Ausbildung in der Softwaretechnik an der Universität Hamburg

Heinz Züllighoven
Arbeitsbereich Softwaretechnik, Fachbereich Informatik, Universität Hamburg

1 Der konzeptionelle Rahmen

Der Arbeitsbereich Softwaretechnik an der Universität Hamburg wird von Christiane Floyd und dem Autor geleitet. Die Gruppe vertritt einen *anwendungsnahen und menschenzentrierten Ansatz* (vgl. [Floyd94]) und befaßt sich primär mit Entwurf und Konstruktion, Auswahl, Einsatz und Weiterentwicklung von interaktiver Software zur Unterstützung von qualifizierten Aufgaben in Wirtschaft und Verwaltung. Aktuell werden zwei Schwerpunkte bearbeitet: *Softwareentwicklung in Organisationen* und *objektorientierte Softwareentwicklung*.

2 Stellenwert im Studium

Die Themen der Softwaretechnik werden an der Universität Hamburg im Grund- und Hauptstudium behandelt. Im Grundstudium stehen die Grundlagen der Softwareentwicklung von der Analyse über den Entwurf bis zur Konstruktion im Vordergrund. Dazu wurden Fragen der Modellbildung und der Semiotik behandelt. Bisher wurden die funktionale und die imperative Programmierung in jeweils siebenstündigen Lehrveranstaltungen vermittelt. Dazu kommt ein vierstündiges Programmierpraktikum. Diese Lehrveranstaltungen werden von 250 bis 300 Studierenden besucht. Im Rahmen der Vordiplomsprüfung wurden diese Lehrveranstaltungen mündlich geprüft.

Im Hauptstudium ist die Softwaretechnik bisher einer von sechs Themenbereichen der Praktischen Informatik und ein mögliches Vertiefungsgebiet. Da vier von sechs Themenbereichen der Praktischen Informatik Bestandteil der Diplomprüfung sind, hat die sog. Kerngebietsvorlesung *Einführung in die Softwaretechnik* einen hohen Stellenwert im Hauptstudium. Wird Softwaretechnik als Vertiefungsgebiet gewählt, so sind aus diesem Bereich 12 Semesterwochenstunden nachzuweisen, davon 4 SWS theoretische Vertiefung.

Die gegenwärtige Studienordnung sieht noch keine studienbegleitenden Prüfungen im Hauptstudium vor. Die Inhalte der Lehrveranstaltungen werden entsprechend nur ein einer Hauptdiplomprüfung im Kern- und Vertiefungsfach geprüft.

3 Die Veranstaltungen

Der folgende Überblick zeigt die Basis des Softwaretechnik-Curriculums, also die Veranstaltungen, die in den letzten Jahren regelmäßig angeboten wurden. Entsprechend sind Lehrveranstaltungen mit wechselnden Themen, die ergänzenden oder vertiefenden Charakter haben, nicht aufgeführt. Anschließend werden einzelne Veranstaltungen des Curriculums unter der Frage betrachtet, welchen Einfluß die veränderte Softwaretechniklandschaft auf die Lehre hat.

3.1 Einführung in die Softwaretechnik

Die vierstündige Lehrveranstaltung „Einführung in die Softwaretechnik“ ist bisher die Basis der softwaretechnischen Ausbildung im Hauptstudium. Dort werden in der Vorlesung die grundlegenden Themen der Softwaretechnik wie Methoden und Werkzeuge, Architekturkonzepte sowie Aktivitäten und Vorgehen in Softwareprojekten vorgestellt (vgl. [FZ97]). Neben den traditionellen Konzepten steht die evolutionäre durch Prototyping unterstützte Vorgehensweise im Vordergrund

Die Vorlesung wird durch Übungen ergänzt, in denen Aufgaben die Studierenden schrittweise mit den Fragen des Entwurfs und der Systemgestaltung vertraut machen sollen.

