

Sidestep: Die Informatik-Initiative von sd&m

Johannes Siedersleben

sd&m AG, Thomas-Dehler-Str. 27, D 81737 München
johannes.siedersleben@sdm.de

Zusammenfassung

Das Softwarehaus sd&m, München, veranstaltet in enger Zusammenarbeit mit der TU München ein sechsmonatiges Ausbildungsprogramm für Quereinsteiger, die ein natur- oder ingenieurwissenschaftliches Studium absolviert haben. Wir beschreiben die Ziele, die Organisation und den Inhalt dieses Programms.

1 Übersicht

Alle Software-Häuser in Deutschland und anderswo klagen über Nachwuchsmangel. Der Erfolg oder Mißerfolg eines Unternehmens entscheidet sich heute vor allem beim Wettbewerb um junge, geeignete Mitarbeiter: Nur wer es schafft, die besten Absolventen in ausreichender Zahl an sich zu binden, wird auf dem Markt bestehen.

Software-Häuser brauchen nicht einfach Informatiker, sondern Informatik-Berater. Das sind zunächst einmal Informatiker, die sich ihre Kenntnisse im Studium oder anderswo angeeignet haben. Sie können aber mehr: Sie schreiben gute Texte, sie halten gute Vorträge, sie treten beim Kunden gewandt und diplomatisch auf, sie sind teamfähig, sie sind belastbar, sie behalten die Übersicht und sie organisieren sich selbst. Sie sind natürlich keine Übermenschen, aber sie sind *wirksam* im Sinn von F. Malik [3].

Solche Leute sind rar. Das Greencard-Programm hilft uns dabei wenig: Es beschert uns Informatiker aus anderen Kulturkreisen, die – allein schon wegen des Sprachproblems – in der Regel erst nach längerer Einarbeitungszeit als Informatik-Berater arbeiten können. Deshalb haben wir bei sd&m einen anderen Gedanken verfolgt:

Es gibt in Deutschland viele junge Leute mit der Eignung zum Informatik-Berater und einer systematischen mathematischen Grundausbildung, denen allerdings das Informatik-Wissen fehlt. Man findet sie unter den Absolventen der natur- und ingenieurwissenschaftlichen Studiengänge. Nun herrscht auch da nicht gerade Arbeitslosigkeit, aber erstens ist der Wachstumsmarkt dort nicht ganz so leer gefegt wie in der Informatik, und zweitens ist die Informatik auch für viele Nicht-Informatiker attraktiv wegen der interessanten Aufgaben und der guten Karriere-

möglichkeiten. Wir hatten bei der Erschließung dieses Potenzials zwei Probleme zu lösen:

- Wie kann man geeignete Nicht-Informatiker in einem vertretbaren Zeitraum mit dem erforderlichen Informatik-Wissen ausstatten?
- Wie kann man geeignete Kandidaten ansprechen und gewinnen?

Oberstes Ziel war dabei die Qualität: Wir wollten ein erstklassiges Programm für erstklassige Kandidaten; Einstiegskurse auf dem Niveau von Windows-Klickerei gibt es schon genug. Hier nun das Konzept:

Wir bieten 25 Quereinsteigern (daher der Name "sidestep") ein 6-monatiges Ausbildungsprogramm von Januar bis Juni 2001. Dieses Programm wurde in enger Zusammenarbeit mit der TU München gestaltet, die den größten Teil der Dozenten stellt. Eine Dozentin kommt von der Ludwig-Maximilian-Universität München, die übrigen von sd&m Research, der Forschungs- und Schulungsabteilung von sd&m.

Das Programm findet statt in den Räumen von sd&m München; jeder Teilnehmer bekommt einen eigenen Arbeitsplatz (Schreibtisch, Laptop usw.). Erfahrene sd&m-Mitarbeiter kümmern sich als Mentoren um das fachliche und seelische Wohl der Teilnehmer. Bereits während der Ausbildungszeit wird ein Gehalt von 2800,- Euro pro Monat gezahlt. Die Teilnehmer erhalten eine Garantie auf weitere Anstellung nach erfolgreicher Abschlußprüfung, und sie verpflichten sich, mindestens 3 Jahre bei sd&m zu bleiben. Im Fall des vorzeitigen Ausscheidens ist eine Pönale fällig.

sd&m finanziert das Programm zu 100%. Allein die Gehälter der Teilnehmer belaufen sich im Ausbildungszeitraum auf 420.000,- Euro.

