MD7 - SWT III 22. April 2002

Gruppe JOBS (Karsten Kruse, 660218)

I. Schwierigkeiten

Die oftmals geäußerte Hoffnung, dass bei Teamarbeit Anzahl der Mitarbeiter (MA) * Einzelproduktivität der MA = Gesamtleistung der Gruppe ist, erscheint mir grundsätzlich illusorisch. Vergessen wird oftmals, dass ein relativ großer Kommunikationsbedarf durch Teamarbeit entsteht, der bei „Einzelkämpfern" entfällt. Wenn dann noch das Know-how unterschiedlich stark bei den MA ausgeprägt ist, besteht Teamarbeit oftmals aus einer „Einzelkämpfer"-Leistung kombiniert mit ein paar Zuarbeiten der Anderen…

Im universitären Umfeld gestaltet sich meines Erachtens Teamarbeit schwieriger als in der freien Wirtschaft:

a. Es fehlt der (wirtschaftliche) Zwang zur Mitarbeit, niemand riskiert seine Existenz, wenn man nur mangelhaft mitarbeitet. Besondere Leistungen können hingegen nur begrenzt gewürdigt werden (kein Anreiz über Geld bzw. Kariere)
b. In vielen Firmen können Teams auf eine vorhandene, etablierte Infrastruktur zugreifen…Studenten müssen hingegen diese oftmals erst evaluieren und benutzen lernen. Manches muß vielleicht sogar privat „organisiert" werden (z.B. DBMS-Server). Mit Email können die meisten umgehen, der Gebrauch von CVS, VSS, ClearCase, Ant ist den meisten nicht geläufig . Und Tools jeglicher Art sind in der Regel nur dann sinnvoll, wenn jeder Sie verwendet (und verwenden kann).
c. Studenten haben viele Verpflichtungen…andere Vorlesungen, Job(s), Privatleben, Krankheiten Kurieren…. Da es keine festen Arbeitszeiten gibt, ist folgende Arbeitszeiteinteilung eines Dreier-Teams(Studenten A,B,C) möglich:
    i. A ist bereit, immer Freitags von 17-21h zu arbeiten
    ii. B ist momentan krank (mit Attest natürlich), muß dann noch ein Referat erstellen und steht erst die letzten 4 Wochen (und dann natürlich rund um die Uhr) zur Verfügung
    iii. C hat sich in 32 Vorlesungen eingeklebt, muß am Semesterende 28 Klausuren schreiben und will daher das ganze Projekt innerhalb der ersten 4 Wochen abarbeiten.

    Das Problem: Obwohl man bei der Entwicklung moderne Techniken einsetzt, gut betreut wird und z.B. auch eine mehrschichtigen Architektur realisiert, treten bis zur vollständigen Implementierung immer wieder Fragen auf, für deren Beantwortung jeder zur Verfügung stehen muß, um die Arbeit der einzelnen nicht zu gefährden. Viele Fragen entstehen natürlich erst, wenn man seine Aufgaben bearbeitet. Im obigen Beispiel unangenehm für C, wenn sich nach der Implementierung (der Schicht) von B Änderungen im Gesamtentwurf anbieten bzw. erforderlich sind.

II. Kritik

Nun, es ist schwer, ohne den Stein der Weisen Kritik zu üben, da in der freien Wirtschaft die Technik der Softwareentwicklung auch nicht ausgereift erscheint…Vorgehensmodelle wie der RUP bringen wahrscheinlich auch erst nach intensiver Anwendung und Übung irgendeinen nutzen…ohne Implementierungserfahrung ist der Normalstudent mit Analyse und Entwicklung überfordert… Viele Studenten kommen meiner Vermutung nach im Rahmen dieses kurzen Projekts nicht zu der Erkenntnis, das die Verwendung von UML-Diagrammen irgendetwas bringt Für die Meisten ist es eher eine Art Malübung…Es fehlt im Bezug auf die UML die notwendige Praxis…denn wenn ich lange überlegen muß, wie ich etwas ausdrücken will in einer bestimmten Sprache, dann behindert es den Work-Flow und verkommt zu einer Pflichtübung

III. Verbesserungsvorschläge

a. Pflichtfach „UML“ ab den ersten Semester, zudem Anwendung in möglichst vielen Fächern
b. Genaue Überprüfung der Aufgabenteilung unter den Studenten durch den Dozenten; Ggf. Terminsetzung für Zwischenschritte; Übereifrige (Einzelkämpfer) ermahnen, den anderen ihre Arbeit zu lassen. Zurückhängenden Studenten ggf. Termine für Teilaufgaben abverlangen (evemtuell mit Sanktionen…1 Woche später 10% Abzug von der Endnote, 2 Wochen später 20% etc.)…klingt zwar nach „Kindergarten“, verhindert aber das Einzelkämpfer alles vorab fertig machen (und nach dem Ellenbogenprinzip die anderen anschwärzen…)
c. Infrastruktur der TFH verbessern…Labore sollten rund um die Uhr geöffnet bleiben, genügend Laborplätze und Stühle organisiern, Software wie Rational Rose, JBuilder und mehr für alle, Hardware zur Verfügung stellen, auf denen die Software läuft, Kaffee unlimited, und „ältere“ Studenten anheuern, die ggf. mit Ihrer Entwicklungserfahrung als Ansprechpartner zur Verfügung stehen, Aufträge aus der Wirtschaft als Entwicklungaufgabe, Bibliothek sollte mit aktuellen Büchern bestückt werden und…na ja, aber berlin muß halt sparen.