

Von anwendungsorientierter Softwareentwicklung zu anwendungsorientierten Lehrveranstaltungen - der Werkzeug & Material-Ansatz in der Lehre

Wolf-Gideon Bleek, Guido Gryczan, Carola Lilienthal, Martin Lippert, Stefan Roock, Henning Wolf, Heinz Züllighoven

Arbeitsbereich Softwaretechnik, Fachbereich Informatik, Universität Hamburg
e-mail: {bleek, gryczan, lilienth, lippert, roock, wolf, zuelligh}@informatik.uni-hamburg.de
<http://swt-www.informatik.uni-hamburg.de>

1 Zusammenfassung

Der Arbeitsbereich Softwaretechnik des Fachbereichs Informatik an der Universität Hamburg bildet seit seiner Gründung 1991 Studierende der Informatik im Bereich Softwaretechnik aus, insbesondere auf dem Gebiet der objektorientierten Softwareentwicklung. Als objektorientierte Methodik wurde der Werkzeug & Material-Ansatz (WAM) ausgearbeitet, der auf eine anwendungsorientierte Softwareentwicklung abzielt.

Passend zu den Anforderungen der Entwicklungsmethodik wurden neue Lehrformen erarbeitet und in einem Curriculum zusammengestellt. Im folgenden stellen wir die Prinzipien dieser neuen Lehrformen und ihre Umsetzung vor. Zuerst geben wir einen kurzen Überblick über unser Verständnis anwendungsorientierter Softwaretechnik. Darauf aufbauend präsentieren wir die verschiedenen Lehrveranstaltungen und ihre Beziehungen untereinander. Wir betonen dabei die Rückkopplung zwischen Lehrenden und Lernenden. Zur Verdeutlichung gehen wir auf einige Lehrveranstaltungen genauer ein und zeigen, wie die Grundprinzipien der aktiven Rückkopplung und der gemeinsamen Arbeit an einem Ergebnis (hier: ein Java-Rahmenwerk für WAM) Lehre und Forschung positiv beeinflussen können.

So soll dargestellt werden, daß auch anspruchsvolle Themen der aktuellen softwaretechnischen Forschung für Studierende erfahr- und erlernbar werden können.

2 Problemfelder

Für die Lehre stellen sich uns die Fragen:

- Wie können Studierende in der relativ kurzen Zeit des Hauptstudiums soweit mit der Objektorientierung und ihren Konzepten vertraut gemacht werden, daß sie den Stand der Kunst beherrschen?

- Wie können Studierenden praxisrelevante Forschungsergebnisse so vermittelt werden, daß sie das Handwerkszeug für einen möglichst nahtlosen Übergang in die berufliche Tätigkeit fortschrittlicher Entwicklerorganisationen besitzen?

Diese Fragen deuten auf ein Dilemma für die softwaretechnische Lehre. Zumindest an der Universität Hamburg ist der Anteil softwaretechnisch orientierter Veranstaltungen im Grundstudium eher gering. Dem steht eine ständig wachsende Themenliste allein im Bereich des technischen Grundwissens für das Hauptstudium gegenüber: Vermittlung einer durchgehenden objektorientierten Methodik (z.B. [Zül98]), objektorientierte Programmiersprachen, Entwurfsmuster, Rahmenwerke, Middleware- und Persistenzkonzepte, andere Entwicklungsmethoden und die Organisation von Softwareprojekten. Hierfür sieht der Studienplan zwei Jahre vor, aus denen in der Regel real rund drei Jahre werden. Dennoch, mit Studien- und Diplomarbeit und den anderen Fächern ist dies eine sehr knappe Zeit für eine solche Themenvielfalt.

Ein zweites Problem betrifft die Lehrformen. Wir betrachten Softwareentwicklung im wesentlichen als einen Lern- und Kommunikationsprozeß, der nur durch eine evolutionäre Vorgehensweise, also dem, was wir Autor-Kritiker-Zyklus nennen, sinnvoll angegangen werden kann. Unsere Erfahrungen in der Lehre machen deutlich, daß auch hier Konzepte nicht allein theoretisch und schon gar nicht nur frontal vermittelt werden können, sondern praktisch geübt werden müssen (s. [WKKL98]). Wir haben daher bei der Vermittlung des Werkzeug & Material-Ansatzes unsere eigenen Prinzipien aus der Softwareentwicklung ernstgenommen und erfolgreich mit neuen Lehrformen experimentiert.

