praktikum entstandenen Anwendungen häufig tatsächlich verwenden oder als Basis für weitere Entwicklungen nutzen, besteht für die Studenten eine hohe Motivation zur Erstellung eines qualitativ hochwertigen Produktes. Aber auch innerhalb des internen Praktikums entstehen immer wieder interessante und ausgereifte Softwarelösungen. Die Motivation zur Anfertigung guter Ergebnisse gründet sich dabei hauptsächlich auf das Erlebnis, systematisch und arbeitsteilig ein eigenes Produkt entstehen zu lassen und im Vergleich mit anderen Teams gut abzuschneiden. Die qualitative Bandbreite der universitären Projekte ist durch die Vielzahl der Produkte allerdings sehr groß. Quantitativ sind die Applikationen aus externem und internem Praktikum vom Funktionsumfang her ähnlich und auch bezüglich ausgewählter Metriken ergeben sich Gemeinsamkeiten. Interessant ist dabei, dass die durchschnittlichen Werte für Lines of Code (LOC) und Anzahl der Klassen in ähnlichen Größenordnungen liegen, externe Gruppen aber durchschnittlich einen geringeren Zeitaufwand für ihre Arbeit angeben. Das kann sowohl auf ein effektiveres Vorgehen als auch auf bessere Voraussetzungen, z.B. bezüglich Programmiererfahrungen, hinweisen.

Bezüglich **technischer Voraussetzungen** kann über keine erheblichen Unterschiede zwischen externem und internem Praktikum berichtet werden. Intern ist es den Studenten freigestellt, ob sie im Fakultätsrechenzentrum oder mit eigener Technik arbeiten. Die Verwendung von Softwareentwicklungswerkzeugen wird zwar empfohlen und durch Bereitstellung von Lizenzen und Tutorials unterstützt, aber nicht vorgeschrieben. Das externe Praktikum weicht nur dann von diesen Gegebenheiten ab, wenn die jeweilige Firma den Teams eigene Arbeitsplätze zur Verfügung stellt oder die Verwendung bestimmter Werkzeuge explizit vorschreibt. Die Integration der studentischen Projekte in die firmeninterne Infrastruktur kann dabei sowohl von Vorteil als auch von Nachteil sein. Einerseits entfallen Entscheidungen zur Entwicklungsumgebung und kann das technische Knowhow der Firma genutzt werden, andererseits entstehen aber Restriktionen, mit denen sich die Studenten zusätzlich zur eigentlichen Projektarbeit auseinandersetzen müssen.

Studenten, die sich vor dem Praktikum bereits kannten, können sich schneller aufeinander einstimmen, besser gemeinsame Termine finden und früher mit der eigentlichen Arbeit beginnen. Dafür werden bestehende Probleme aus falsch verstandener Freundschaft oft nicht angesprochen und führen mitunter später zum Zerwürfnis in der Gruppe. Umgekehrt benötigen Teams, deren Mitglieder sich zunächst nicht kennen,

zu Beginn einen höheren **Kommunikations- und Koordinationsaufwand**, gehen aber meist offener mit Problemen um. Beide Konstellationen können daher sowohl zum Erfolg des Projektes als auch zu dessen Scheitern führen. Im externen Praktikum ist die Ausrichtung auf das gemeinsame Ziel jedoch häufig stärker ausgeprägt, so dass hier eher Lösungen für Teamprobleme gesucht werden und seltener Teams oder einzelne Mitglieder aus dem Praktikum ausscheiden. Die Auswahl besonders leistungsstarker Studenten für das externe Praktikum kann für die Teamarbeit sowohl Vor- als auch Nachteile haben. Im internen Praktikum sind die Gruppen oft heterogen bezüglich der Fähigkeiten ihrer einzelnen Mitglieder. Teamprobleme resultieren oft aus unterschiedlicher Arbeitsleistung. Im externen Praktikum sind alle Teammitglieder auf einem etwa gleich hohen Niveau und werden damit eher gleich stark am Projekt beteiligt. Evtl. Teamprobleme sind hier oft darauf zurückzuführen, dass starke Individualisten aufeinander treffen. Diese Studenten entsprechen oft dem Klischee des „Hackers“, der zudem keine Teamerfahrung mitbringt. Bei der Betreuung sind beide Phänomene zu berücksichtigen und entsprechend zu behandeln.

