Ausbildung in Softwaretechnik an der Freien Universität Berlin

Klaus-Peter Löhr
Institut für Informatik
Fachbereich Mathematik und Informatik
lohr@inf.fu-berlin.de

1 Stellenwert im Curriculum

Softwaretechnische Inhalte werden hauptsächlich in Lehrveranstaltungen des 3., 4. und 5. Semesters vermittelt. Diese Veranstaltungen werden von Studierenden des Diplomstudiengangs, der Lehramtsstudiengänge und der Magisterstudiengänge besucht. („Magister" bedeutet *nicht* „Master"; es handelt sich um primär geisteswissenschaftliche Studiengänge mit Informatik als zweitem Hauptfach.) Das Angebot umfaßt folgende Veranstaltungen:

> *3. Semester: Algorithmen und Programmierung III* (4 SWS Vorlesung + 2 SWS Tutorien à 15 Teilnehmer): Behandelt werden Datenstrukturen und Algorithmen, sowohl in funktionaler Formulierung (Haskell) als auch in objektorientierter Formulierung (Java). Dabei werden u.a. die folgenden softwaretechnischen Inhalte behandelt: Datenabstraktion, Spezifikation/Repräsentation/Implementierung von ADT, Korrektheit der Repräsentation (abstrakte/konkrete Invariante, Abstraktionsfunktion, abstrakte/konkrete Pre/Postconditions), Benutzung von Haskell als Spezifikationssprache. Einführung in Entwurfsmuster.
>
> *4. Semester: Software-Praktikum* (4 SWS Tutorien à 15 Teilnehmer): Unter Leitung eines Tutors wird arbeitsteilig ein größeres Softwaresystem entwickelt (gewöhnlich in Java). Als Thema wird häufig die Programmierung eines Brettspieles gewählt (was die Möglichkeit eröffnet, die entwickelten Systeme zu Semesterende in einem Turnier gegeneinander antreten zu lassen). Zur Systemspezifikation werden zahlreiche Vorgaben gemacht, so daß zügig mit dem Entwerfen begonnen werden kann. In der Regel ist damit sichergestellt, daß am Ende des Semesters ein lauffähiges und (mäßig gut) getestetes System vorliegt. Ziel der Veranstaltung ist das sichere Beherrschen der im 1. – 3. Semester studierten Methoden und Techniken für das Programmieren im Kleinen, ferner die Sensibilisierung für die bei der Entwicklung größerer Systeme auftretende Probleme (Beherrschung von Komplexität, arbeitsteilige Vorgehensweise).
>
> *5. Semester: Softwaretechnik* (4 SWS Vorlesung + 2 SWS Praktikum): siehe 2.
>
> *7. + 8. Semester: Software-Projekt* (2 x 4 SWS Projekt): siehe 3.

Die Veranstaltungen im 3. und 4. Semester sind Pflicht für alle Informatik-Studiengänge. Die *Softwaretechnik* im 5. Semester ist eine von drei Pflichtveranstaltungen im Hauptstudium des Diplomstudiengangs. (Die anderen beiden sind *Rechnerarchitektur* und *Entwurf und Analyse von Algorithmen*.)

2 Die Lehrveranstaltung „Softwaretechnik“

Nach dieser Veranstaltung sollten die Teilnehmer die zentralen Begriffe, Prinzipien und Methoden der Softwaretechnik beherrschen und im Kontext kleiner Projekte zu systematischer Software-Entwicklung entsprechend dem Stand der Kunst fähig sein.

2.1 Inhalte

In der vierstündigen Vorlesung wird vermieden, die Abfolge der Stoffvermittlung streng an einem Phasenmodell zu orientieren. Der Inhalt ist in vier Teile gegliedert:

1. Teil – *Einführung:* Wozu Softwaretechnik? Software-Qualität, Qualitätskontrolle. Phasenmodelle, Prototyping, Evolutionäre Entwicklung.

2. *Teil – Das Handwerkszeug:* Spezifikation versus Implementierung, Formale Methoden, Systeme und Teilsysteme, objektorientierte Konstruktion kleiner Teilsysteme, persistente Objekte und Datenbankanschluß, Benutzeroberflächen und GUI Builders.

3. *Teil – Frühe Phasen:* Anforderungsanalyse, Modellierung, Systemspezifikation, Software-Architektur, Entwurf, Entwurfsmuster und Frameworks (alles nur objektorientiert).

4. *Teil – Späte Phasen:* Testen, Pflegen; Reengineering. Varia.

Im begleitenden zweistündigen Praktikum werden vorwiegend umfangreichere Aufgaben bearbeitet, anfangs mehr Trockenaufgaben, später mehr praktische Aufgaben am Rechner. Dabei werden für die Spezifikation teils Z, teils deklarative Sprachen (Haskell, Prolog), teils Statecharts verwendet. Für die Programmierung werden Java und GJ eingesetzt, zum gegenwärtigen Zeitpunkt – zwecks nahtloser Einbeziehung von Persistenz und Benutzeroberflächen – zusammen mit ObjectStore und VisualAge. Modellierung und Entwurf erfolgen mit UML, am Rechner gestützt durch Rational Rose.