3.2 Objektorientierte Softwareentwicklung

Eine Schwerpunktsetzung für die Softwaretechnikausbildung im Hauptstudium findet im Rahmen der zweisemestrigen Lehrveranstaltung „Objektorientierte Softwareentwicklung“ statt. Gegenstand ist die Analyse, Modellierung und Konstruktion eines interaktiven Arbeitsplatzsystems. Die konzeptionelle Grundlage ist der am Arbeitsbereich Softwaretechnik vertretene Werkzeug & Material-Ansatz (vgl. [Zue98]). Entsprechend umfaßt die Themenliste die Grundkonzepte der Objektorientierung sowohl in der Softwarekonstruktion als auch im Vorgehensmodell, Entwurfsmuster, Rahmenwerke und Architekturfragen; dazu kommen anwendungsorientierte Dokumenttypen, Leitbilder und Entwurfsmetphern als Anleitungen für die Gestaltung interaktiver Anwendungssoftware.

Im ersten Semester steht die Analyse und der fachliche Entwurf im Vordergrund; im zweiten Semester

soll auf entsprechend konzeptioneller Basis ein Prototyp konstruiert werden.

3.3 Softwareentwicklung in Organisationen

Der thematische Schwerpunkt „Softwarentwicklung in Organisationen“ befindet sich im Aufbau. Exemplarisch für die Ausrichtung betrachten wir ein aktuelles Seminar zum gleichen Thema. Hier stehen die Wechselwirkung zwischen Softwareentwicklung und -einsatz sowie die Herausbildung und der Fall von Organisationsgrenzen im Mittelpunkt. Einzelthemen sind:

- Auswirkungen des Einsatzes von Informationstechnik auf Organisationsgrenzen in konkreten Anwendungsbereichen von Informationstechnik wie z.B. Bibliotheksnutzung, Universitätsausbildung, Gesundheitswesen, Finanzdienstleistungen, Kommunalverwaltung, Fertigungslogistik.
- Veränderungen der Softwareentwicklung durch Globalisierung am Beispiel konkreter Projekte in Organisationen (Softwarefirmen, Anwenderunternehmen).
- Vergegenständlichung von Organisationskonzepten in Softwareprodukten (z.B. aus den Bereichen Workflowmanagementsysteme und Groupware).

In dieser Lehrveranstaltung wird Konzept- und Literaturarbeit mit der experimentellen Analyse und Bewertung exemplarischer Softwaresysteme verzahnt.

3.4 Programmier-, Entwurfs- und Spezifikationskonzepte

Eine Vorlesung und ein vertiefendes Seminar behandeln die wichtigsten Konzepte zur formalen Behandlung von Programmtexten auf der Ebene Syntax und Semantik. Sie vermitteln ein vertieftes Verständnis der Konstrukte von imperativen Programmiersprachen und geben eine Einführung in formale Kalküle zur Semantikbeschreibung (operational, axiomatisch, denotational). Ferner stellen sie den Zusammenhang zur Softwareentwicklung (Spezifikation und Verifikation von Programmen) her. Ein Arbeitsbereich „Programmiersprachen und Compilerbau“ existiert in Hamburg nicht.

4 Neue Formen und Inhalte

Die Techniken, Vorgehensweisen und Hilfsmittel professioneller Softwareentwicklung sollten im Studium erfahrbar sein. Aber der semesterweise Lehrbetrieb und die Diplomarbeit machen es schwer, große Softwaresysteme unter professionellen Bedingungen selbst zu erstellen.

Wir erläutern im folgenden an einzelnen Beispielen, wie die Prinzipien der evolutionären Systementwicklung, die Probleme der Softwareentwicklung im Team sowie die Planung und Organisation eines Projekts in den universitären Lehrbetrieb eingehen können.

4.1 Einführung in die Softwaretechnik

Konventionell werden Kerngebietsveranstaltungen in Hamburg als Vorlesungen durchgeführt. Wir haben die Einführung in die Softwaretechnik von vorne herein als Kombination aus Vorlesung und Übung vorgesehen. In den Übungen steht die projektartige Gruppenarbeit im Vordergrund. Dabei beschäftigen sich die Studierenden zu Anfang mit der Analyse eines vorgegebenen Anwendungssystems, um es dann nach unserem evolutionären Ansatz auf der Basis von use cases schrittweise zu erweitern. Die recht hohe Teilnehmerzahl (ca. 150) rechtfertigt aus unserer Sicht diese Zweiteilung in Vorlesung und Übungsgruppen.

