Ziele und Wege für Softwaretechnik-Praktika

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

(Erschienen in: Jörg Raasch, Thomas Bassler (Hrsg.) "Software Engineering im Unterricht der Hochschulen SEUH 93", Workshop des German Chapter of the ACM und der Gesellschaft für Informatik am 25. und 26. Februar 1993 in Hamburg, B.G. Teubner Stuttgart, German Chapter of the ACM Berichte 38, 1993, pp 78–82.)

Zusammenfassung

Bei der Planung eines an der Hochschule durchzuführenden Praktikums in Software-Engineering gilt es zunächst zu entscheiden, was die hauptsachliche Zielsetzung des Praktikums sein soll. Die zahlreichen Lernaspekte, die von einem solchen Praktikum angesprochen werden können, sind nämlich mit verschiedenen Durchführungsvarianten unterschiedlich gut abzudecken. Eine *bewußte Auswahl* der Schwerpunktsetzung ist daher Voraussetzung für richtige Planung des Praktikums. Dieser Beitrag führt mögliche Ausbildungsziele an und zeigt grundsätzliche Merkmale von Praktika zu ihrer Erreichung auf. Aus der teilweisen Widersprüchlichkeit der Ziele ergibt sich eine Liste von fundamentalen Praktikumsausrichtungen. Diese werden schließlich einer Beurteilung unterzogen.

1 Mögliche Ausbildungsziele

"Wozu dient eigentlich ein Software-Engineering-Praktikum ?". Diese Frage, einem zuständigen Hochschulmenschen gestellt, wird zunächst verständnislose Blicke hervorrufen und dann eine Antwort der Art "Damit die mal praktische Erfahrungen sammeln; die Methoden, die sie theoretisch gelernt haben mal selber anwenden. Und das XYZ CASE-Tool versteht man auch erst, wenn man es mal selber benutzt hat." Und hier beginnt der Schlamassel.

Denn hinter solchen wolkigen Aussagen verbergen sich zahlreiche unterschiedliche Ausbildungsziele, die sich in der begrenzten Zeit eines Praktikums[1] nicht alle zugleich zufriedenstellend erreichen lassen und zwischen denen offensichtlich überhaupt nicht bewußt ausgewählt oder gewichtet wird. Diesem Mißstand soll der folgende Beitrag beginnen abzuhelfen.

Sammeln wir zunächst einmal mögliche Ausbildungsziele eines SE-Praktikums; Vollständigkeit ist dabei zwar angestrebt, aber sicher nicht erreicht. Die Ziele überlappen sich teilweise. Die Frage nach der Zweckmäßigkeit der einzelnen Ziele soll hier *nicht* behandelt werden. Mögliche Lernziele eines Software-Engineering Praktikums können sein:

1. Motivation: Eine positive Einstellung gegenüber der Anwendung von SE-Techniken erzielen.

[1] Unter einem Softwaretechnik- oder Software-Engineering-Praktikum wollen wir im Folgenden eine einsemestrige Veranstaltung mit ca. 4 bis 20 Zeitstunden faktischem Wochenaufwand der Teilnehmenden verstehen.

