Wir geben unser Bestes !

Programmier Methoden Praktikum: Aus der Sicht der Betreuenden

Ich möchte hiermit auf einige der Kritikpunkte, die von Alan Krempler in NATAN, 1-93/94 veröffentlicht wurden, antworten.

Wir haben im letzten Sommersemester erstmals versucht ein neues Arbeitskonzept für Tutoren und Assistenten zu entwickeln, das die Arbeiten rund um die Konstruktionsübungen PMP (Programmiermethoden Praktikum) effizienter gestalten sollte. Folgende Ziele wurden angestrebt:

(a) automatisierte Prüfumgebung
(b) fertig programmierte Musterbeispiele am Netz
(c) Änderung der Methodik der Einsichtnahme (mehr Zeit für die Studenten)

1. ad Beurteilung

Es gibt eine Software Umgebung, die ein teilautomatisiertes Prüfen von Programmen erlaubt.

Ein Abgabeprogramm unterstützt die Abgabe der Übungsprogramme und speichert diese auf einem entsprechenden Verzeichnis auf einem Hintergrundrechner. Ein Prüfprogramm holt sich das Programm eines Studenten, ruft den PASCAL Compiler auf und startet das Programm mit den entsprechenden Prüfungstestdaten. Die am Schirm angezeigten Ergebnisse werden auf ein Formblatt übertragen. Später berechnet ein Benotungsprogramm anhand eines Notenschlüssels und des Formblatts eine Note.

Dieses System ist meiner Meinung nach sehr objektiv. Die Anzahl der Testdaten und die Aufschlüsselung dieser in Funktions- und Belastungstest, sowie User Interface - Test wurde mit Vertretern der ÖH und mit den Tutoren besprochen.

Ohne ein solches System sollte man sich den Korrekturaufwand von Übungen, bei denen im Schnitt 500 Programme zu verbessern sind, einmal ausrechnen. Pro Programm ohne automatisierte Unterstützung braucht man 20 Minuten und das ergibt dann insgesamt 166 Stunden reine Korrekturtätigkeit. Das würde bedeuten, daß eine Person 4-5 Wochen lang 8 Stunden pro Tag ohne Kaffeepause mit Korrekturen beschäftigt wäre.

Aus diesem Grund haben wir uns auf folgende Vorgangsweise geeinigt:

- Assistent und Professor legen die Aufgabenstellung fest (Tutoren + ÖH konnten hierbei Vorschläge einbringen)
- Zwei Tutoren werden ausgewählt und extra über Werkvertrag bezahlt, um beide Aufgabenstellungen als Musterlösung auszuprogrammieren
- Assistent und Professor legen die Testdaten und die Prüfungstestdaten fest
- Assistent und Professor erabeiten eine Prüfmethodik (z.Bsp. Testdokumentationsblätter, Design der Testumgebung)
- eine automatisierte Prüfungsumgebung wird eingesetzt
- die Programme werden in die Testumgebung eingespannt und halbautomatisch beurteilt (Um die Zeit für das Prüfen zu verringern werden auch Tutoren eingesetzt, die aber keine Noten geben, sondern nur nach formaler Methodik die am Schirm gezeigten Resultate auf Formblätter übertragen)
- nach dem vom Professor und Assistent festgelegten Notenschlüssel werden die Programme dann letztendlich beurteilt.

Vorteile für die Studenten:

- es existiert eine Bespiellösung am Netz, die für die Studenten hilfreich ist, um zu sehen, was genau als Resultat gefordert ist
- das Beurteilen dauert nicht mehr 4 Wochen sondern nur mehr eine Woche
- die gewonnene Zeit kann für die Einsichtnahme genutzt werden

2. ad Einsichtnahme

Die durch die Atomatisierung und durch zusätzliche Unterstützung von Tutoren gewonnene Zeit konnte dazu eingesetzt werden, um mehr Zeit für die Studenten in der Einsichtnahme zu investieren. Dafür entwickelten wir folgendes Modell:

- die Prüfungstestdaten werden auf das Netz gespielt
- jeder Student kann sich sein Programm anhand der Prüfungstestdaten ansehen und seine Fehler bestimmen
- erst nach den Schritten 1 und 2 kommt der Student in die Einsichtnahme
- der Student bekommt 5 Minuten Zeit, die Fehler in Anwesenheit des Assistenten (+der Tester) auszubessern; gelingt ihm das, wird das Programm neu beurteilt. Dieser neue Modus wurde eingeführt, da oft nur kleine Code- änderungen einen Absturz verhindern, der vorher aufgetreten ist.

Daher war es der Versuch einer studentenfreundlichen Art der Einsichtnahme (und nicht eine falsche Beurteilung), die erlaubte, daß sich viele Studenten noch auf eine positive Note verbessern konnten.

3. ad Tutoren

Ich finde es falsch und unfair, die Tutoren als Flaschen zu bezeichnen, da sich viele weit mehr als nur in der vorgeschriebenen Zeit für die Studenten einsetzen. Es kann schon vorkommen, daß ein Student mit einem der Tutoren nicht zufrieden ist, man soll aber nicht gleich auf die Allgemeinheit schließen.