4.2 Objektorientierte Softwareentwicklung

Diese Lehrveranstaltung hat sich aus einem Paar von Vorlesung und Übung entwickelt. Nun ist daraus eine integrierte Lehrveranstaltung für ca. 25 Studierende geworden, in der sich kompakte Workshops mit Projektarbeit abwechseln. Zu Beginn und in der Mitte eines Semesters finden ein- bis zweitägige Workshops mit einem Dozenten statt. Dort wechseln sich Vortragseinheiten mit kleinen Gruppenübungen und Diskussionen ab. Themen sind unser Verständnis der Objektorientierung, das sich im genannten Werkzeug & Material-Ansatz niederschlägt. Dies umfaßt die Bereich Analyse, Entwurf und Konstruktion ebenso wie eine methodische Vorgehensweise in Softwareprojekten. Dazu kommen unterschiedliche Dokumentationstypen, bei denen wir vor allem auf die sog. anwendungsorientierten Dokumenttypen Wert legen.

Im Semester wird in Projektteams von 3 bis 5 Personen an einer praktischen Aufgabenstellung gearbeitet. Dabei versuchen wir, in engen Rückkopplungszyklen zwischen analysierenden, modellierenden und evaluierenden Tätigkeiten zu wechseln. Schon während der initialen Workshops werden Ergebnisse von Gruppenarbeit vorgestellt und diskutiert. In der längeren Projektphase stellen die einzelnen Teams ihren beiden Betreuern Zwischenergebnisse vor. Am Ende jedes Semesters ist dann jeweils ein dreitägiger Workshop für die Vorstellung und das Review der Ergebnisse vorgesehen. Hier kritisieren sich zunächst die verschiedenen Teams auf der Basis vorher verteilter Unterlagen gegenseitig. Dann diskutieren die Lehrenden die Entwürfe und Ergebnisse, wobei weiterführende Konzepte und Techniken vorgestellt werden. Diese Formen der Autor-Kritiker-Zyklen geben den Lehrenden und Studierenden ein sehr gutes Bild über den erreichten Wissens- und Kenntnisstand.

4.3 Softwareentwicklung in Organisationen

Generell befassen wir uns hier mit Fragen wie:

- Das Verständnis des Verhältnisses Mensch / Technik / Organisation bezogen auf Softwareentwicklung,
- Rollenmodelle und Leitbilder bei organisationsbezogener Softwareentwicklung,
- Softwareentwicklung und –einsatz bezogen auf Organisationsentwicklung,
- Reinterpretation der Aktivitäten der Softwareentwicklung (z.B. Anforderungsermittlung) bezogen auf Organisationsentwicklung,
- Vergegenständlichung von Organisationskonzepten in Software,
- Methodische Einführung von Software.

Da sich dieser Themenschwerpunkt im Aufbau befindet, haben wir auch hier mit neuen Veranstaltungsformen experimentiert. Die aktuelle Lehrveranstaltung wird gemeinsam mit dem Institut für Informatik (IFI) der Universität Zürich durchgeführt. Die Themen werden in Kleingruppen erarbeitet und durch die Veranstalter betreut. Die Literatur- und Materialsammlung sowie die Präsentation der Ergebnisse erfolgen weitgehend über gemeinsam bearbeitbare Hypertexte im WWW. Dazu gibt es synchrone Diskussionen und Vorträge mit Hilfe von Internettechniken (Chat, Internettelefonie, Bildübertragung).

Um Erfahrungen mit dem Einsatz synchroner und asynchroner Internet-Techniken zu sammeln, führen wir parallel zu diesem Seminar ein Projekt durch, in der diese Technologien erprobt und evaluiert werden. Dieses Projekt stellt das die Infrastruktur für das Seminar bereit.