2 Das Programm

Wie kann man das Informatik-Studium auf 6 Monate verdichten? Fünf Schritte führen zum Ziel:

1. Man streicht die Mathematik-Grundausbildung, denn das können die Teilnehmer schon.
2. Man streicht das Nebenfach, denn das haben die Teilnehmer als Hauptfach studiert.
3. Man streicht die Diplomarbeit, denn die Teilnehmer haben in ihrem Hauptfach bereits eine geschrieben.
4. Man streicht solche Informatik-Fächer, die für uns als Software-Haus nur untergeordnete Bedeutung besitzen (z. B. Automatisches Beweisen und Problemelösen).
5. Man vertieft diejenigen Fächer, die uns besonders wichtig sind: Programmieren, Datenbanken, Software-Entwurf.

Das Ergebnis ist ein Programm, das 560 Vorlesungs- bzw. Übungsstunden à 45 Minuten in 20 Netto-Wochen unterbringt. Diese sind verteilt auf die ersten 26 Wochen

des Jahres 2001; die Lücken ergeben sich durch den Start des Programms am 8. Januar, Osterferien, diverse Feiertage und die Prüfungswoche am Schluß. Die resultierenden 28 Wochenstunden sind hart, aber machbar. Das Programm ist gedacht als zielgerichtetes Aufbaustudium für Teilnehmer, die schon wissen, wie man lernt. Es ist aufgeteilt in zwei gleich lange Abschnitte:

Fach	Abschnitt 1 Stunden	Abschnitt 2 Stunden	Summe	Dozent
Grundlagen der Informatik	60	60	120	Paul (TUM)
Programmierpraktikum	120	60	180	Siedersleben (sd&m)
Software-Engineering		60	60	Siedersleben (sd&m)
Grundlegende Algorithmen und Datenstrukturen	40		40	Steger (TUM)
Datenbanken	40	40	80	Kossmann (TUM)
Betriebssysteme	40		40	Baumgarten (TUM)
Rechnerarchitektur	20			Linhoff-Popien (LMU)
Rechnernetze		20	40	Linhoff-Popien (LMU)
Summe	300	260	560	

Tab 1: Vorlesungsplan

Die Stundenzahlen enthalten Vorlesungen und Übungen; das Verhältnis ist etwa 3:2. Die Veranstaltungen finden im wesentlichen vormittags statt; die Nachmittage dienen der Vor- und Nachbereitung.

Weiche Themen wie Vortragstechnik oder Gesprächsführung sind nicht enthalten, denn: Erstens stehen die Teilnehmer ab dem 1. Juli 2001 im Beruf und können diese Fähigkeiten täglich trainieren. Zweitens gibt es bei sd&m gerade in diesem Bereich spezielle Fortbildungsmaßnahmen, in deren Genuß jeder sd&m-er kommt, also auch die sidestep-Leute. Und schließlich waren diese Fähigkeiten wichtiges Kriterium bei der Auswahl der Kandidaten.

Der sidestep-Absolvent ist selbstverständlich kein voller Informatiker, aber er besitzt eine solide Kenntnis der Informatik-Grundlagen und wird von daher in der Lage sein, sich rasch in neue Gebiete einzuarbeiten. Die genannten 560 Vorlesungs- und Übungsstunden entsprechen nur etwa 60 Semesterwochenstunden oder 40 ECTS-Punkten¹, aber diese schematische Betrachtung wird dem sidestep-Programm nicht gerecht. Dafür gibt es drei Gründe:

