

Der Unified Process im Grundstudium – Didaktische Konzeption, Einsatz von Lernmodulen und Erfahrungen –*

Corina Kopka, Doris Schmedding, Jens Schröder
Lehrstuhl für Software-Technologie
Fachbereich Informatik der Universität Dortmund
{kopka, schmedding, schroeder}@ls10.de

Abstract: Ein umfassendes, lehrveranstaltungsübergreifendes didaktisches Konzept zur Vermittlung eines komplexen Lehrinhalts aus der Softwaretechnik, dem Unified Process, wird vorgestellt. Dazu werden im Grundstudium die traditionellen universitären Lehrveranstaltungen wie Vorlesung, Übung und Praktikum mit Mitteln des eLearnings wie multimediale Lehrbücher und Werkzeuge unterstützt. Dass Vorgehen wird zusammen mit dem Hochschuldidaktischen Zentrum der Universität Dortmund evaluiert.

1 Einführung

Angesichts immer komplexer werdender Softwaresysteme, die hohen Qualitätsanforderungen genügen müssen, ist es wichtig, den Studierenden der Informatik die Notwendigkeit eines strukturierten Prozesses während der Softwareentwicklung zu vermitteln.

Um eine praxisnahe Ausbildung zu gewährleisten, wurde der *Unified Process* (UP) [JBR98] als Vertreter iterativer Vorgehensweisen, die sich in der Industrie bewährt haben, als Lehrinhalt gewählt. Der UP bietet einen generischen Rahmen für die Modellierung iterativer Softwareentwicklungsprozesse, der für jedes Projekt konkretisiert wird. Die bisherige Vorgehensweise im Dortmunder Softwarepraktikum, die einem im wesentlichen sequentiellen Prozessmodell folgt, konnte in diesen Rahmen integriert werden. Im UP wird der Entwicklungsprozess eines Softwaresystems zeitlich in mehrere Zyklen unterteilt, an deren Ende ein neues Produkt-Release freigegeben wird. Jeder Zyklus ist zeitlich in vier Phasen unterteilt, die jeweils durch einen Meilenstein begrenzt werden, an dem über die Fortsetzung des Projekts entschieden wird. Jeder Zyklus wird orthogonal zur zeitlichen Unterteilung in Phasen inhaltlich in *Kernarbeitsprozesse* (engl. *Workflows*) unterteilt. Ein Durchlauf durch diese Arbeitsprozesse ist eine Iteration. Phasen bestehen aus mehreren Iterationen. Innerhalb der Arbeitsprozesse werden Aktivitäten und deren Ergebnisse (Artefakte) festgelegt. Die Gestaltung der Iterationen und der Arbeitsprozesse sind von der Größe und Ausprägung des Projekts abhängig. Diese Anpassbarkeit bietet den Vorteil, dass der UP auch für

*Diese Arbeit wurde gefördert durch das Bundesministerium für Bildung und Forschung unter dem Kennzeichen 08NM098A.

kleinere Aufgabenstellungen/Projekte angewandt werden kann, wie sie im Studium vorkommen. Die Studierenden können damit im Kleinen üben, was sie später im Beruf in umfangreichen Softwareprojekten einsetzen.

Für die Lehre von Softwareentwicklungsprozessen, insbesondere des UP, wurde im Rahmen des Projekts *MuSoft – Multimedia in der SoftwareTechnik* [DE02] die Lerneinheit *Durchführung von Softwareprojekten mit dem Unified Process* entwickelt, deren Konzept hier vorgestellt wird. *MuSoft* soll die Lehre der Softwaretechnik durch den Einsatz von eLearning in verschiedenen Lerneinheiten unterstützen. Ziel dieser Lerneinheit ist, durch aktives Lernen das Bewusstsein der Studierenden für den Softwareentwicklungsprozess zu stärken und durch selbstständiges Planen eines Projektes und die Durchführung dieses Projekts gemäß der Planung Kenntnisse und Fähigkeiten für die Anwendung zu vermitteln.

Im Folgenden wird zunächst auf verwandte Arbeiten eingegangen, bevor das didaktische Konzept der hier präsentierten Lerneinheit vorgestellt wird. Es schließt sich die Umsetzung der Konzeption mit Mitteln des eLearning und die Beschreibung des Vorgehens zur Evaluation der Lerneinheit, die gesammelten Erfahrungen und die Ergebnisse der Evaluation an. Am Ende wird ein Fazit gezogen und ein Ausblick zum weiteren Einsatz gegeben.

2 Verwandte Arbeiten

Aus dem Bereich der Ausbildung im Projektmanagement sind einige Lerneinheiten bekannt, die die aktive Planung von Projekten vorsehen.

Für die Ausbildung von Projektleitern wird an der Universität Stuttgart ein System zur interaktiven Simulation von Softwareprojekten eingesetzt (SESAM-Projekt). Unter Beachtung von Vorgaben wie geschätzter Projektumfang, zeitlicher Rahmen, verfügbares Budget und Qualitätsanforderungen wird ein Projekt geplant und anschließend am Simulator durchgeführt. Erst der Einsatz des Systems in einem simulationsbasierten Ausbildungskonzept, das neben der Nutzung des SESAM-Systems als eigenständiges Lehrmittel insbesondere die Analyse der geplanten Projektverläufe zusammen mit Tutoren vorsieht, brachte den gewünschten Lernerfolg [MS01].

In der *MuSoft*-Lerneinheit *Projektmanagement* [Ke02] werden für Projektplanungszwecke Methoden der Aufwandschätzung und Netzplantechniken gelehrt. Zur Visualisierung von Netzplanberechnungen auf Basis von Vorwärts- und Rückwärtsverkettung wird ein Netzplansimulator eingesetzt, der für konkrete Beispiele die auszuführenden Rechenschritte anzeigt.