Demgegenüber ist die Lehre in Hamburg noch weitgehend an konventionellen Formen wie Vorlesung, Seminar und Übung orientiert. Hinzu kommt, daß die Studierenden im Hauptstudium vielfach keine Vollzeitstudenten sind. Ein bedeutender Anteil der Teilnehmer unserer Lehrveranstaltungen arbeitet. Dadurch erwachsen zusätzliche Anforderungen an das Studium: Die angebotenen Veranstaltungen müssen mit der Nebentätigkeit sowohl zeitlich als auch praktisch vereinbar sein. Eine Chance besteht darin, daß Studierende das erworbene Wissen in der Praxis direkt ausprobieren können. Im Gegenzug können sie selbst Erfahrungen aus den Anwendungsbereichen in die Forschung und Lehre einbringen. Es ist dabei allerdings zu berücksichtigen, daß das Niveau in den Unternehmen bei Projektarbeit und Technologieeinsatz recht unterschiedlich ist.

3 Anwendungsorientierte Softwaretechnik

Die *Softwaretechnik* befaßt sich mit der professionellen Entwicklung großer Softwaresysteme, wobei Anwendungssoftware im Vordergrund steht [FZ97]. Softwareentwicklung sowie die Anwendung softwaretechnischer Methoden und Tech-

niken ist für Unternehmen aber stets nur ein Mittel zum Zweck. Erfolgreiche Unternehmen sehen das Ziel ihrer Tätigkeit heute vielfach darin, kundenorientierte Dienstleistungen zu erbringen. Eine Softwareentwicklung und die dazugehörige Technik, die sich diesem Ziel unterordnet, nennen wir *anwendungsorientiert*.

Für eine anwendungsorientierte Softwaretechnik eröffnen sich vielfältige Themen. Zunächst sind Fragen nach den aktuellen technischen Entwicklungen zu beantworten. Objektorientierung ist dabei in ihren Facetten als Paradigma und Technik aus unserer Sicht am besten geeignet, anwendungs- und kundenorientierte Problemlösungen bereitzustellen. Die Gründe dafür haben wir an verschiedenen Stellen diskutiert (vgl. [Zül98]). Zu den objektorientierten Themen zählen u.a. Entwurfsmuster, Rahmenwerke, Web-Computing, Standards für verteilte Anwendungen (CORBA) und objektorientierte Datenbanken.

Weiterhin wird zunehmend erkannt, daß Softwaretechnik sich nicht nur auf die generische Beherrschung von Techniken konzentrieren darf (vgl. [CACM97]). Techniken müssen in Zusammenhang mit den jeweils zu konstruierenden Anwendungen gebracht werden. Erst durch diese ganzheitliche Betrachtungsweise wird das Potential der Softwaretechnik gewinnbringend nutzbar gemacht.

Für Softwareentwickler stellt sich damit die Frage, wie Techniken und das Wissen um sie (also die Technologien) in einem spezifischen Einsatzkontext verwendet werden können. Anders gefragt: Worin besteht die Pragmatik der jeweiligen Technologie und wie passen verschiedene Technologien zueinander, um ein für Anwender sinnvoll einsetzbares Hilfsmittel zu konstruieren? In diesem Zusammenhang läßt sich die Frage nach dem Stellenwert von Techniken für eine anwendungsorientierte Softwaretechnik beantworten.¹

Die Ausrichtung der Softwaretechnik-Lehre muß vor diesem Hintergrund gerechtfertigt werden. Für die Lehrenden in der Softwaretechnik bleibt zu klären, wie die Komplexität des Gebiets und der umfassende Anspruch, der sich mit der anwendungsorientierten Softwaretechnik verbindet, in einem überschaubaren Zeitraum systematisch vermittelt werden kann. Wir werden dieser Frage im folgenden am Beispiel von Rahmenwerken und Entwurfsmustern nachgehen.

4 Das OO-Curriculum: Autor-Kritiker-Zyklus revisited

Ziel dieses Abschnitts ist, das Softwaretechnik-Curriculum unter dem Gesichtspunkt zu diskutieren, wie Lehrveranstaltungsziel und -veranstaltungsform zusammenhängen. Dabei betrachten wir vor allem die objektorientierten Lehrveranstaltungen, die auf neue Art und Weise durchgeführt werden.

¹ In diese Richtung geht auch die Argumentation von M. Jackson zu sog. Problem Frames (vgl. [Jac95]).

