Praxis versus Praktika
oder
Die Naivität SE-Ausbildung in der Hochschule

Lutz Prechelt (prechelt@ira.uka.de)
Institut für Programmstrukturen und Datenorganisation
Universität Karlsruhe, Postfach 6980
D–76128 Karlsruhe, Germany
++49/721/608-4068, Fax: ++49/721/694092

(In ähnlicher Form erschienen in: Jörg Raasch, Thomas Bassler (Hrsg.) "Software Engineering im Unterricht der Hochschulen SEUH 92", Workshop des German Chapter of the ACM und der Gesellschaft für Informatik am 27. und 28. Februar 1992 in Stuttgart, B.G. Teubner Stuttgart, German Chapter of the ACM Berichte 37, 1992.)

Ein in der SE-Ausbildung gern zitiertes Faktum ist, daß zwischen 60 und 90 Prozent der Kosten eines Softwareprodukts erst in der Wartungsphase anfallen, also dann, wenn es eigentlich schon "fertig" ist. Sollte es irgendwie gelingen, dank Anwendung von SE-Methoden den anfänglichen Herstellungsprozeß noch zu verbilligen, steigt dieser Anteil (ceteris paribus) sogar noch an. Das Hauptziel von SE muß es folglich sein, die Wartung von Software zu erleichtern.

Diese Sichtweise sollte die SE-Ausbildung natürlich widerspiegeln. Zwar ist es wahr, daß die meisten SE-Methoden in etwa gleichem Maße die Herstellung wie die Wartung eines Produkts erleichtern, aber dies ist durchaus nicht für alle Techniken richtig und der Wartungsgesichtspunkt wird im SE allzu oft nur in Nebensätzen berücksichtigt, anstatt gerade auf solche Techniken abzuheben, die sich vor allem auch in der Wartung positiv bemerkbar machen. Es wird also häufig versäumt, auf die Bedeutung der längerfristigen Nützlichkeit bestimmter Vorgehensweisen gründlich hinzuweisen (Ausnahme z.B. Parnas).

Tatsächlich läßt sich in der industriellen Praxis oft beobachten, daß auch dort, wo bei der Entwicklung eines Produkts einigermaßen konsequent nach SE-Methodik vorgegangen wurde, bei der späteren Wartung die Anarchie schnell Einzug hält und die Entropie in der Software kontinuierlich zunimmt. (Dies ist wohl nur damit zu erklären, daß zu selten die Personen ein System warten, die es auch entwickelt haben, oder das bei der Wartung andere Management- und Kostenrechnungsmaßstäbe angelegt werden als bei der Entwicklung).

Man mag in der theoretischen SE-Ausbildung diesen Mangel an Konsequenz in der Betonung des Wesentlichen noch als tragbar erachten; in der praktischen SE-Ausbildung wird er jedoch zur Todsünde: Die Motivation zur Entwicklung und zum Einsatz des SE insgesamt waren und sind schlechte Erfahrungen mit ad-hoc-Vorgehensweisen. Die Motivation zum Erlernen und Einsetzen von SE-Methoden für eine Einzelperson sollte also idealerweise ebenfalls durch schlechte eigene Erfahrungen erfolgen oder zumindest ergänzt werden. Alternativ wäre ein Aha-Erlebnis der Art "Das hätte ich ohne den Einsatz von SE-Methoden nicht geschafft" als ausreichende Motivation möglich.

Solche Mißerfolgs- oder Aha-Erlebnisse stellen sich jedoch bei den in der Hochschulausbildung üblichen praktischen Ausbildungsteilen keinesfalls von alleine ein: Die Projekte oder Aufgaben sind meist klein genug, um auch ohne sorgfältigen Einsatz von SE-Methoden in der gegebenen Zeit lösbar zu sein oder zumindest den Eindruck solcher Lösbarkeit zu erwecken. Das Praktikum ist dann insofern naiv, als es zwar SE-Methoden einübt, sie aber nicht als Notwendigkeit vermittelt, sondern eher als Selbstzweck erscheinen läßt. Bei Auftreten ernsthaften (Zeit-)Drucks, so ist in Praktika immer wieder zu beobachten, werden folglich auch sogleich fast alle vom SE geforderten Verhaltensweisen stark verwässert oder total ignoriert.

Der Unterschied zwischen Praktika und Praxis besteht vor allem darin, daß es zu einem Wartungserlebnis im Rahmen von Praktika in der Regel nicht kommt. Erst hierbei aber könnte sich der Sinn und Zweck der SE-Methodik wirklich veranschaulichen. Das Erlebnis, sich einmal durch fremden, schlecht strukturierten und falsch oder gar nicht kommentierten Programmkode gewühlt zu haben, zu dem es nur einen unvollständigen Modulführer und keinerlei Entwurfsdokumente (und wahrscheinlich auch keinen Entwurf) gibt stellt wohl die beste Motivation für eine sorgfältige Anwendung der theoretisch erlernten Vorgehensmodelle dar. Wenn das vorliegende Softwaresystem gut entworfen, kodiert und dokumentiert sein sollte, wird es immer noch genügend kleine Mängel geben, an denen sich der Wert des guten Rests eindrucksvoll ermessen läßt.

In dieser Hinsicht ist also die Lehre an den Hochschulen, was die didaktisch geschickte Vermittlung des SE angeht, immer falsch aufgebaut, wenn sie praktische Erfahrungen nicht erzwingt und sogar auch fast immer, wenn sie es tut, falls nämlich dabei eine idealisierte Situation hergestellt wird, die nur die "bessere Hälfte" der Welt zeigt. Diese Analyse läßt sich folgendermaßen zuspitzen:

Behauptung:

Es ist für eine effektive SE-Ausbildung erforderlich, das Erlebnis der Änderung oder Erweiterung eines komplexen fremden Softwaresystems herbeizuführen. Andernfalls wird der Sinn der SE-Methodik nicht verinnerlicht, sondern nur intellektuell erlernt und die Motivation zur Anwendung des Gelernten hält dem Druck von Widrigkeiten in der Praxis nicht stand.

Implementierung:

Es gibt zwei verschiedene Modelle, diese Überlegung umzusetzen. Ein zweiteiliges Praktikum (evtl. über zwei Semester) könnte im ersten Teil ablaufen wie gewohnt, um ein System zu erstellen. Im zweiten Teil sollen dann gleiche oder neugemischte Teams ein zweckgleiches, aber fremdes System erheblich erweitern. Ein einteiliges Praktikum kann von vornherein statt mit Nichts mit einem funktionierenden oder teilfertigen, größeren Softwaresystem beginnen und dieses erweitern. Der Unterschied zwischen beiden Modellen besteht vor allem darin, daß beim zweiteiligen Praktikum das zu erweiternde System ständig mit dem wohlvertrauten eigenen Produkt verglichen werden kann. Vermutlich eröffnet diese Vergleichsmöglichkeit viele Einsichten, die sonst nicht möglich wären.