2. "Demotivation": Die Konsequenzen einer Nicht- oder Falschanwendung von SE-Techniken verdeutlichen.
3. Theoretische Methoden einmal angewendet haben: Examplarisches Ausprobieren möglichst vieler Methoden, etwa nach Art einer Mathematik-Übungssammlung.
4. Theoretische Methoden handwerklich erlernen: Methoden so oft und gründlich anwenden, daß sie bis hin zu einer gewissen Routine erlernt werden.
5. Werkzeuge einmal angewendet haben: Möglichst viele Werkzeuge kennenlernen und ausprobieren.
6. Werkzeugbenutzung handwerklich erlernen: Werkzeuge so intensiv anwenden, daß ihre Eigenschaften und Bedienung vertraut werden.
7. Teamarbeit üben: Kommunikation, Entscheidungsfindung, gegenseitige Hilfe etc.
8. Management üben: Projektplanung, Planungsüberwachung, Resourcenverteilung, Pflichtenverteilung, Entscheidungsfindung ohne Konsens etc.
9. Disziplin lernen: Standards, Planungsvorgaben, Termine etc. einhalten.
10. Projektverantwortung tragen lernen.
11. Kompletten Software-Lebenszyklus selber ausführen.
12. Anforderungsanalyse lernen: Kommunikation mit Anwendern, die aus einem anderen Fachgebiet stammen.
13. Entwurf lernen.
14. Programmieren üben.
15. Dokumentieren lernen.
16. Testen/Fehlersuche üben.
17. Validation: Reviews und Abnahmetests (mit und ohne Anwender) üben.
18. Weg von den Spielzeug-Beispielen: Eine Aufgabe lösen, die so groß und komplex ist, daß sie von einer Person und ohne SE-Methoden nicht mehr in der vorgegebenen Zeit beherrscht werden kann.
19. Kontakt mit der schmutzigen Wirklichkeit. Lernen, wie groß die Unsauberkeiten in einem real existierenden größeren Softwaresystem sind, auch wenn es vermeintlich mit SE-Methoden erstellt wurde. Erkennen, wie groß der Nachholbedarf in der Praxis ist.
20. Arbeiten unter Sachzwängen: Lernen, welche Sachzwänge in einer realen Entwicklungssituation auftreten können, wie mit ihnen umzugehen ist und wie sie sich auf Projektausführung und -ergebnis auswirken.

2 Vorgehensweisen

Für viele der obengenannten Ausbildungsziele liegen die entsprechenden passenden Vorgehensweisen für Planung und Durchführung eines Praktikums auf der Hand, auch wenn es im Einzefall durchaus schwierig ist, konkret z.B. eine geeignete Entwurfsaufgabe zu finden (13) oder eine einigermaßen realistische Anforderungsanalysesituation (nämlich mit "dummen" Anwendern) herzustellen (12).

Für einige Ziele sollen hier stichwortartig einige Aspekte der benötigten Vorgehensweisen angesprochen werden:

1. Die Ziele Teamarbeit, Management und Disziplin (7–9) bilden zwar einen Komplex, erfordern aber durchaus unterschiedliche Ansätze zu ihrer Betonung: Teamarbeit kann am besten betont werden, wenn die Aufgabe in eher kleinen Teams bearbeitet wird und keine sehr ausgeprägte Arbeitsteilung zuläßt und zugleich ein Erfolgsdruck, z.B. mittels Durchführung des Praktikums als eine Art Wettbewerb zwischen mehreren Gruppen mit derselben Aufgabe, hergestellt wird. Dagegen erfordert Managementbetonung eine größere Projektgruppe mit einer formalen Struktur. In beiden Fällen ist außerdem notwendig, daß jede Gruppe zumindest zum großen Teil gemeinsame

Arbeitszeiten für das Praktikum hat. Disziplin kann entweder durch eifrige Gruppenmanager oder durch kontinuierliche Überwachung und Kritik seitens der Praktikumsleiter und -leiterinnen betont werden.