Tabelle 1 skizziert die wesentlichen Veranstaltungen mit ihren Zielen und Formen. Wenn wir diese Lehrveranstaltungen mit unserem bisherigen Angebot vergleichen, dann lassen sich einige charakteristische Unterschiede nennen:

- Keine der aufgeführten Lehrveranstaltungen wird mehr nach dem “Vorlesung-Übung“-Schema durchgeführt.
- Die zentralen Lehrveranstaltungen sind auf lauffähige Ergebnisse und auf die Arbeit mit und an einem Rahmenwerk ausgerichtet.
- Die Lehrveranstaltungen laufen über zwei Semester.

Lehrveranstaltung	Ziel	Form
Objektorientierte Software-Entwicklung (OOSE I & II)	Die Studierenden sollen die Prinzipien und Konzepte der Objektorientierung kennenlernen. OOSE I: Systementwurf. OOSE II: Systemkonstruktion	4 x 3 Tage Kompaktseminar (jeweils zu Beginn und Ende eines Semesters)
Entwicklung & Konstruktion objektorientierter Anwendungen I & II	I: Beherrschung von Interviewtechniken, Fähigkeit zur Erstellung von Analysedokumenten und Demonstrationsprototypen. II: Beherrschung wesentlicher Konstruktionstechniken für interaktive und verteilte Anwendungen	2-semesteriges Projekt mit übergreifender Aufgabe; begleitende Intensiv-Kurse (2-tägig); parallel zu OOSE I & II
JWAM - Ein Java-Rahmenwerk für WAM (JWAM)	Die Studierenden sollen Konstruktionsprinzipien und Vorgehensweise bei der Entwicklung von Rahmenwerken kennenlernen.	Arbeitsgruppe mit regelmäßigem Seminartermin
Ausgewählte Themen der Softwaretechnik (AT)	Vorstellung und Evaluation aktueller Ansätze im Bereich der Softwaretechnik; Vorstellung von Ergebnissen aus Studien-/Diplomarbeiten sowie Projekten	Seminar

Tabelle 1: Lehrveranstaltungen zur Objektorientierung

Diese Lehrveranstaltungsformen wurden begründet auf einem engen Autor-Kritiker-Zyklus zwischen Lehrenden und Lernenden ausgerichtet:

- Blockveranstaltungen und Gruppenarbeit steigern den Lernerfolg und die Einsatzbereitschaft der Studierenden. So lassen sich mehr Inhalte in einem kürzeren Zeitraum vermitteln.
- Einjährige Lehrveranstaltungen intensivieren die Zusammenarbeit der Studierenden untereinander und mit den Lehrenden. Die Bedeutung der Gruppenarbeit wird betont.

- Eingebettete Anwendungssysteme (vgl. [Leh90]) lassen sich nicht in kürzeren Zyklen realisieren.
- Die praktische Umsetzung von Entwürfen in Software ist unersetzlich, um den Studierenden objektorientierte Technologien (Entwurfsmuster/Rahmenwerke) zu vermitteln.
- Die Bedeutung von Kooperation und Kommunikation für Softwareentwicklung wird erst in Projektarbeit ab einer bestimmten „Größenordnung“ deutlich.
- Die Kooperation mit Industriepartnern erlaubt uns, den Studierenden die Erfahrung eines realen Projekts zu vermitteln.
- Studierende können ihre Erfahrungen aus der Praxis in den Lehrbetrieb einfließen lassen.
- Themen für Studien- oder Diplomarbeiten evolvieren aus den bearbeiteten Projekten.

Begleitend zu den Lehrveranstaltungen führen wir Seminare mit wechselnden Themen durch. Dabei stehen oft Literaturarbeit und Fragen aus eigenen Studien- und Diplomarbeiten im Vordergrund. Einige Seminare haben höhere praktische Anteile, z.B. die Entwicklung oder Weiterentwicklung von Komponenten, der am Arbeitsbereich Softwaretechnik eingesetzten Java- und C++-Rahmenwerke. Die OOSE-Lehrveranstaltung und das damit verzahnte Projekt beschreiben wir in einem anschließenden Abschnitt; ebenso die JWAM-Veranstaltung. Das Seminar „Ausgewählte Themen der Softwaretechnik“ wird durchschnittlich von ca. 25 Studierenden besucht. Es bildet durch seine Kontinuität ein Zentrum zum Austausch zwischen allen Studierenden, die sich in Softwaretechnik vertiefen.

