Projektbezogenes Lernen objektorientierter Techniken

Maritta Heisel, Stephan Herrmann, Wilfried Koch, Peter Rüppel

Technische Universität Berlin*

21. Februar 1995

Zusammenfassung

Studierende der Informatik an der TU Berlin können auf ein breites Lehrangebot zum Erlernen objektorientierter Techniken zurückgreifen. Aufbauend auf einer allgemeinen Softwaretechnik-Vorlesung können objektorientierte Programmierung in der Sprache Eiffel sowie verschiedene objektorientierte Methoden und Werkzeuge erlernt werden. Während ein solches Angebot inzwischen zum universitären Standard gehören dürfte, sind *studentische Projekte* eine Berliner Besonderheit. In diesen Projekten erfolgt eine Integration aller Arbeitsschritte und und Rollen innerhalb der objektorientierten Softwareentwicklung. Objektorientierte Techniken können auf diese Weise viel tiefergehend erlernt werden als mit anderen Veranstaltungsformen.

1 Das objektorientierte Lehrprogramm an der TU Berlin

Bevor wir genauer auf die Konzeption und Durchführung studentischer Projekte eingehen, geben wir einen Überblick über die Veranstaltungen, in denen an der TU Berlin objektorientierte Techniken gelehrt werden. Erfahrungen aus den studentischen Projekten, ihre Auswirkungen auf die Forschung sowie ein Ausblick beschließen unsere Ausführungen.

Die einzelnen Veranstaltungen sowie ihr Zusammenhang sind in Bild **??** gezeigt.

Softwaretechnik und Systemgestaltung. Dies ist die Basisveranstaltung, die die Grundlagen der Softwaretechnik vermittelt. Hier werden sowohl traditionelle als auch objektorientierte Ansätze betrachtet und kontrastiert. Ausgehend von der Theorie abstrakter Datentypen werden die grundlegenden objektorientierten Konzepte eingeführt und ihre Anwendung anhand von Beispielen demonstriert. Da diese Veranstaltung Überblickscharakter hat, ist es hier nicht möglich, eine objektorientierte Programmiersprache zu erlernen oder auf objektorientierte Analyse und Entwurf genauer einzugehen. Die gebotenen Konzepte und Ideen sind aber ausreichend, um in vielen Studierenden ein tieferes Interesse an objektorientierten Techniken zu wecken und sie zum Besuch der weiterführenden Veranstaltungen zu motivieren.

*Forschungsgruppe Softwaretechnik (FR5-6), Franklinstr. 28/29, D-10587 Berlin, Germany.

Objektorientierte Softwareentwicklung und Programmierung. Grundlage dieser Veranstaltung ist Bertrand Meyers Buch „Object-oriented Software Construction“. Obwohl das Hauptaugenmerk dieses Buches auf der Präsentation der Programmiersprache Eiffel liegt, wird der Motivation und Erkärung grundlegender objektorientierter Konzepte breiter Raum gewidment. Dies kommt unserem Ziel entgegen, in dieser Veranstaltung nicht nur die Programmiersprache Eiffel zu vermitteln, sondern insgesamt das Denken in objektorientierten Kategorien zu lehren. Im Vorlesungsteil der Veranstaltung wird großer Wert auf die Präsentation von Fallstudien und den Vergleich zu anderen objektorientierten Sprachen wie Smalltalk, C++ oder Simula 67 gelegt. Im praktischen Teil der Veranstaltung werden Programme mittlerer Größe entwickelt; im Wintersemester 94/95 war dies beispielsweise ein Transaktionssystem für Banken.

Obwohl diese Veranstaltung gegenüber der Basisveranstaltung eine wesentliche Vertiefung darstellt, so ist es trotzdem nicht ihr Anspruch, objektorientierte Techniken vollständig abzudecken. Stattdessen wird der Schwerpunkt auf objektorientiertes Programmieren gelegt.

Methoden und Werkzeuge zur Softwareproduktion. Diese Lehrveranstaltung hat den Zweck, Methoden, Verfahren und technische Werkzeuge der Softwaretechnik zu behandeln, ohne deren prinzipielle Existenzberechtigung im Softwareproduktionsprozeß zu begründen. Die Darstellung ist dabei auf eine breite Präsentation der gesamten methodischen Palette ausgerichtet. Objektorientierte Methoden spielen eine dominierende Rolle, weil sie zu den innovativsten gehören und deshalb bei den Studierenden auch auf größtes Interesse stoßen. Von allen Teilnehmern werden Grundkenntnisse objektorientierter Sprachen vorausgesetzt. Deshalb kann der Schwerpunkt auf objektorientierte Analyse- und Design-Methoden und die zugehörige Werkzeugunterstützung gelegt werden. Derzeit etwa werden die Methoden von *Coad und Yourdon*, *OMT*, *Booch*, *Jacobson (Use case driven approch)* und die *Fusion method* behandelt.
In der Veranstaltung wird versucht, thematisch zusammengehöriges präsentiertes Material gegenüberzustellen und, wenn möglich, zu werten. Es hat sich bewährt, eine “integrierte Veranstaltung” anzubieten. Dies heißt, daß die Veranstalter nur an bestimmten Interaktionspunkten Vorlesungen halten oder daß zentrale Diskussionen ein thematisch zusammenhängendes Gebiet abschließen. Der Rest besteht aus Vorträgen der Teilnehmer und – soweit möglich – aus Übungen mit den präsentierten Werkzeugen. Viele Teilnehmer arbeiten bereits in Softwareprojekten der zahlreichen kleineren Firmen des Berliner Umfeldes und bringen eigene Erfahrungen mit Methoden und Werkzeugen ein, die sonst an der Universität nicht verfügbar wären. Von den Veranstaltern wird dieser Trend bewußt dadurch gefördert, daß Vertreter der sog. Praxis gezielt zu gelegentlichen Vorträgen eingeladen werden.