¹ European Credit Transfer System

- Die sidestep-Teilnehmer sind nach strengen Maßstäben ausgewählt. Sie haben gute mathematische Vorkenntnisse, und sie sind – nicht zuletzt auch dank der Bezahlung – hoch motiviert.
- Die sidestep-Teilnehmer haben eine optimale Arbeitsumgebung. Sie sind keine Einzelkämpfer, sondern sind bereits während der Ausbildung als sd&m-Mitarbeiter integriert. Die persönlichen Mentoren helfen in schwierigen Situationen.
- Die Ausbildung ist zugeschnitten auf das Unternehmensziel von sd&m: Entwicklung großer betrieblicher Informationssysteme. Daher haben Themen wie Datenbanken oder Programmierpraktikum ein größeres Gewicht als in manchen Informatik-Studiengängen.

Der sidestep-Absolvent ist auf den Beruf als Informatik-Berater vielleicht besser vorbereitet als ein Informatiker, der die für uns relevanten Fächer nur im vorgeschriebenen Mindestumfang belegt hat.

3 Programmierpraktikum und Software-Engineering

Dieser Abschnitt beschreibt zwei Fächer, die uns im sidestep-Programm besonders am Herzen liegen: Programmierpraktikum und Software-Engineering. Die Grundlage dazu bilden die einwöchigen, intensiven Programmier- und Designer-Schools, die sd&m seit drei Jahren für junge sd&m-Mitarbeiter durchführt.

3.1 Programmierpraktikum

Jedes Code-Review bestätigt folgende These: Viele Programme sind zu kompliziert und zu lang für das, was sie eigentlich tun sollen. Softwarequalität beginnt mit der Qualität der Programme. Deshalb wollen wir den Teilnehmern zeigen, wie man einfache, klare und im Idealfall elegante Programme schreibt. Einige Stichpunkte dazu:

- Denken in einfachen mathematischen Begriffen (Mengen, Funktionen, Prädikate),
- Programmiermuster
- Objektorientierung mit Augenmaß; Schnittstellen statt Vererbung
- Denken im richtigen Paradigma (Generierung, Vererbung, Vorlage; vgl. Coplien[1]).
- Themenorientiertes Programmieren: Wie programmiert man einen Datenbank-Zugriff, wie schreibt man einen Dialog, wie nutzt man Middleware wie Corba?

Die verwendeten Programmiersprachen sind Java, C++ und Cobol; als Übungsmaterial dienen Projekte aus der sd&m-Praxis in angepaßter Form.

3.2 Software-Engineering

Zentraler Punkt der Vorlesung ist der Software-Entwurf. Hier gilt sinngemäß dasselbe wie bei der Programmierung: Viele Systeme sind schon vom Entwurf her zu kompliziert; Software-Architekten haben einen fatalen Hang zum Komplexen. Hier unser Ansatz: Es gibt eine Reihe von immer wiederkehrenden Entwurfsaufgaben, z.B.:

- Datenbankzugriff mit Standardthemen wie Caching oder Objektidentität
- Dialogprogrammierung mit Standardthemen wie Zustandskontrolle, Fehlerbehandlung, Feldplausibilitäten, Kreuzplausibilitäten
- Berechtigungen
- Regelauswerter
- Anwendungsthemen wie Stückliste, Workflow.

Zu diesen Entwurfsaufgaben entwickeln wir Musterlösungen (nur Text und Bilder, kein Code), die mit unterschiedlichem Aufwand in verschiedene Programmiersprachen (z.B. Java, Cobol) und Programmierumgebungen (z.B. EJB, CICS) übertragbar sind (Man vergleiche hierzu Fowler[2].) Diese Musterlösungen, die sich ständig weiterentwickeln, sind das Kernstück der Ausbildung zum Software-Designer. Anhand dieser Musterlösungen vermitteln wir das Erste Gebot des Software-Entwurfs: Denken in Komponenten und Schnittstellen.

4 Zwischenbilanz

Der erste Schritt unseres Projekts – die Erstellung des Programms und die Rekrutierung der Dozenten – verlief völlig reibungslos. Hier ist besonders die Hilfe von Prof. Broy (TU München) hervorzuheben.