4.1 Von der Vorlesung zur aktiven Zusammenarbeit

„Traditionell“ haben wir den am Arbeitsbereich vertretenen Werkzeug & Material-Ansatz im Hauptstudium im Rahmen einer Vorlesung (2 SWS) über zwei Semester vermittelt. Weil wir Konzepte der Vorlesung um praktische Erfahrungen ergänzen wollten, wurde jeweils zur Vorlesung eine zweistündige Übung angeboten. Die Weiterentwicklung des Objektorientierung allgemein und unseres Ansatzes hatte zur Folge, daß Vorlesung und Übung jedes Jahr überarbeitet werden mußten - teils erheblich. Die Bilanz aus fünf „traditionellen“ Durchgängen stellt sich für uns so dar:

- Ungefähr 50 bis 70 Studierende besuchten die Vorlesung am Anfang. Gegen Ende des zweiten Semesters waren noch etwa 35 bis 40 regelmäßige Teilnehmer übrig. Die Übungen wurden anfangs von etwa 30, gegen Ende von weniger als 15 Studierenden besucht.
- Die Diskussionen in Vorlesungen beschränkten sich auf einzelne Studierende. Der Rest legte eher eine interessierte Konsumhaltung an den Tag.

Die Mitarbeit in den Übungen war nur bei wenigen Studierenden als intensiv zu bezeichnen.

- Obwohl die Lehrveranstaltung von den Studierenden immer positiv bewertet wurde, hatten die Lehrenden (etwa in Prüfungen) den Eindruck, daß nur wenig von den angebotenen Lehrinhalten wirklich dauerhafte Wirkung hinterlassen hatte.

Wir haben deshalb die gesamte Lehrveranstaltung grundlegend umgebaut:

Zu Beginn jedes Semesters findet ein dreitägiges Kompaktseminar statt. Dort wird in kurzen Vorträgen (max. 60 Minuten) ein erster thematischer Überblick vermittelt. Jedem Vortrag folgt eine Gruppenübung, deren Ergebnisse anschließend im Plenum von den Gruppen vorgetragen und von allen diskutiert werden. Dies ergibt den ersten Rückkopplungskreis.

Die eigentliche Semesterarbeit läuft als Projekt. Auf der konzeptionellen Basis des einleitenden Kompaktseminars wird eine praktische Entwicklungsaufgabe von jeder studentischen Gruppe erarbeitet. Diese Aufgabe hat insofern Projektcharakter als sie über ein Jahr analysierende, entwerfende und konstruktive Tätigkeiten bis hin zu einem lauffähigen Prototyp umfaßt und mit „echten“ Anwenden erledigt wird. Während der Projektarbeit finden regelmäßige Besprechungen der einzelnen Teams mit Betreuern statt. Durch die Projektarbeit werden die Konzepte und Techniken, die im Kompaktseminar erstmals vorgestellt wurden, selbst intensiv angewendet und in den Gruppengesprächen reflektiert. Dies stellt den zweiten Autor-Kritiker-Zyklus dar.

Am Ende eines Semesters werden die Ergebnisse wieder von den Projektteams in einem dreitägigen Review vorgestellt. Dieses Review ist zweistufig. Zunächst findet ein Peer-Review statt. Dies bedeutet, daß sich die Studentengruppen anhand vorbereiteter Unterlagen und einer längeren Präsentation wechselseitig unter - Moderation der Dozenten begutachten. In den zweiten Teil der Diskussion zu jeder Gruppe lassen dann die Dozenten vertiefende Konzepte, grundlegende Problemstellungen und ihre Lösungen einfließen. In diesem abschließenden Kompaktseminar werden nicht nur bekannte Inhalte wiederholt, sondern es werden weitergehende Lösungen für Probleme angeboten, die erst durch die Projektarbeit im Semester erfahrbar geworden sind. In diesem Sinne wird das inhaltliche Angebot direkt durch Forschungsthemen, die Anforderungen der Studierenden und durch die aktuelle Projektsituation bestimmt.

Derzeit nehmen ca. 30 Studierende an den Veranstaltungen OOSE I & II sowie dem begleitenden Projekt teil. Die Zahl nimmt im zweiten Semester nur geringfügig ab. Betreut werden die Studierenden vom Leiter der Veranstaltung sowie einem wissenschaftlichen Mitarbeiter in den Plenarsitzungen sowie in separaten Beratungsterminen mit den einzelnen Teams. Im letzten Durchgang wurde den Studierenden angeboten, daß sie freiwillig neben Java als gewählter -

Implementierungssprache noch C++ erlernen konnten. Dazu haben die Dozenten die Unterlagen eines fertigen C++-Kurs bereitgestellt und einen Studenten, der schon umfangreiche C++ Programmierkenntnisse hatte, bei der Einarbeitung in das Kursmaterial unterstützt. Auf dieser Basis fanden an Wochenenden und in einer Ferienwoche zwei gut besuchte C++ Kurse für die Teilnehmer statt. Dadurch wurden die Studierenden in die Lage versetzt, Prototypen zu bauen.