2. Das Tragen von Projektverantwortung (10) ist ein einer Hochschulsituation nur schwer zu erlernen. Die beste Annäherung ist wohl zu erreichen, wenn einerseits die Aufgabe schwierig ist, andererseits das Arbeitsergebnis *tatsächlich* von jemandem benötigt wird.
3. Für den Verzicht auf Spielzeug-Beispiele (18) gibt es zwei Wege: komplette Erstellung eines größen Systems mit einer großen Gruppe oder Erweiterung eines existierenden großen Systems mit einer Gruppe beliebiger Größe.
4. Das Kennenlernen von Sachzwängen (20) und das Tragen von Projektverantwortung (10) kann teilweise erreicht werden, wenn das Praktikum in Kooperation mit einer Firma abgewickelt wird, die die Aufgabe definiert und das Ergebnis gebrauchen kann. Gegebenenfalls kann hierbei auch Bekanntschaft mit einer großen realen Software gemacht werden (19).
5. Die Ziele Motivation und "Demotivation" (1, 2) sind im Grunde nur zwei verschiedene didaktische Herangehensweisen an das selbe Ziel: Software-Engineering soll ernstgenommen werden, die Wichtigkeit und Nützlichkeit seiner Techniken erkannt. Motivation wird erreicht durch ein Erfolgserlebnis, das von den Teilnehmenden auf die Anwendung von SE-Methoden zurückgeführt wird. Um ein solches Erfolgserlebnis zu gewährleisten, muß das Projekt in Umfang und Schwierigkeit sorgfältig ausgewählt und vorbereitet und seine Ausführung ausreichend überwacht werden; ggf. ist vorsichtig helfend einzugreifen. Um Mißerfolge durch Ausfall einzelner Gruppenmitglieder zu vermeiden, sollten die Projektgruppen nicht allzu klein sein.
   Das selbe Ziel kann ex negativo durch ein Mißerfolgserlebnis erreicht werden, das von den Teilnehmenden auf mangelhafte oder fehlende Anwendung von SE-Methoden zurückgeführt wird. Der einfachste Weg dazu ist eine Projektaufgabe, die die Erweiterung eines größeren Softwaresystems vorsieht. Geeignete schlecht entworfene, mangelhaft strukturierte und wenig dokumentierte Softwaresysteme dürften überall in ausreichender Auswahl zur Verfügung stehen. Die Praktikumsbetreuer müssen ihre Begleitung und Hilfestellung so balancieren, daß der Mißerfolg zwar spürbar wird, aber nicht in Frustration und Unlust umschlägt. Beachte, daß Mißerfolg nicht bedeuten muß, daß die Aufgabe nicht gelöst wird, sondern nur, daß übergroße Schwierigkeiten auftreten, die von schlechtem SE herrühren.

Eine wichtige allgemeine Anforderung an das Vorgehen in einem Praktikum ergibt sich aus folgender Überlegung: Da mit dem Praktikum kein Geld verdient wird und auch selten Sanktionen für schlechte Ausführung zu befürchten sind, ist die Neigung groß, die gestellte Aufgabe nicht vollständig ernst zu nehmen und nicht ausreichend Mühe und Sorgfalt in sie zu investieren. Dieser Effekt kann viele Planungen gefährlich unterlaufen und den Lernerfolg in Frage stellen. Die wohl besten Methoden ihn abzudämpfen sind erstens die glaubhafte Aussicht, daß die im Praktikum erstellte Software später auch wirklich eingesetzt werden wird und zweitens eine explizite Wettbewerbssituation zwischen verschiedenen Praktikumsgruppen.

3 Kombinierbarkeit, Widersprüche, Praktikumstypen

Die obengenannten Lernziele und daraus resultierenden Vorgehensweisen in einem Praktikum schließen sich zu großen Teilen gegenseitig aus. Diese Widersprüche lassen sich meist auf die Form Breite versus Tiefe bringen: So schließt die Behandlung von *zahlreichen* Methoden oder Werkzeugen (Breite) aus, daß die Teilnehmenden in ihnen so geübt werden, daß sich sagen ließe, sie hätten die Methoden oder Werkzeuge wirklich "erlernt" (Tiefe); stattdessen muß man eher von "kennengelernt" sprechen. Ähnlich kann eine Phase, sei es Anforderungsanalyse, Entwurf, Programmieren oder Testen, wohl nur dann gründlich geübt werden (Tiefe), wenn man ihren Zeitumfang zuungunsten anderer Phasen vergrößert, statt alle Phasen in einem realistischen Verhältnis einzubeziehen (Breite).

Anders gelagert ist dagegen die Entscheidung zwischen einem direkt motivierenden Erfolgserlebnis als Ziel und einem eher indirekt motivierenden warnenden Mißerfolgserlebnis. Hier findet keine Abwägung zwischen Breite und Tiefe statt, sondern eine Auswahl aus entgegengesetzten didaktischen Mitteln: Einerseits Motivation durch Funktionsbeweis oder andererseits Motivation durch Einsicht in eine Notwendigkeit.

In Anbetracht der beschränkten Zeitreserven in einem Praktikum erscheint es nicht vielversprechend, die Lehrzielsetzung auf einen sehr großen Teil der Lernziele anzulegen; es muß vielmehr eine Untergruppe von möglichst gut kompatiblen Zielen ausgewählt werden.