Es hat sich gezeigt, dass es notwendig ist, den Tutoren konkrete Entscheidungshilfen und „Druckmittel“ zur **Durchsetzung der Praktikumsanforderungen** zur Verfügung zu stellen und der Einsatz dieser häufig auch erforderlich ist. Speziell im internen Praktikum führt die Arbeit an einem komplexen, zeitaufwändigen Projekt ohne realen Kunden bei vielen Studenten zu geringem Engagement. Der Lernerfolg und der Nutzen für zukünftige Projekte sind für sie in dieser frühen Phase des Studiums oft schwer abzuschätzen. Daher neigen die Studenten dazu, die Anforderungen nicht ernst zu nehmen sowie Fehler bzw. Zeitüberschreitungen zu bagatellisieren. Hier ist es entscheidend, Meilensteine klar zu definieren und deren Nichteinhaltung mit Konsequenzen zu verbinden. Ebenso muss das Einfordern definierter Dokumente und Ergebnisse zu konkreten Terminen durch die Tutoren konsequent durchgesetzt werden. Problematisch ist in diesem Zusammenhang die Einschätzung von Zwischenergebnissen. Häufig können abgegebene Dokumente oder Codefragmente nicht explizit als richtig oder falsch eingestuft werden. So sind z.B. bereits erste richtige Ansätze als Erfolg zu werten, während Fehler aufgrund mangelnder Arbeitsmoral nicht toleriert werden können. Daraus resultieren gewisse Unsicherheiten sowohl bei den Tutoren als auch bei den Studierenden, die durch strenge Richtlinien und intensive Anleitung der Tutoren minimiert werden müssen. Dazu finden wöchentliche Tutorenbesprechungen statt.

Bei den Studenten im externen Praktikum ist der Einsatz von Druckmitteln seltener nötig. Durch die Arbeit in einer Firma an einer realen Aufgabe ist die Motivation erheblich höher. Hier liegen die Probleme dann oft in anderen Bereichen. So können Diskrepanzen zwischen den Anforderungen der Firma und den Fähigkeiten der Studenten entstehen, die im Grundstudium mit vielen Technologien noch nicht vertraut sind. Auch dadurch kann es zu einer Demotivation kommen, die jedoch andere Ursachen hat und mit anderen Mitteln als im internen Praktikum zu beheben ist. Als hilfreich hat sich das Durchführen einer **Zwischenpräsentation** erwiesen, bei der die

Studierenden den aktuellen Arbeitsstand vorstellen und selbst einen Überblick über bisher Erreichtes erhalten. Dabei können dann auch seitens des Lehrstuhls die Ansprüche der Firmen mit den Leistungsgrenzen der Studenten abgeglichen werden. Im internen Praktikum sind solche Zwischenpräsentationen aufgrund der Vielzahl der Gruppen nicht möglich, so dass Probleme oft tatsächlich erst gegen Projektende sichtbar werden.

## 5 Was haben wir gelernt?

Neben den oben beschriebenen konkreten Beobachtungen und Erfahrungen aus unserem Praktikum ergeben sich für uns eine Reihe von Lehren, die sich vermutlich auch gewinnbringend auf ähnliche Veranstaltungen andernorts übertragen ließen:

Die **Betreuung durch studentische Tutoren** hat sich als sehr effektiv erwiesen. Dies setzt eine sorgfältige Auswahl der Tutoren sowie deren intensive Betreuung durch den Lehrstuhl voraus. Des Weiteren ist der wöchentliche Gedanken- und Erfahrungsaustausch der Tutoren untereinander ein wirksames Instrument bei der qualitativen Absicherung in der Ausbildung. Eine das Praktikum im Sommersemester 2004 begleitende psychologische Studie [iko 04] konnte die Vorbildwirkung des studentischen Tutors klar herausarbeiten. Tutoren, die motivieren, intellektuell anregen und Stärke ausstrahlen, können allerdings nur dann positiven Einfluss auf das Team und dessen Projektergebnisse nehmen, wenn sich die Teammitglieder auch engagieren.

Ein weiterer wesentlicher Faktor für den Erfolg des Praktikums ist die **Motivation der Studenten**. Hier ist das externe Praktikum klar im Vorteil, da die Identifikation mit einem realen Projekt leichter fällt als mit einem Übungsbeispiel und das Ziel der Arbeit deutlicher wird. Daher ist es notwendig, im internen Praktikum den persönlichen Wissenszuwachs stärker hervorzuheben und den Studenten häufiger positive Rückmeldungen zu geben. Zu Beginn des Praktikums ist die Demonstration eines vorbildlichen Projektes aus dem Vorjahr nützlich. Dem Bericht von Studenten wird erfahrungsgemäß eher Glauben geschenkt als dem von Lehrbeauftragten. Das Stiften von Preisen für die besten Praktikumssteams hat sich als weiteres Mittel zur Motivationssteigerung erwiesen.

**Reale Projekte** können nur mit leistungsfähigen und motivierten Studenten erfolgreich durchgeführt werden. Die Trennung des externen Praktikums vom universitären Umfeld birgt Risiken in sich. Die Firmen kennen den aktuellen Ausbildungsstand eines Studenten im Grundstudium unzureichend und stellen ggf. den Fähigkeiten der Studenten inadäquate Anforderungen an den Verlauf der Arbeit und an das Ergebnis. Außerdem wird in den Firmen meist zu wenig Wert auf die systematische Softwareentwicklung gelegt, sondern der Fokus zu stark auf das Endprodukt gerichtet. Das widerspricht den eigentlichen Zielen des Praktikums, softwaretechnische Kompetenzen zu fördern. Für beide Probleme stellt der Einsatz von studentischen Betreuern auch für externe Praktikumsgruppen eine geeignete Lösung dar. Durch den intensiven



Kontakt der Betreuer mit den Teams, dem Lehrstuhl und den Firmenvertretern kann sowohl das Arbeitspensum immer wieder abgestimmt als auch das Einhalten eines sauberen Entwicklungsprozesses kontrolliert werden. Für die Teilnehmer am externen Praktikum erweist sich die Möglichkeit, Erfahrungen auf dem Gebiet der Softwareentwicklung an einem realen Projekt unter kontrollierten Bedingungen zu sammeln, als besonders positiv. Da die Studenten hier stärker selbstverantwortlich bezüglich Zeiteinteilung, Einarbeitung in neue Technologien und Arbeitsteilung handeln und sich zudem einem realen Kunden gegenüber behaupten müssen, wird das Bewusstsein für die dabei entstehenden Probleme geschärft.