4.2 JWAM: Rahmenwerke & Entwurfsmuster in der Lehre

Die „JWAM“ Veranstaltung bildet das Forum für die an der Konstruktion von Rahmenwerken interessierten Studierenden und wird von ca. 20 Personen besucht.

Rahmenwerke und Entwurfsmuster als Thema stellen Lehrende in der Softwaretechnik vor ein prinzipielles Problem: Der Anwendungsbereich für diese Techniken läßt sich nur vor dem Hintergrund einer vergleichsweise großen Programmiererfahrung und Aufgabenstellung motivieren. Die wenigsten Studierenden verfügen über diesen Hintergrund. Selbst wenn sie bereits berufstätig sind, werden solche Themen noch nicht berücksichtigt.

Andererseits zeigt sich bei praktischen Studien- und Diplomarbeiten, daß die Studierenden gerne auf ein Rahmenwerk aufbauen würden. Rahmenwerke sind nützlich, weil darin grundlegende Dienste, wie Behälterklassen, Metaobjektprotokoll, Fenstersystemanschluß, Abstraktion von der Middleware für die Verteilung, Werkzeugkonstruktion, Desktop und einen Datenbankanschluß zur Verfügung gestellt sind. So können sich die Studierenden auf das eigentliche Thema ihrer Arbeit konzentrieren. Nicht zuletzt ist die (Weiter-)Entwicklung von Rahmenwerken ein wichtiges Forschungsgebiet.

Deshalb haben wir Rahmenwerke für die Entwicklung interaktiver und eingebetteter Anwendungssysteme gebaut und weiterentwickelt (s.[LRW+97]). Seit 1½ Jahren arbeiten wir verstärkt an einem Java-Rahmenwerk, das bisher in zwei Versionen vorliegt (s.[FLL+98]).

Für Java in der Lehre sprechen aus unserer Sicht mehrere Gründe:

- Plattformunabhängigkeit: Java bietet die Möglichkeit, das Rahmenwerk sowohl in der Universität als auch zu Hause einzusetzen.
- Vollständigkeit: Das Java Development Kit (JDK) enthält standardmäßig Bibliotheken zur Anbindung von Benutzungsoberflächen, Dateizugriff und andere Basisdienste, auf die einfach und schnell zugegriffen werden kann.
- Beherrschbarkeit: Java ist leichter zu erlernen als C++.
- Praxisrelevanz: Java liegt heute im Trend von Forschung und Wirtschaft. Wir gehen davon aus, daß die Studierenden das Wissen um Java auch beruflich einsetzen können.

- Verfügbarkeit: Für Java sind sehr viele Bücher, Informationen, Online-Tutorials und Programmierertools verfügbar. Diese sind meistens kostenlos.
- WWW-Applikationen: Mit Java können besonders leicht Internetanwendungen konstruiert werden.

An unserem Arbeitsbereich hat sich eine Rahmenwerksgruppe aus wissenschaftlichen Mitarbeitern und Studierenden etabliert, die sich regelmäßig trifft. Seit dem WS 98/99 ist der Gruppe ein Seminar angegliedert. Im Semester trifft sich die Rahmenwerksgruppe wöchentlich für mindestens zwei Stunden, bei Bedarf auch länger; in der vorlesungsfreien Zeit finden alle zwei bis drei Wochen Sitzungen statt. Die Sitzungen werden von einer aus der Gruppe gewählten zweiköpfigen Leitung vorbereitet und moderiert. Es hat sich bewährt, daß die Gruppe von einem wissenschaftlichen Mitarbeiter und einem Studierenden gemeinsam geleitet wird.

Ziel der Aktivität ist, die einzelnen von Studierenden im Rahmen ihrer Arbeiten bearbeiteten Konzepte koordiniert in ein Rahmenwerk einfließen zu lassen. Von daher liegt das Hauptaugenmerk weniger auf den einzelnen bearbeiteten Themenkomplexen als vielmehr auf übergeordneten Belangen wie Dokumentationsstandards, dem Verhältnis von Rahmenwerkskomponenten zueinander, der zugrundeliegenden Architektur sowie dem Projektmanagement und der Teamarbeit. Die Studierenden lernen dabei ihre Arbeit in einen größeren Kontext einzuordnen. So erfahren die Studierenden die Probleme und Besonderheiten des Projektmanagements und der Teamarbeit „am eigenen Leib“. Die Mitarbeiter sehen ihre Rolle auch darin, neue Konzepte und Problemlösungen zu vermitteln, wenn der Bedarf besteht. Auf diese Weise können Studierende neue Techniken und Konzepte „an Ort und Stelle“ ausprobieren.