4.4 Das Rahmenwerkprojekt

Das Problem der Entwicklung großer Anwendungssysteme an der Universität ist nicht einfach zu lösen. Ein Mittel ist die Bereitstellung von Anwendungsrahmenwerken und Klassenbibliotheken für Lehrveranstaltungen und Diplomarbeiten. Wir haben bereits vor einigen Jahren die Entwicklung von Rahmenwerken begonnen (vgl. [BibV30]). Studientische Initiative hat jetzt zu einer Lehrveranstaltung geführt, die die Entwicklung eines Java-Rahmenwerks für die generischen Aspekte des Werkzeug & Material-Ansatzes zum Gegenstand hat (vgl. [JWAM98]). Diese Lehrveranstaltung kann als mehrsemestrige Projeklehrveranstaltung verstanden werden. Studierende und Lehrende aus dem Mittelbau haben sich in der Form eines Softwareteams organisiert. Es gibt Teilgruppen und unterschiedliche Verantwortlichkeiten. Neben den regelmäßigen „Projektsitzungen“ werden Workshops zu wechselnden Themen organisiert. Dort stellen Studierende Ergebnisse aus ihren Diplomarbeiten vor oder die Lehrenden berichten über neue Konzepte und Techniken. Das dabei entwickelte Rahmenwerk ist gleichzeitig konstruktive Basis für Diplomarbeiten und Dissertationen. Um dieses Rahmenwerk an professionelle Standards heranzuführen, setzen wir es in Kooperationsprojekten mit der Wirtschaft ein. Dieser „Echteinsatz“ wirkt zusätzlich motivierend auf die Studierenden, die ihre Ergebnisse praktisch verwendet sehen. Wir haben durchweg gute Erfahrungen in der Zusammenarbeit mit technisch innovativen Firmen gesammelt, die wesentlich zu einer Evaluierung unserer Ansätze und zu praxisnahen Weiterentwicklung beitragen.

5 Werkzeuge, Hilfsmittel und Techniken

Gerade in der Softwaretechnik stellt sich die Frage nach dem Werkzeugeinsatz. Aber professionelle Hilfsmittel sind relativ komplex und erfordern einen hohen Einarbeitungsaufwand. Oft sind sie recht kostspielig. Schließlich ist in diesem Bereich der Markt sehr „in Bewegung“, so daß etwa bei Java-Entwicklungsumgebungen kaum von bewährten und zukunftsträchtigen Systemen gesprochen werden kann.

In unseren Lehrveranstaltungen bieten wir eine Werkzeugunterstützung an, die verschiedenen Kriterien gerecht werden soll. Die Entwicklungsumgebungen sollen möglichst sowohl auf Windows-PCs als auch für Unix-Workstations verfügbar sein. Diese Umgebungen sollen in ihrer Grundfunktionalität so einfach sein, daß sie innerhalb kurzer Zeit beherrschbar sind. Spezielle Werkzeuge wie Versionsverwaltungssysteme und Testumgebungen, setzen wir gezielt nur in Projektveranstaltungen ein.

In fast allen Lehrveranstaltungen des Hauptstudiums setzen wir Java ein. Wir schätzen aber ein, daß Java als einzige objektorientierte Programmiersprache gegenwärtig nicht hinreicht. In der objektorientieren Softwareentwicklung ist C++ aktuell als Entwicklungssprache nicht wegzudenken. Daher haben wir im Rahmen der Lehrveranstaltung Objektorientierte Softwareentwicklung zwei studentische C++-Kurse initiiert. Wir haben zwei studentische Dozenten mit unseren C++-Kursunterlagen vertraut gemacht und den organisatorischen Rahmen für die Kurse geboten. Beide sehr gut besuchten Kurse fanden in Randzeiten (Wochenende und Pfingstferien) statt und waren aus Sicht aller Beteiligten recht erfolgreich.

Zu den Techniken, die die Studierenden im Rahmen unserer Lehrveranstaltungen einsetzen, gehörigen einige, die gewöhnlich nicht zum engeren softwaretechnischen Handwerkzeug gezählt werden:

- Interviewtechniken, sind die Grundlage für einige anwendungsorientierte Dokumenttypen wie Szenarios oder Glossareinträge;
- Präsentationsunterlagen, die in Lehrveranstaltungen und Projekten eingesetzt werden, reichen von

Folien bis zu Wandzeitungen und Demonstrationsprototypen.

Wenn wir noch die Bereitstellung von Unterlagen und Kommunikationskanälen über das Intranet und die aufwendige Technologie für das Internetseminar hinzurechnen, ergibt sich eine recht hohe Anforderung an die technische Infrastruktur durch diese Art von Lehrveranstaltungen. Insgesamt geht der Aufwand deutlich über traditionelle Lehrveranstaltungen hinaus.