Die Teams im **internen Praktikum** benötigen aufgrund ihrer meist leistungsmäßig heterogenen Zusammensetzung ständige Kontrolle und erhöhte Aufmerksamkeit, um die gleichmäßige Beteiligung aller Studenten am Praktikum zu gewährleisten. Hier muss von erhöhten Betreuungsanforderungen ausgegangen werden, denen durch das Anbieten verschiedener Anlaufstellen bei Problemen (Forum, Tutor, Lehrstuhl) und umfangreicher organisatorischer und technischer Hilfestellungen Rechnung zu tragen ist. Durch die ständige Kontrolle der Teams und die intensive Anleitung im internen Praktikum lassen sich positive Auswirkungen auf den Lernerfolg erzielen. Die Studenten können hier die Softwareentwicklung an einem Übungsbeispiel „gefahrlos“ ausprobieren und finden dabei jederzeit Unterstützung in allen Bereichen, ohne dass z.B. Fehlentscheidungen negative Konsequenzen für das gesamte Projekt bedeuten. Wenn diese Chance von den Studenten erkannt wird, kann der Lerneffekt sogar höher ausfallen als im externen Praktikum, wo durch den höheren Erfolgsdruck schnell gewohnte Vorgehensweisen bei der Programmierung adaptiert und dabei die Ziele eines sauberen Softwareentwurfs nicht verinnerlicht werden. Eine Anforderung an die Studierenden aus dem letzten Praktikumszyklus, die sich auch nach studentischen Aussagen sehr bewährt hat, ist, dass jedes Teammitglied in der Entwurfsphase einen eigenen kleinen Prototypen schreiben muss.

Eine weitere wichtige Erfahrung bei der Durchführung des Praktikums sowohl für universitäre als auch für reale Projekte ist die **Durchsetzung eines festen Endtermins**. Im internen Praktikum kommt noch die konsequente terminliche Einhaltung von Meilensteinen dazu. Die teilweise geringe Motivation bzw. fehlende softwaretechnische, programmiertechnische und auch soziale Kompetenz der Studenten in den universitären Projekten führt häufig zu einer legeren oder sogar ablehnenden Haltung gegenüber dem Praktikum. Dem ist durch den drohenden Ausschluss aus dem Praktikum von einzelnen Teammitgliedern oder eines gesamten Teams entgegenzuwirken. Damit wird auch die Position der Betreuer gegenüber den Gruppen gestärkt und so die Durchsetzung softwaretechnischer Anforderungen gewährleistet. Im externen Praktikum ist die Ausübung von Druck auf die Studenten nicht oder kaum nötig, da hier qualitativ hochwertige Ergebnisse aus eigenem Antrieb entstehen und Termin- und Teamprobleme meist selbständig erkannt und gelöst werden.

Im Hinblick auf die Teilung des Angebots in **interne und externe Projekte** fassen wir zusammen:

- Unter direkter Anleitung und strikter Kontrolle durchgeführte Projekte eignen sich für unerfahrene Entwickler, d.h. für die Mehrzahl der Studenten im Grundstudium.
- Reale, stärker selbstverantwortliche Projekte im betrieblichen Umfeld stellen für die Gruppe der Studenten mit besserem softwaretechnischen Hintergrund eine interessante Herausforderung dar.
- Externe Projekte sind für den Lehrstuhl in Vorbereitung und Durchführung aufwändiger als interne Projekte.
- Die vorgesehene Semesterwochenstundenzahl wird künftig von zwei auf vier SWS erhöht, um dem Arbeitsaufwand und der Stellung des internen als auch des externen Praktikums besser gerecht zu werden.

Beide Konzepte lassen sich im Grundstudium erfolgreich durchführen, sofern geeignete Randbedingungen hergestellt werden. Mit dem Angebot beider Alternativen können wir auf die unterschiedlichen Interessen und Begabungen verschiedener Praktikumssteilnehmer eingehen.

## Literatur

[SEU 99] SEUH'99, B. G. Teubner, 1999

[Stp 04] TU Dresden, Fakultät Informatik: Softwaretechnologie-Praktikum

<http://www-st.inf.tu-dresden.de/Lehre/praktikum.htm>

[Dem 99] B. Demuth, H. Hußmann, S. Zschaler, L. Schmitz: Erfahrungen mit einem frameworkbasierten Softwarepraktikum. In [SEU 99]

[iko 04] TU Braunschweig, Institut für Psychologie: Initiative für Kommunikation in der Softwarebranche (ikoso). <http://www.ikoso.de/>