Neben der Leitung der Gruppe gibt es weitere Funktionen, die größtenteils von Studierenden übernommen werden. So werden z.B. nach Themen geordnet Diskussionen und Architekturentscheidungen im Web dokumentiert (je Thema gibt es einen „verantwortlichen“ Protokollanten). Es gibt Zuständige für die Web-Darstellung des Rahmenwerks, Infrastruktur, für Styleguides, Dokumentation incl. Reviews, und Veröffentlichungen. Den Studierenden wird durch diese Aufteilung bewußt, daß Softwareentwicklung in einem größeren Rahmen mehr als nur Programmieren ist, und sie lernen, bei Übernahme einer dieser Zuständigkeiten Verantwortung zu übernehmen.

Insgesamt können die Studierenden dieser Veranstaltung, das ihnen vermittelte Wissen anhand eines größeren konkreten Beispiels anwenden. Sie erfahren dabei den erhöhten Koordinationsaufwand größerer Projekte und die vielfältigen zusätzlichen Aufgaben, die sich dadurch ergeben. Unter den Veranstaltungsteilnehmern finden sich häufig kleinere Gruppen von Studierenden, die an ähnlichen Themen arbeiten. In diesen kleineren Gruppen können die Themen dann bis ins Detail diskutiert werden.

Wir verbinden noch weitere Themen und Ziele mit dem Einsatz des JAVA-Rahmenwerks (JWAM-Rahmenwerks) in Forschung und Lehre:

- **Verwendung:** Die Anwendung von Rahmenwerken kann am JWAM-Rahmenwerk geübt werden. Dabei müssen sich die Studierenden mit unterschiedlichen Konzepten befassen, die in den verschiedenen Rahmenwerkkomponenten realisiert worden sind, um einen Überblick über das Zusammenspiel dieser Komponenten zu gewinnen.
- **Entwicklung:** Die Weiterentwicklung von Rahmenwerken kann am JWAM-Rahmenwerk geübt werden. Durch seine Verwendung bekommen die Studierenden unmittelbar Rückkopplung über ihre Arbeit. Sie können ihre Ergebnisse in einer größeren Anwendergruppe diskutieren und so im engen Autor-Kritiker-Zyklus überprüfen.
- **Schichtenarchitektur:** Das JWAM-Rahmenwerk ist gemäß der WAM-Schichtenarchitektur (vgl. [BGK+ 97]) aufgebaut. Es hat sich insbesondere gezeigt, daß die Schichtenarchitektur dann „lebendig“ wird, wenn versucht wird, neue Komponenten in die Schichten einzuordnen.
- **Umgang mit großen Systemen:** Das JWAM-Rahmenwerk hat zwar nicht den Umfang, den Anwendungssysteme heute in großen Unternehmen haben, es ist jedoch deutlich größer als die sonst üblichen Beispielprogramme. Das JWAM-Rahmenwerk hat inzwischen eine Größe erlangt, die es nicht mehr erlaubt, daß jeder Beteiligte des komplette Rahmenwerk gut kennt und vollständig versteht.
- **Entwurfsmuster:** Im JWAM-Rahmenwerk kommen eine Reihe von Entwurfsmustern zum Einsatz. Hier kann demonstriert werden, wie Entwurfsmuster modifiziert werden, um spezielle Anforderungen zu erfüllen und wie verschiedene Entwurfsmuster miteinander kooperieren.
- **Teamarbeit und Projektmanagement:** Durch die Größe des JWAM-Rahmenwerks kann es nur von einem Team weiterentwickelt werden. So werden Aspekte der Teamarbeit, Koordination und Kooperation, Management und Planung für die Beteiligten deutlich gemacht. Wir sehen hier einen wichtigen Beitrag zur Forschung. Schließlich werden Aspekte des Projektmanagements an Universitäten häufig wenig betrachtet, da keine geeigneten Demonstrationsobjekte zur Verfügung stehen. Da mit dem JWAM-Rahmenwerk tatsächlich ein länger lebendes Produkt hergestellt wird, bekommen diese Aspekte auch für die universitäre Lehre eine größere Bedeutung.
- **Studien-/Diplomarbeiten:** Heute gehen Studien- und Diplomarbeiten in der Konstruktion oft über das hinaus, was früher üblich war. Dazu müssen diese Arbeiten auf häufig vorhandenen Diensten aufsetzen. Wie sich an ver-

schiedenen laufenden Arbeiten gezeigt hat, ist JWAM dazu geeignet. Die Ergebnisse der Arbeiten gehen wieder ins Rahmenwerk ein und werden so für andere Studierende verfügbar. Dadurch sind die Ergebnisse von Studien- und Diplomarbeiten mehr als Prototypen und kommen in den Bereich "echter" Softwareentwicklung. Da die eigenen Ergebnisse von anderen verwendet werden, steigt auch die Motivation der Beteiligten.