2.2 Überlegungen zu praktischer Arbeit und Werkzeugen

Es ist nicht leicht, einen klugen Mittelkurs zu fahren zwischen den Extremen „Keine – ohnehin ephemeren! – Werkzeuge einsetzen!“ und „State-of-the-art-Software-Entwicklung mit vielen tollen Werkzeugen!“ Wir sind davon abgekommen, komplexe professionelle Entwicklungssysteme einsetzen zu wollen; der Einarbeitungsaufwand steht in keinem vernünftigen Verhältnis zum Erkenntnisgewinn. Anders ist die Situation bei kleinen, leichtgewichtigen Werkzeugen: der Umgang mit ihnen ist schnell erlernt; und ihr Stellenwert wird tatsächlich erst im praktischen Umgang einsichtig (typisches Beispiel: GUI Builders).

Sicher ist eine Vorlesungsteilnahme ohne eigene praktische Arbeit im Fall der Softwaretechnik noch abwegiger als bei vielen anderen Fächern. Und da der Gegenstand der Softwaretechnik Software-*Projekte* sind, könnte man auch fordern, daß die Vorlesung untrennbar mit einem Projekt verbunden sein sollte, das sich über mindestens zwei Semester erstreckt. Wir sind von diesem – früher praktizierten – Modell abgekommen, weil es sich für die Qualität des Projekts als nachteilig erwies, daß die Teilnehmer während einer frühen Phase noch keinen soliden Überblick über Erfordernisse und Vorgehensweise der bevorstehenden Phasen hatten. Das jetzt praktizierte Modell holt die Teilnehmer dort ab, wo

sie nach *Algorithmen und Programmierung III* und dem *Software-Praktikum* des Grundstudiums stehen, und vertieft und erweitert die dort erworbenen Kenntnisse. Das zweistündige Praktikum – wie oben beschrieben – hat die Funktion, typische Tätigkeiten der Software-Entwicklung im Rahmen kleiner Fallstudien einzuüben und dabei einige wenige typische leichtgewichtige Werkzeuge kennenzulernen. Nach Abschluß der Veranstaltung sollten die Teilnehmer in der Lage sein, zügig und kompetent in ein größeres Projekt einzusteigen.

3 Weiterführende Veranstaltungen

Neben Seminaren ist hier vor allem das zweisemestrige *Software-Projekt* (2 x 4 SWS) zu nennen. Den Teilnehmern der *Softwaretechnik* wird empfohlen, nach zusätzlicher Absolvierung der Veranstaltung *Datenbanksysteme* am Software-Projekt teilzunehmen. Das Software-Projekt ist als Wahlpflicht-Veranstaltung zu klassifizieren: die Prüfungsordnung schreibt „Praktikum oder Projekt mit mindestens 6 SWS“ in einem frei gewählten Bereich vor.

Das Software-Projekt wird 1999 zum ersten Mal durchgeführt werden. Erfahrungen liegen somit noch nicht vor. Es ist geplant, mit einer Projektgruppe von maximal 15 Teilnehmern über zwei Semester hinweg eine Aufgabe aus dem Bereich *Verteilte Informationssysteme* zu lösen. Dabei sollen einschlägige Technologien wie WWW, XML, CORBA (und/oder DCOM?) zum Einsatz kommen. Im Gegensatz zum Software-Praktikum des Grundstudiums ist das Software-Projekt realistisch angelegt: Ausgangspunkt ist nicht ein festes Pflichtenheft, sondern lediglich eine vage Wunschliste; die Projektteilnehmer verfügen nicht über einen einheitlichen Kenntnisstand hinsichtlich der zu verwendenden Technologien; existierende Systeme und Komponenten müssen in die Entwicklung einbezogen werden.

4 Erfahrungen

Unter den Disziplinen der Informatik (wie z.B. *Datenbanksysteme* in der Praktischen oder *Textverarbeitungssysteme* in der Angewandten Informatik) nimmt die Softwaretechnik als Meta-Disziplin eine Sonderrolle ein. Die daraus resultierenden Charakteristika sind bekannt: Ausfransende Ränder, viel „weiche“ Materie, Heterogenität. Dies hat Auswirkungen auf die Lehre: Ein kohärentes Bild von Softwaretechnik ist schwer vermittelbar. Die propagierte systematische Vorgehensweise wird oft als „bürokratische“ empfunden; die erforderlich „Internalisierung“ der zentralen softwaretechnischen Prinzipien kommt oft nicht zustande. Ein hübsches Werkzeug (graphische Oberflächen!) wird gerne angenommen, aber der eigene Entwicklungsstil wird oft nur geringfügig verbessert. Wir merken das an manchen externen Diplomarbeiten, die trotz intensiver Verfolgung seitens der Hochschule softwaretechnische Qualitätsmängel aufweisen.

Damit solide softwaretechnische Arbeitsweise in Fleisch und Blut übergeht, genügt es nicht, an einem oder zwei Praktika teilzunehmen. Wir erwarten, daß das zweisemestrige Projekt die beabsichtigte positive Wirkung haben wird. Offen bleibt gegenwärtig die Frage, ob im Hinblick auf ein sinnvolles Qualifikationsprofil der Absolventen das Projekt zur Pflichtveranstaltung gemacht werden sollte.