6 Einschätzung

Es scheint mir mehr als ein Gemeinplatz zu sein, daß sich die Softwaretechnik an der Universität verstärkt fragen lassen muß, ob sie den gestiegenen Anforderungen an wissenschaftliche Forschung und Ausbildung in einem hoch dynamischen Umfeld entspricht. Wenn heute in industriellen Kontexten vielfach interessantere und komplexere Forschungs- und Entwicklungsthemen bearbeitet werden, als an Hochschulen, dann ist dies aus meiner Sicht eine unbefriedigende Situation. Wir haben in diesem Artikel einige Lösungsansätze für die Verbesserung der universitären Lehre in der Softwaretechnik skizziert (s.a. [WKKL98], [GPZ93]), die im engen Zusammenhang mit den Möglichkeiten einer praxisorientierten Forschung steht. Die Leitlinien sind dabei evolutionäre Systementwicklung, Anwendungsorientierung sowie die Arbeit mit praxisnahen Systemen und Techniken. Obwohl die neuen Lehr- und Veranstaltungsformen aufwendiger sind, als die traditionellen, ermutigen uns die besseren Lernerfolge und die hohe Akzeptanz bei den Studierenden.

Beruhigend ist die Situation trotzdem nicht. Aus heutiger Sicht können wir zwar durch Neuerungen in der Softwaretechniklehre einiges erreichen, aber wir können die Komplexitätskluft zwischen studentischen Entwicklungsvorhaben und industriellen Softwareprojekten nur bedingt verkleinern.

7 Ausblick

Das Informatikstudium an der Universität Hamburg befindet sich im Umbruch. Mit dem Wintersemester 1998/99 ist eine neue Studienordnung in Kraft gesetzt worden (vgl. [Studium98]). Die Aussagen über die Ausbildung in der Softwaretechnik in diesem Beitrag beziehen sich auf die Praxis innerhalb der alten Studienordnung.

Nach der neuen Studienordnung gibt es allerdings inhaltlich eine vergleichbare Ausrichtung, da Softwaretechnik weiterhin Bestandteil der obligatorischen Grundlagen in der Praktischen Informatik sein wird. Im Hauptstudium ist die Softwaretechnik ein Schwerpunkt sowohl im Profil *Informatiksysteme in Organisationen* als auch im Profil *Softwaresystemtechnik*. Wie sich dies konkret auswirkt, wird in ca. zwei Jahren deutlich werden, wenn die Studierenden nach der neuen Studienordnung das Grundstudium abgeschlossen haben.

8 Literaturhinweise

[BibV30] Carola Lilienthal, Stefan Roock, Ulfert Weiss, Henning Wolf: *C++-Framework*. Universität Hamburg, Fachbereich Informatik, Arbeitsbereich Softwaretechnik,. http://swt-www.informatik.uni-hamburg.de/Software/BibV30, 1997.

[JWAM98] Niels Fricke, Carola Lilienthal, Martin Lippert, Stefan Roock, Henning Wolf: *Java Framework*. Universität Hamburg, Fachbereich Informatik, Arbeitsbereich Softwaretechnik. http://swt-www.informatik.uni-hamburg.de/Software/JWAM, 1998.

[Floyd94] C. Floyd: *Software-Engineering - und dann?* Informatik Spektrum, Band 17, Heft 1, Springer-Verlag, Berlin, Heidelberg, New York, Tokio, 1994, S. 29-37.

[FZ97] C. Floyd, H. Züllighoven: *Softwaretechnik*. In: Rechenberg/Pomberger (Hrsg.) Informatik-Handbuch, Hanser Verlag, München, Wien, 1997. S. 641-667.

[GPZ93] G. Gryczan, U. Piepenburg, H. Züllighoven: *Erfahrungen mit Industriekooperationen unter Ausbildungsaspekten.* In: J. Raasch, Th. Bassler (Hrsg.): Software Engineering im Unterricht der Hochschulen SEUH '93. Berichte des German Chapter of the ACM 38, Teubner, Stuttgart, 1993. S. 140-150.

[Studium98] *Studienordnung '98, Prüfungsordnung '98, Rahmenstudienplan des Fachbereichs Informatik der Universität Hamburg*. www.informatik.uni-hamburg.de/Info/Studium/informatik.html

[WKKL98] Ingrid Wetzel, Ralf Klischewski, Anita Krabbel, Carola Lilienthal: *Kooperation für Software für Kooperation: Erfahrungen aus einem partizipativen Softwaretechnikprojekt*, GI Tagung zu Informatik und Ausbildung, April 1998.

[Zue98] Züllighoven, H. et al.: *Das objektorientierte Konstruktionshandbuch nach dem Werkzeug & Material-Ansatz*. d-punkt Verlag, Heidelberg, 1998.