Folgende Grundtypen von Praktika lassen sich unterscheiden:

1. Lebenszyklus-Praktikum. Es findet keine Schwerpunktsetzung statt. Alle Arbeitsschritte werden in dem für das gestellte Problem angemessenen Rahmen durchgeführt. Ein solches Praktikum leidet an seiner großen Lernzielbreite.
2. Theorieorientiertes Überblickspraktikum. Es werden möglichst viele der in Vorlesungen erlernten Methoden einmal praktisch angewendet. Die Anwendung hat exemplarischen Charakter. Sie kann wegen der geringen Intensität (wg. wenig Zeit pro Methode) weder ein handwerkliches Einüben im Sinne einer gewissen erworbenen Routine, noch eine Anwendung auf Beispiele nennenswerter Größe leisten.
3. Praxisorientiertes Überblickspraktikum. Wie vor, jedoch liegt der Schwerpunkt eher auf Werkzeugen als auf Methoden. Der wissenschaftliche Anspruch wird bewußt reduziert, um nützliches Wissen über Betriebssysteme, Übersetzer, Editoren, Dokumentenverwaltungssysteme, Versionsverwaltungssysteme, Debugger, Funktionsbibliotheken etc. aus SE-Sicht zu vermitteln.
4. Handwerklich orientiertes Praktikum. Hier wird die behandelte Stoffmenge zugunsten von Tiefe reduziert. Das Praktische Arbeiten soll eine Intensität erreichen, die bleibende Fertigkeiten hervorzurufen verspricht. Dazu wird mit Beispielen von praxisnaher Größe gearbeitet. Das handwerkliche Praktikum gibt es wiederum in einer eher methoden- und einer eher werkzeugorientierten Ausprägung.
5. Praktikum zum Erlernen einzelner Fertigkeiten. Sehr starke Betonung eines einzelnen Aspekts in die Tiefe, z.B. Anforderungsanalyse, Entwurf, Implementierung oder Test und ausdrücklicher Verzicht auf Breite.
6. Managementorientiertes Praktikum. Durch Stellung einer einfachen aber umfangreichen Aufgabe werden die organisatorischen und kommunikativen Aspekte betont.
7. Kontakt-mit-der-realen-Welt-Praktikum. Das Hauptanliegen dieses Praktikumstyps liegt darin, den Unterschied zwischen der Software-Engineering-Vorlesung und der Software-Engineering-Praxis zu verdeutlichen, in der Hoffung, daß die Teilnehmenden in Zukunft einen Teil ihrer Anstrengungen darauf konzentrieren, diesen Unterschied zu verringern.

Was also ist zu tun ? Es muß eine *bewußte* Schwerpunktsetzung vorgenommen werden und anhand dieser Setzung ein Praktikum entwickelt werden, das den gewünschten Lernaspekten möglichst weitreichend Rechnung trägt und die parasitären Zeitanteile der weniger angestrebten Lernziele minimiert.

Einen Überblick über mögliche Praktikapläne, der einen Teil der hier geschilderten Aspekte ausdrücklich bespricht, bietet [ST91]. Mein persönliches Votum für die Betonung der Aspekte Demotivation, Disziplin und Kontakt mit der schmutzigen Wirklichkeit findet sich in [Pre92].

4 Kritik der Praktikumstypen

Alle Praktikumstypen müssen sich, zum Teil abhängig vom Typ der Hochschule, Vorwürfe gefallen lassen:

1. Lebenszyklus-Praktikum. Ein solches Praktikum verkörpert zwar den Praktikumsgedanken in Reinform, leidet jedoch angesichts der geringen verfügbaren Zeit erheblich an seiner großen Lernzielbreite.
2. Theorieorientiertes Überblickspraktikum. Eine solche Veranstaltung trüge besser den Namen *Übung* anstatt *Praktikum*; den einzelnen Übungsabschnitten fehlt Zusammenhang.
3. Praxisorientiertes Überblickspraktikum. Der fraglos vorhandene praktische Nutzwert einer solchen Veranstaltung macht sie für Fachhochschulen und Berufsakademien erwägenswert. Dem wissenschaftlichen Anspruch einer Universität wird sie jedoch nicht gerecht.
4. Handwerklich orientiertes Praktikum. Dies ist mehr eine Hülse eines Praktikumstyps, die noch mit anderen Ausrichtungen angereichert werden muß, um ein komplettes Praktikum zu ergeben ("Was machen wir denn nun konkret?"). Die Schwierigkeiten ergeben sich hier nicht aus der (lobenswerten) Hülse, sondern erst aus der Anreicherung.
5. Praktikum zum Erlernen einzelner Fertigkeiten. Hier wäre eher ein entsprechend engerer Name angebracht. Außerdem ist z.B. ein Anforderungsanalyse-Praktikum oder Entwurfspraktikum allenfalls für Universitäten angemessen, ansonsten zu speziell.
6. Managementorientiertes Praktikum. Von einem solchen Praktikum können nicht alle Teilnehmenden in gleichem Maß profitieren. Außerdem müß sich diese Veranstaltung fragen lassen, ob sie erstens in den Bereich der Informatik gehört und ob zweitens ihre Ausrichtung eine *systematische* Vermittlung von Fertigkeiten zuläßt.
7. Kontakt-mit-der-realen-Welt-Praktikum. Sollte dies überhaupt eine Hochschulveranstaltung sein?

Dieser letzte Punkt bringt uns zu einem generellen Problem: Ein grundsätzlicher Kritikpunkt für Softwaretechnik-Praktika an der Hochschule liegt in der Frage "Leistet ein Industriepraktikum nicht wesentlich mehr?". Dies gilt sowohl für Praktika, die versuchen der Praxis ähnlich zu sein (denn was ist der Praxis ähnlicher als die Praxis selbst?), als auch für Praktika, die einen mehr eigenständigen Charakter haben (denn Praxisnähe ist ja ein Grundgedanke von Praktika). Ähnlich wie viele andere Ingenieurdisziplinen könnte auch die Informatik zu dem Schluß kommen, Praxiserfahrung könne nur in der Industrie effektiv erworben werden. Andererseits haben aber ja auch Industriepraktika ihre Tücken, z.B. höherer Zeitbedarf, je nach Firma stark schwankende "Qualität der Ausführung" und Mangel an Kontrolle über die Inhalte.

Natürlich gibt es noch andere Möglichkeiten: Die radikalste besteht darin, auf Praktika gänzlich zu verzichten. Dies ist sicherlich zumindest für Fachhochschulen vollkommen unakzeptabel. Ferner kann man die Grundstruktur der Lehrveranstaltung Praktikum verändern, um einige der Probleme zu beseitigen, z.B. durch Ausdehnung, wie beim Bremer Projektstudium [VS93] oder durch verstärkte Betreuung [Sch93].

Die Frage, ob man ein Praktikum in SE an der Hochschule (im Gegensatz zu: in der Industrie) *überhaupt* für sinnvoll und notwendig hält, muß natürlich zu allererst beantwortet werden und ist sicherlich eine ausführliche Diskussion wert — besonders an den Universitäten, die sich ja sehr oft dem Vorwurf aus der Industrie ausgesetzt sehen, ihre Absolventen seien vollkommen an der Praxis vorbei ausgebildet.

Literatur

[Pre92] Lutz Prechelt. Praxis versus Praktika: Die Naivität der SE-Ausbildung in der Hochschule. In Jochen Ludewig und Kurt Schneider, Herausgeber, *Software Engineering im Unterricht der Hochschulen*, Seiten 96–98, Stuttgart, Germany, 1992. Berichte des German Chapter of the ACM, Band 37, Teubner Verlag.

[Sch93] Kurt Schneider. Auf der Suche nach maßgeschneiderten Unterrichtsformen: Das angeleitete Praktikum. In *diesem Band*, 1993.

[ST91] Mary Shaw und James E. Tomayko. Models for undergraduate project courses in software engineering. Technischer Report CMU-CS-91-174 or CMU-SEI-91-TR-10, Software Engineering Institute and School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, September 1991.

[VS93] Karin Vosseberg und Andreas Spillner. Das KOKS-Projekt: Das Bremer Projektstudium am Beispiel vorgestellt. In *diesem Band*, 1993.