- **Forschung:** Die Architektur von Rahmenwerken, die Prozesse zur Rahmenwerksverwendung und -entwicklung sowie die zur Weiterentwicklung sinnvollen Organisationsstrukturen sind Gegenstand unserer wissenschaftlichen Arbeit.
- **Dokumentation:** Durch die Langlebigkeit des JWAM-Rahmenwerks müssen besonders hohe Anforderungen an die Qualität der Dokumentation gestellt werden. Die Studierenden erfahren, welche Ansprüche an Dokumentation erfüllt werden müssen und auch, welche Auswirkungen unzureichende Dokumentation für die Verständlichkeit hat.

Im ganzen bildet das JWAM-Rahmenwerk also eine geeignete Basis für eine Reihe von Lernzielen, denen wir uns heute stellen müssen. Nachteilig ist bisher, daß heute noch kein großes Anwendungssystem auf Grundlage des Rahmenwerks existiert. Dies ist vor allem für die Weiterentwicklung des Rahmenwerks problematisch. Anwendungen, die im Rahmen der oben beschriebenen Lehrveranstaltungen erstellt werden, können eine bestimmte Größe nicht überschreiten. Dies reicht nicht aus, um das JWAM-Rahmenwerk mit professionellem Anspruch weiterzuentwickeln. Daher haben wir begonnen, das JWAM-Rahmenwerk zu professionalisieren und in die industrielle Praxis zu tragen. In diesem Zuge erhoffen wir uns Anregungen für die weitere Entwicklung des Rahmenwerks.

Die beschriebenen Vorteile sind jedoch nicht ohne Aufwand zu haben. Von Seiten der Studierenden und Mitarbeiter ist ein hoher Einsatz und viel Engagement notwendig, um das JWAM-Rahmenwerk weiterzuentwickeln. Zudem hat sich gezeigt, daß der Aufwand für die Vermittlung des Rahmenwerks an die Studierenden sehr hoch ist. Da viele Fragen und Probleme erst bei vertiefter Arbeit mit dem Rahmenwerk auftreten, reicht es nicht aus, in Blockveranstaltungen das Rahmenwerk zu vermitteln. Außerdem muß immer ein Ansprechpartner für auftretende Fragen und Probleme bereitstehen. Die speziell hierfür eingerichtete Newsgroup wird sehr selten in Anspruch genommen wird; dafür müssen aber die Entwickler des Rahmenwerkes häufig persönlich Rede und Antwort stehen. Andererseits zahlt sich die intensive Beschäftigung mit Fragen der Rahmenwerkskonstruktion auch in einer effizienten Vorbereitung von Lehrveranstaltungen aus.

Erst in der Zukunft wird sich zeigen, ob das JWAM-Rahmenwerk längerfristig "überleben" und die hohe Fluktuation von Studierenden und Mitarbeitern kompensiert werden kann. Erste Erfahrungen sprechen dafür: Derzeit entwickeln ca.

10 Studierende und 3-4 Mitarbeiter das Rahmenwerk. Die Studenten bleiben ½ bis 1½ Jahre in der Gruppe, während die wissenschaftlichen Mitarbeiter aufgrund zeitlich befristeter Arbeitsverträge etwa die doppelte Zeitspanne an der Entwicklung teilnehmen können. Bei gleichmäßiger Fluktuation kann somit eine kontinuierliche Weiterentwicklung erreicht werden. Die Nutzung des JWAM-Rahmenwerks in industriellen Projekten kann als weiterer Mechanismus gesehen werden, die Entwicklung längerfristig zu stabilisieren.

