Softwaretechnik als Lehrfach – ein Thema, zwei Tagungen, ein Bericht

Jochen Ludewig, Institut für Softwaretechnologie

http://www.iste.uni-stuttgart.de/se/people/ludewig.html

Zusammenfassung

Der Bericht reflektiert und vergleicht zwei Tagungen zum Thema Softwaretechnik als Unterrichtsfach an den Hochschulen, die im Februar und März 2003 stattgefunden haben.

Stichwörter Softwaretechnik-Lehre, CSEE&T, SEUH

Die Tagungen

Der Zufall wollte, dass im Abstand von drei Wochen zwei Tagungen zum Thema "Software Engineering als Unterrichtsfach" stattfanden: die 8. SEUH (Software Engineering im Unterricht der Hochschulen) am 27. und 28.2.2003 in Berlin (Siedersleben, Weber-Wulff, 2003), die CSEE&T 2003 (Conference on Software Engineering Education and Training) vom 20. bis 22.3.2003 in Madrid (Knoke et al., 2003).

Ich habe beide besucht und liefere hier, ohne jeden Anspruch auf Neutralität oder Objektivität, einen kurzen Rückblick.

Die Gemeinsamkeiten

Beide Tagungen

- haben eine lange Tradition und damit einen gewissen engeren Zirkel unter den Teilnehmern;
- waren von etwa 60 bis 80 Leuten besucht;
- haben ein Programm aus referierten und eingeladenen Vorträgen sowie Podiumsdiskussionen geboten;
- haben keine umwerfenden Neuigkeiten gebracht.

Tatsächlich erzeugen Tagungen dieser Art immer wieder das Gefühl der Wiederholung: Wie oft haben wir schon gehört, dass für ein sinnvolles Praktikum sowohl die Betreuungskapazität als auch der Raum im Curriculum fehlen? Wie oft haben wir schon geförderte Projekte gesehen, die gezeigt haben, dass e-learning ein auffällig schlechtes Kosten-Nutzen-Verhältnis bietet? Auf welcher Tagung war nicht beklagt worden, dass Werkzeuge eingesetzt werden sollten, aber allzu teuer sind oder allzu viel Einarbeitungsaufwand erfordern? Diese Liste lässt sich fortsetzen.

Auch, wenn lokal spezielle Schwierigkeiten hinzukomen, trifft die Lehre der Softwaretechnik doch meist auf die gleichen Hindernisse; die wichtigsten sind:

Die übrigen Dozenten demonstrieren in der Regel kein im Sinne des Software Engineerings vorbildliches Verhalten.

Wir lehren Softwaretechnik in einer Umgebung die der Industrie überraschend ähnlich ist. Überall Lippenbekenntnisse zur Qualität, überall praktische Entscheidungen, die das Gegenteil sagen. Und natürlich ist der Grund hier wie dort, dass die Betreffenden letztlich tief und fest glauben, dass es letztlich fast genauso gut ohne geht.

Es soll auch Fälle geben, wo der Bock zum Gärtner gemacht ist und den Lehrstuhl für Softwaretechnik besetzt hält. Man könnte das als feindliche Übernahme bezeichnen.

Studenten sind nicht für die Softwaretechnik motiviert.

Die heute gängigen Grund- oder Einführungsvorlesungen vermitteln keinen Software-Begriff, wir lehren die Informatik nach wie vor bottom-up (mit Betonung auf bottom). Ein simples Programm, das von einer Person entwickelt wird und nicht lange genug eingesetzt wird, um Änderungen und Anpassungen unterworfen zu sein, erfordert (frei nach Parnas) kein Software Engineering. Überall sehen die Studenten, dass die Welt ohne Software-Disziplin auszukommen scheint. Woher soll da die Motivation kommen?

Einen positiven Effekt haben die Eindrücke aus Praktika in der Industrie – wenn die Studenten bereits von positiven Alternativen gehört haben und genötigt werden, ihre Eindrücke zu reflektieren. Sehr oft habe ich von den im wahrsten Sinne Betroffenen gehört, dass sie nach dieser Erfahrung den Sinn des Software Engineerings verstehen.

Der Umfang des möglichen Lehrstoffs ist riesig, ein Konsens über die Prioritäten und Gewichte darin fehlt.

SWEBOK ist ein Versuch, das Fachwissen der Softwaretechnik zu ordnen und zu bewerten (SWEBOK,

2001). Einen Kanon der Lehre kann man daraus kaum ableiten, allenfalls eine Checkliste, ein Haufen von Mosaiksteinen ist noch lange kein Bild.

Dieter Rombach hat in einem eingeladenen Vortrag in Madrid auf sein neues Buch (mit A. Endres) hingewiesen, das die wichtigsten Themen benennt (Endres, Rombach, 2003).