4. ad Kooperation

Wie einige Studenten der Basisgruppe wissen müßten, war ich immer bereit bei auftretenden Problemen gemeinsam eine Lösung zu finden. Dies gilt auch heute noch, und ich bitte daher Herrn Krempler sich wenigstens einmal bei mir einzufinden und seine Probleme zu schildern.

Richard Messnarz

Der Tutor will mich richten

Zu meinem letzten Artikel über Tutoren gab es ungewöhnlich viele und heftige Reaktionen. Zwei negative, ein gutes Dutzend positive. Ich fühle mich zum Teil mißverstanden. Wollte ich aus alter Gewohnheit meinen alljährlichen Schlag gegen die Programmierübungen führen? Nein, diesmal nicht.

Die Programmiermethoden-Übungen sind zum Handkuß gekommen, weil die Situation den meisten bekannt ist und die Probleme am deutlichsten zu Tage treten. Die Programmierübungen sind auch besonders heikel, weil sie den ersten Kontakt der Studierenden zum Tutorensystem darstellen und daher für einige Semester die Einstellung zu Übungen und Tutoren prägen. Alternativen? Als ich zu studieren begonnen habe (was schon einige Jahre her ist) gab es noch keine Tutoren, die Situation war dementsprechend katastrophal. Katastrophaler als im letzten Jahr, sogar katastrophaler als im vorletzten Jahr. Kein Wunder, 2 Assistenten für 200-300 Studierende sind einfach zu wenig.

Abhilfe wurde gesucht.

Und auch gefunden. Zuerst 8 Tutoren, dann 14. Vieles wurde besser, insbesondere die Betreuung der Übungen vor Ort. Die Aufgabenstellungen wurden genauer und nur noch 4-6 mal im Semester geändert. Die Aufgabenstellungen wurden immer früher vor ihrer Bekanntgabe fertig. Eine Woche später, zwei Tage später, gleichzeitig, eine Woche früher... Beispielprogramme wurden zur Verfügung gestellt. Viele dieser Verbesserungen gehen auf Initiative von Herrn Messnarz zurück.

Warum also Kritik?
Weil besser als katastrophal noch lange nicht gut bedeutet. 14 Tutoren sind besser als keine, aber erst Tutoren, die ihr Wissen auch weitergeben können, sind gut. Beispiellösungen sind besser als raten, was gefordert ist. Aber erst dann, wenn die Beispiellösung so gut ist, daß sie positiv bewertet würde, bringt es Nutzen. Es ist schön, wenn es Einsichtnahme gibt, aber erst dann, wenn alle Beteiligten wissen, worum es geht, werden die hohen Ziele erreicht.

Was fehlt mir im System?

Klarheit und Information. Wenn ich mir über die Bedeutung von vorläufigen Noten und zwangsläufiger Einsichtnahme im Klaren bin, kann ich entsprechend handeln. Nur so wird die Einsichtnahme zum Segen statt zum Glücksspiel.

Vorbereitung und Ausbildung. Es kann nicht genügen, zu sagen, daß keine Probleme mehr auftreten können, weil es ja 14 Tutoren und ein Konzept gibt. Sind die Tutoren, auf ihre Aufgabe, Wissen zu vermitteln vorbereitet? Dafür sind sie doch eigentlich angestellt worden, nicht? Oder ist es einfach nur Poker, ob ich an jemanden gerate, der Wissensvermittlung zufälligerweise im Blut hat?

Aufgabentrennung. Gebt den Tutoren, was der Tutoren ist, und den Assistenten, was der Assistenten ist. (Das könnte man noch einige Stufen so weiterführen, ungefähr bis Gott). Es sollte klar sein, was Tutoren tun sollen, und wenn sie es nicht können, gefeuert werden. Ebenso sollte klar sein, was sie nicht zu tun brauchen.

All das fehlt mir nicht nur bei der angesprochenen Übung. **Alle Übungen, die von Tutoren betreut werden, sind an den oben genannten Punkten gefährdet.** Manche mehr, manche weniger, nicht immer wird es so deutlich sichtbar wie bei den Programmiermethoden. Über allen schwebt die beruhigende Aussage "Wir haben etwas unternommen, wir haben Tutoren eingestellt, alles ist in Butter". Jeder Tutor ist ein potentielles Feigenblättchen, das die Sicht auf tiefergehende Probleme verstellt. Bei den Programmiermethoden-Übungen wurde heuer so ein tiefergehendes Problem angegangen. Die Übungsaufgaben wurden überschaubarer, somit die Aufgabenstellung leichter zu erstellen und zu erklären, das Programm leichter zu korrigieren. Weiter so.

Ich möchte nochmals betonen: Die Programmiermethoden liefern nur besonders griffige Beispiele, wie man es gut oder schlecht machen kann. Alle, die Übungen von Tutoren betreuen lassen, sind zum mitdenken eingeladen.

(ak)