5 Fazit

Die Erfahrungen mit der „reformierten“ OOSE-Lehrveranstaltung sind fast durchgängig positiv. Fast alle Studierenden haben betont, daß die Lehrveranstaltung durch ihre Form und Inhalte überzeugen konnte. Hervorgehoben wurde, daß durch die eigene intensive Beschäftigung mit einer praxisnahen Aufgabe viele Konzepte und Techniken erst verständlich und in ihrem Stellenwert für die Softwareentwicklung deutlich wurden. Die zeitlich gestraffte Vermittlung von Grundkonzepten und die anschließende Freiheit, die Arbeit im Semester an die eigenen Bedürfnisse und Möglichkeiten anzupassen kamen der persönlichen (und beruflichen) Situation der Studierenden sehr entgegen. Die Prinzipien der anwendungsorientierten Softwareentwicklung konnten mit Leben gefüllt werden und fanden Anklang. Einige Arbeitsgruppen waren am Ende recht weit von ihrem selbstgesteckten Ziel entfernt. Obwohl das Ergebnis aus technischer Sicht dort eher enttäuschend war, wurde die Lehrveranstaltung auch von diesen Studierenden positiv bewertet. Sie analysierten ein halbwegs gescheitertes Projekt und fanden Gründe und Zusammenhänge, die aus unserer Erfahrung als praxisnah bezeichnet wurden.

Die Arbeit an einem für eine Universität recht großen Rahmenwerk, das zugehörige Seminar und die Rahmenwerk-Gruppe haben den Lehrenden und Lernenden sehr geholfen, theoretisches Wissen über Entwurfsmuster und Rahmenwerke technisch umzusetzen. Dadurch wird die Entwicklung von komplexen Systemen, Teamarbeit, Projektmanagement und die Arbeit in einem größeren Kontext möglich und erfahrbar.

Aus Sicht der Lehrenden sind die neuen Lehrveranstaltungen sicherlich befriedigender als die traditionellen. Die intensiven Diskussionen und die Vielfalt der Ergebnisse gehen über das hinaus, was bisher möglich war. Wir haben den Eindruck, daß Konzepte und Techniken wesentlich intensiver erarbeitet wurden als in den Vorjahren. Allerdings erfordert die neuen Lehrveranstaltungsformen auch einen deutlich intensiveren Einsatz. Ohne gute Vorbereitung und Konzentration lassen sich in den Blockveranstaltungen und dem JWAM-Seminar die vielen vorgetragenen Ergebnisse und Ideen nicht schnell genug einordnen und mit weiterführenden Konzepten verbinden.

6 Literatur

- [BGK+ 97] D. Bäumer, G. Gryczan, R. Knoll, C. Lilienthal, D. Riehle, H. Züllighoven: Framework Development for Large Systems. Communications of the ACM, October 1997, Vol. 40, No. 10, S. 52–59.
- [LRW+97] C. Lilienthal, S. Roock, U. Weiss, H. Wolf: *C++-Framework*. Universität Hamburg, Fachbereich Informatik, Arbeitsbereich Softwaretechnik,. <http://swt-www.informatik.uni-hamburg.de/Software/BibV30>, 1997.
- [CACM97] Communications of the ACM, Themenheft *Electronic Performance Support Systems*, 40 (1997) 7, S. 31 – 63.
- [FZ97] C. Floyd, H. Züllighoven: *Softwaretechnik*. In: P. Rechenberg, G. Pomberger (Hrsg.): Informatik Handbuch. München, Wien: Carl Hanser Verlag, 1997, S. 641–667.
- [FLL+98] N. Fricke, C. Lilienthal, M. Lippert, S. Roock, H. Wolf: *Java Framework*. Universität Hamburg, Fachbereich Informatik, Arbeitsbereich Softwaretechnik. <http://swt-www.informatik.uni-hamburg.de/Software/JWAM>, 1998.
- [Jac95] M. Jackson: *Software Requirements and Specifications - a lexicon of practice, principles and prejudices*. Wokingham, Addison-Wesley, 1995.
- [Leh90] M.M. Lehman: *Uncertainty in Computer Applications is Certain*. In: Proceedings of the 1990 IEEE International Conference on Computer Systems and Software Engineering, IEEE, Tel Aviv, May 1990.
- [WKKL98] I. Wetzel, R. Klischewski, A. Krabbel, C. Lilienthal: *Kooperation für Software für Kooperation: Erfahrungen aus einem partizipativen Softwaretechnikprojekt*, In: Volker Claus (Hrsg.) Informatik und Ausbildung GI-Fachtagung 98 Informatik und Ausbildung Stuttgart, 30. März bis 1. April 1998; Springer-Verlag 1998 (Reihe: Informatik aktuell)
- [Zül98] H. Züllighoven: *Das objektorientierte Konstruktionshandbuch nach dem Werkzeug & Material-Ansatz*. Heidelberg: dpunkt-Verlag, 1998.