Aber wie die einzelnen Themen zu gewichten sind, wieviel Formales, wieviel Handwerk, wieviel Erfahrungswissen nötig und sinnvoll sind, das bleibt weiterhin offen.

Praktika finden unter höchst unbefriedigenden Randbedingungen statt (Betreuungskapazität, Laufzeit, Aufwand, Praxisnähe der Fragestellungen).

Software-Praktika sollen Erfahrung vermitteln. Dazu brauchen die Projekte eine gewisse Größe, einen echten Kunden, auch Spielräume für Entscheidungen und damit für Fehler. In den meisten Studienplänen ist dafür kein Platz. Außerdem müssen die Praktika intensiv betreut werden. Das ist heute weniger denn je möglich.

In dCurricula mit Bachelor- und Master-Kursen kommen erhebliche Probleme hinzu, weil sich Praktika bei den Undergraduates nur im Mini-Format einplanen lassen; bei den Graduates sind die Probleme nicht geringer, weil die Vorkenntnisse der Teilnehmer ganz unterschiedlich sind. Die Kollegen werden gelb vor Neid, wenn man ihnen vom Stuttgarter Modell mit zwei anspruchsvollen einjährigen Projekten erzählt.

Die Unterschiede

Die Unterschiede zwischen den beiden Tagungen lassen sich sehr übersichtlich durch eine Tabelle darstellen:

	SEUH	CSSE&T
Ort der Veranstaltung	Hochschule	Nobel-Hotel
Dauer	knapp 2 d	2,5 d
Teilnahmegebühren (mit max. Reduktion)	EUR 150,--	$ 400,--
parallele Sitzungen	–	3
Anteil Diskussion	50 %	5 %
kritische Auseinandersetzung	ständig	–

Die Veranstaltung in Madrid war ein Betriebsausflug der US-Szene nach Europa, mit einigen Beiträgen vor allem aus dem Commonwealth und aus Spanien. Große Teile des Programms drehten sich um Fragen, die für uns einfach keine Rolle spielen, vor allem die Anerkennung des Ingenieur-Status in den Staaten der USA und andere Probleme der Akkreditierung. Hinzu kommen all die Probleme eines rigiden Schemas, die wir in den Diplomstudiengängen nicht haben. Die Neigung, Vorträge zu diskutieren oder gar zu kritisieren, war sehr schwach ausgeprägt, sicherheitshalber ließ auch der Zeitplan dafür kaum Raum, man war einfach nett zu einander. Eine solche Konferenz hat aus meiner Sicht kaum Nutzen, sie ist vor allem ein touristisches Ereignis.

Auch auf der SEUH treffen sich verschiedene Kulturen, schon darum, weil hier Universitäten und Fachhochschulen gleichermaßen präsent sind. Auch gehört es zu den Traditionen der SEUH, dass Studenten den Verlauf der Tagung kommentieren – eigentlich eine Selbstverständlichkeit, denn das sind unsere Kunden. Aber diese Vielfalt provoziert lebhafte und kontroverse Diskussionen, so dass selbst schwache Vorträge indirekt oft neue Einsichten bringen.

Faktisch ist also die SEUH für uns das einzige sinnvolle Forum. Wenn es sie nicht seit 1992 gäbe, müsste sie sofort erfunden werden. Und es ist zu hoffen, dass sie bleibt, wie sie bisher war: einfach und billig, wie es unserem Status als Hochschulmenschen entspricht, frisch und ein bisschen aggressiv, offen und unmittelbar an unsere Realität geknüpft. Und eine richtige Hochschule, im Fall der jüngsten SEUH die FHTW im Osten Berlins, womöglich mit bekritzelten Tischen und einem gewissen Maß an Improvisation, ist dafür weit passender als ein poliertes Luxushotel mit bürokratischer Organisation und einem penetranten Streben nach Profit.

Die nächste CSEE&T findet im März 2004 in Norfolk, Virginia statt, die nächste SEUH im Februar 2005 in der RWTH Aachen.

Referenzen

Johannes Siedersleben, Debora Weber-Wulff (Hrsg.) (2003): SEUH 8, Software Engineering im Unterricht der Hochschulen. dpunkt.verlag, Heidelberg, ISBN 3-89864-201-1, 120 S.

Peter Knoke, Ana Moreno, Michael Ryan (2003): Sixteenth Conference on Software Engineering Education & Training. IEEE Computer Society, Order no. PR01869, ISBN 0-7695-1869-9, 342 S.

SWEBOK (2001): http://www.swebok.org/

Albert Endres, Dieter Rombach (2003): A Handbook of Software and Systems Engineering: Empirical Observations, Laws and Theories. Pearson Addison Wesley 2003; 352 pages. ISBN 0-321-15420-7.