2 Das Software Engineering Projekt

Diese Lehrveranstaltung bildet den Abschluß unseres objektorientierten Lehrzyklus an der TU Berlin. Die Teilnehmer sollten zumindest die grundlegenden Techniken innerhalb der Softwareentwicklung überschauen (Lehrziel der Basisveranstaltung) und die Grundkonzepte der Objektorientierung wenigsten in einer objektorientierten Programmiersprache anwenden können (Lehrziel der Veranstaltung ’objektorientierte Softwa-

reentwicklung').

Tiefergehende Kenntnisse im Themenbereich der Methoden und Werkzeuge werden nicht vorausgestetzt, da

- wir zu Beginn der Veranstaltung eine Einführung in die einzusetzende objektorientierte Analyse- und Entwurfsmethode geben und
- Werkzeuge i.d.R. zu wenig standardisiert sind, um hier vorab eine sinnvolle Festlegung treffen zu können.

Bezüglich der Analyse- und Entwurfsmethode sollen also vielmehr praktische Erfahrungen beim Einsatz der Methodig gewonnen werden anstatt mehrere Methoden vergleichend gegenüber zu stellen.

Bezüglich der einzusetzenden Entwicklungswerkzeuge sollen die Studierenden dazu angeleitet werden

- Werkzeuge auszuwählen und
- sich teils selbständig, teils unter Anleitung der Lehrveranstalter oder anderer Studierenden in ein Werkzeug einzuarbeiten.

Neben diesen rein softwaretechnischen, produktorientierten Lehrzielen sollen aber auch die prozeßorientierten Anteile in der Softwareentwicklung für die Teilnehmer der Lehrveranstaltung erfahrbar gemacht werden. Unter prozeßorientierten Tätigkeiten verstehen wir:

Projektmanagement: Das Festlegen und die Überwachung von Zuständigkeiten von einzelnen Personen zu bestimmten Aufgaben innerhalb des Projektteams. Auch Erstellung einer Zeitplanung fällt unter diesen Tätigkeitsbereich.

Qualitätssicherung: Das Festlegen und die Überwachung von Qualitätsmerkmalen aller Zwischen- und Endprodukte durch *Reviews, Walktroughts* in den Analyse- und Entwurfstätigkeiten;
die Überwachung der prozessorientierten Tätigkeiten selbst anhand von *Audits* und
Überprüfung der implementierten Klassen durch dynamisches Testen.

Produktverwaltung: Aufbauen einer Projektbibliothek und Bereitstellung von Mechanismen zur Sicherung ihrer Konsistenz.

Objektorietierte Softwareentwicklung bietet hier, im Vergleich zu traditionellen Entwicklungsansätzen, bzgl. der Durchführug solcher prozeßorientierten Tätigkeiten bessere Voraussetzungen zum Einsatz von zyklischen, iterativen Vorgehensmodellen, da in allen Entwicklungsaktivitäten mit denselben Konzepten (Klasse, Vererbung, Polymorphie) gearbeitet wird.

2.1 Durchführung

Die Veranstaltung erstreckt sich über zwei Semester (Projekt I + II). Zu Beginn der Veranstaltung werden die Teilnehmer auf verschiedene *Entwicklungsteams* von maximal

zwölf Studierenden aufgeteilt. Jedes Team bearbeitet unter Anleitung eines Lehrveranstalters innerhalb der beiden Semester die gleiche Aufgabenstellung. Im Falle der letzten beiden Projekten bestand diese in der *Entwicklung eines Informationssystems für eine Hausmüll Entsorgungsanlage* .

Die einzelnen Entwicklungsteams treffen sich zur Bearbeitung der Aufgabenstellug jeweils zwei Mal pro Woche. Außerdem sind pro Woche alle Teilnehmer des Projektes in einem Plenum versammelt, innerhalb dessen:

- die methodischen und technischen Rahmenbedingungen für alle Entwicklungsteams gelegt.
  Zum Beispiel fiel bei den zuletzt durchgeführten Projekten hierunter eine Einführung in die Methode von *Coad & Yourdon*.