Der zweite Schritt war die Rekrutierung der Teilnehmer: Die Prognosen lagen zwischen 10 und 1000 Bewerbungen. Es sind (Stand Anfang November) etwa 200, nachdem wir im Juli und im September Anzeigen im redaktionellen Teil großer Zeitungen plaziert hatten (FAZ, Süddeutsche und ZEIT, ferner auch Spektrum der Wissenschaften und Physikalische Blätter). Daneben gab es die üblichen Kanäle wie Internet und persönliche Kontakte.

Die Qualität der Bewerbungen war wesentlich höher als bei unseren normalen Rekrutierungsaktionen. Wir konnten 70 Kandidaten mit ausgezeichneter Papierform einladen. Daraus ergaben sich mühelos 25 Verträge mit festen Zusagen.

Ein häufig gehörter Einwand lautet: Fast alle Software-Häuser beschäftigen doch schon heute Nicht-Informatiker in großer Zahl und fahren nicht schlecht damit – wozu also sidestep? Es sind vor allem drei Gründe:

1. sidestep macht geeignete Bewerber, denen nur das nötige Grundwissen fehlt, fit für unseren Beruf.

2. sidestep eröffnet neue Mitarbeiterquellen: sd&m erreicht gute Leute, die sonst nicht zu sd&m gekommen wären.
3. sidestep erhöht den Anteil an Mitarbeitern mit solidem Basiswissen in Informatik.

Auch bei sd&m liegt der Informatiker-Anteil nur bei etwa 50%. Daneben gibt es zahlreiche Mathematiker, Physiker, Ingenieure und Absolventen anderer Fachrichtungen, die sich – ausgehend von mehr oder weniger soliden Grundlagen – nach und nach in die Informatik eingearbeitet haben. Als grobe Schätzung kann man sagen, daß von den 50% Nicht-Informatikern bei sd&m etwa die Hälfte ausreichende Informatik-Grundlagen mitbringen. Die Übrigen (bezogen auf alle Mitarbeiter sind das also 25%) gehen einen steinigen und risikoreichen Weg: Der Festkörper-Physiker, der seine Versuchsreihen mit Hilfe großer Fortran-Programme ausgewertet hat, ist trotz seiner Fortran-Kenntnisse für die Arbeit als Software-Ingenieur in der Regel nur unzureichend vorbereitet. Training-on-the-Job mit knappen oder unzureichenden Grundlagen ist mühsam, frustrierend für den Mitarbeiter und teuer für den Arbeitgeber.

5 Folgerungen für die Informatikausbildung an den Hochschulen

Unser Ziel war die Erschließung eines neuen Mitarbeiterpotenzials, nicht die Reform der Informatikausbildung. Insofern ist das sidestep-Programm klassisch; die angebotenen Vorlesungen gibt es in dieser oder ähnlicher Form wohl in jedem Informatik-Curriculum.

Belastbare Erkenntnisse, die auch auf den Unterricht an Hochschulen übertragbar sind, werden wir frühestens nach Abschluß des Programms im Juli 2001 besitzen, spätestens nach der Bewährung der sidestep-Absolventen in der Praxis, also im Jahr 2002.

Wir wagen heute zwei Hypothesen, die im weiteren Verlauf zu verifizieren sind:

1. Das sidestep-Programm insgesamt kann Grundlage sein für vergleichbare Aufbaustudiengänge an Hochschulen. Dabei wäre es auf zwei oder drei Semester zu strecken. Eine Spezialisierung für andere Bereiche (etwa Echtzeitanwendungen) ist sehr gut vorstellbar.
2. Die in Abschnitt 3.1 und 3.2 herausgestellten Elemente des Programmierpraktikums und des Fachs "Software-Engineering" sind eine Bereicherung für entsprechende Veranstaltungen an den Hochschulen.

Literatur

1. Coplien, J.: Multi-Paradigm Design for C++. Addison-Wesley, 1999
2. Fowler, M.: Analysis Patterns. Addison Wesley, 199
3. Malik, F.: Führen, Leisten, Leben. DVA, 2000