- die einzelnen Entwicklungsteams zu vorab festgelegten *Meilensteinen* den Stand ihrer Arbeit dokumentieren.
  Z.B. stellen die Teams hier ihre Analyse- und Entwurfsergebnisse oder Videopräsentationen von Prototypen und Versionen des Softwaresystems vor.

- Teilnehmer durch *Referate* bestimmte Entscheidungen in den Entwicklungsteams vorbereiten.
  Z.B. werden an der TU Berlin verfügbare, objektorientierte Programmiersprachen mit ihrer Programmierumgebung durch Referate vorgestellt, damit in den einzelnen Teams dann entschieden werden kann, mit welcher Sprache implementiert werden soll.

Innerhalb des ersten Semesters (Projekt I), werden zunächst die funktionellen Rollen Projektmanagement, Qualitätssicherung, Produktverwaltung und Entwicklung auf einzelne Personen verteilt. Dabei wird von seiten der Lehrveranstalter darauf geachtet, daß jeder Teilnehmer sowohl eine prozeßorientierte als auch eine produktorientierte Rolle (daher die Rolle Entwickler) einnimmt. Nachdem der organsatorische Rahmen zur produktorientierten Arbeit gelegt ist, wird zunächst unter Zuhilfenahme der Methode von *Coad & Yourdon* der Gegenstandsbereich analysiert und schrittweise in einen Entwurf des Programmsystems überführt.

Spätestens mit Beginn des zweiten Semesters (Projekt II), soll der Schwerpunkt der Tätigkeiten innerhalb der Entwicklungsteams sich weg von Analyse und Entwurf hin zu Entwurf und Implementierung verlagern.

Zum Abschluß des Projektes findet ein öffentlich zugängliches Kolloquium statt, bei dem die einzelnen Entwicklungsteams jeweils die letzte Version ihres Softwaresystems per Videopräsentation vorstellen und ein Resümee unter die von ihnen geleistete Arbeit setzten.

2.2 Erfahrungen

Mit der allgemeinen *Konzeption der Lehrveranstaltung* wurden gute Erfahrungen gesammelt. Positiv aufgefallen ist vor allem die sehr gute Motivation der Teilnehmer, für die wir folgende Ursachen sehen:

- Freiräume, wie die Wahl der Implementierungssprache, schufen ein besonderes Klima der Eigenverantwortlichkeit innerhalb der einzelnen Entwicklungsteams.

- Vorab bestimmte Meilensteine, bei denen die Entwicklungsteam ihre Arbeit präsentieren mußten, führten zu einer anspornenden Konkurrenzsituation zwischen den unterschiedlichen Teams.

Als Problematisch erwies sich jedoch bei allen hier durchgeführten Projekten die *Umsetzung von qualitätssichernden Maßnahmen.* Vor allem die Testaktivitäten wurden im Laufe der Implementierung gern vernachlässigt:

- Die Teilnehmer schienen die Notwendigkeit zum Test ihrer Klassen oft zu unterschätzen.

- Das Testen von objektorientierten Programmen verlangt neue Teststrategien, die Konzepte wie Datenkapselung, Vererbung, Polymorphie und Generizität berücksichtigen. Solche Verfahren und Techniken sind aber zur Zeit erst im Entstehen.

- Testen verlangt in besonderem Maße nach Werkzeugunterstützung, die aber kaum verfügbar ist.
  Ohne adäquate Werkzeuge zur Erzeugung, Ausführung und Auswertung von Testfällen geraten Testaktivitäten schnell zu 'Sisyphusarbeit'.

Generell ließ sich bezüglich der eingesetzten Werkzeuge immer wieder feststellen:

- Projektteams, die C++ als Programmiersprache einsetzten, waren bei der Implementierung langsamer als Teams die Eiffel verwandten[1].
  Der Grund hierfür schien einerseits in der höheren Komplexität von C++ zu liegen und andererseits in der stärkeren Heterogenität der unter C++ eingesetzten Klassenbibliotheken.

- Der Einarbeitungsaufwand entscheidet über den Einsatz von Werkzeugen.
  Werkzeuge mit einer einfach zu bedienenden, graphischen Oberfläche wurden wesentlich leichter angenommen, als Kommandozeilen orientierte Werkzeuge.

- Werkzeuge zur Unterstützung von objektorientierten Analyse- und Designmethoden sind an der TU Berlin aus finanziellen Gründen kaum verfügbar.

Vor allem der letzte Punkt bedingte immer wieder große Probleme bei der Wahrung der *Konsistenz zwischen Entwicklungsdokumenten* aus früheren Aktivitäten, da Revisionen in späten Phasen oft manuelle Änderungen durch alle vorangegangenen Dokumente erforderlich machten.

[1] Interessanterweise haben sich in den letzten Projekten alle Entwicklungsteams für C++ oder Eiffel entschieden.