Ziel des Praktikums *Management von Informatikprojekten* am Institut AIFB der Universität Karlsruhe ist, die verschiedenen Dimensionen eines Informatikprojekts wirklichkeitsnah erleben zu können [Ri99]. Die inhaltlichen Schwerpunkte liegen auf Managementaspekten und in geringem Maße auf Softwareentwicklung. Die Teilnehmer des Praktikums lernen, ein Projekt zu planen, abzuarbeiten, zu verfolgen und zu steuern. Dazu wird ein fiktives Unternehmen erfunden und den Teilnehmern ein ausführliches Projektumfeld zur Verfügung gestellt. In Projektsitzungen wird über den Projektfortschritt berichtet und in Feedback-Runden Inhalt und Ablauf der Projektsitzungen analysiert.

In allen Lerneinheiten steht die Ausbildung von Projektmanagern im Vordergrund. Dabei setzen die Ausbildungskonzepte in Stuttgart und Siegen zur Wissensvermittlung Simulationen ein. Lernen erfolgt durch Feedback mit Tutoren (Stuttgart, Karlsruhe) oder durch Visualisierung auszuführender Rechenschritte (Siegen).

3 Didaktische Konzeption

Im Unterschied zu den oben vorgestellten Projekten unterstützt unsere *MuSoft*-Lerneinheit die Studierenden in der Projektplanung auf Basis des UP und begleitet sie in der Softwareentwicklung. Im Vordergrund der Planung stehen nicht die Vorgabe eines verfügbaren Budgets oder eine Kosten- und Aufwandschätzung, sondern die Identifizierung sinnvoller Aktivitäten im Entwicklungsprozess, ihre sorgfältige Einordnung innerhalb eines zeitlichen Rahmens und die Planung von sinnvollen Iterationen, also insbesondere die richtige Anwendung des Unified Process als generisches Modell für spezifische Projektzwecke.

Um diese Planungskenntnisse zu vermitteln, wird der Entwicklungsprozess nicht nur in der Vorlesung sondern auch in einem Praktikum thematisiert, weil für die Studierenden aufgrund ihrer bisherigen Erfahrungen die Notwendigkeit des strukturierten Vorgehens in der Softwareentwicklung noch nicht erkennbar ist. Ein Praktikum hat den Vorteil, dass durch konkrete Projekte abstrakte Vorgehensmodelle viel anschaulicher vermittelbar sind.

In einem Softwarepraktikum ist typischerweise der Einsatz bereits bekannter Sprachen und Methoden für Analyse, Entwurf, Implementierung und Test vorgesehen. Darüberhinaus werden aber auch genaue Kenntnisse über Softwareentwicklungsprozesse benötigt. Diese Kenntnisse werden aber oft nicht explizit gelehrt. In der Regel werden Prozessmodelle vorgegeben, die Planung von Prozessen, ihre Befolgung und evtl. eine Änderung des Prozessplans während der Projektdurchführung sind hingegen nicht vorgesehen. Auch im Dortmunder Softwarepraktikum, in dem zwei Projekte durchgeführt werden, wurde bisher die Projektplanung von den Praktikumsveranstaltern vorgegeben [Sc01]. Für die Wissensbildung stufen wir aber eine aktive Auseinandersetzung (aktives Lernen) mit dem Prozess als sehr wichtig ein. Dies soll im Rahmen der vorhandenen Veranstaltungsformen zur Lehre der Softwaretechnik (Vorlesung mit Übung, Praktikum) geschehen. Neben Materialien für eine theoretische Einführung in einer Vorlesung bietet unsere Lerneinheit daher Unterstützung, um den Entwicklungsprozess anhand des Beispiels UP in einem Softwarepraktikum zu lehren und durch konstruktives Tun seitens der Studierenden zu erlernen. Der Einsatz unserer Lerneinheit basiert auf einer engen inhaltlichen Verzahnung zwischen Softwaretechnikvorlesung/-übung und Softwarepraktikum.

Neben einer umfassenden Einführung in Vorgehensmodelle in der Softwareentwicklung legt unser Lernszenario den Schwerpunkt darauf, dass Studierende ihr neu erworbenes Wissen über den Softwareentwicklungsprozess konstruktiv anwenden und Erfahrungen sammeln. Die Studierenden sollen daher den Entwicklungsprozess für Softwareprojekte selbst modellieren und durchführen und werden dabei mit den Möglichkeiten des eLearning durch den Einsatz von Softwarewerkzeugen unterstützt. Idealerweise sollte die Planung für ein Beispielprojekt vorgegeben werden, an dem sich die Studierenden bei der

Planung orientieren können. Im Dortmunder Softwarepraktikum wurde daher das Prozessmodell für das erste Projekt (Bibliotheksverwaltung) vorgegeben. Im zweiten Projekt (Halma-Spiel) modellierten die Studierenden selbstständig mit Hilfe des Betreuers den Entwicklungsprozess. Die Planung wurde während der Durchführung des Projekts von den Studierenden überprüft. Unter Umständen liegen triftige Gründe vor, den Plan zu ändern. Unabhängig von der Güte des modellierten Entwicklungsprozessplans sollen die Studierenden damit angeregt werden, über den Prozess und ihr eigenes Handeln zu reflektieren und Erfahrungen für die Planung eines zukünftigen Projektes zu sammeln.

Lernen folgt damit neuen konstruktivistischen didaktischen Ansätzen ([vG98]), die vom reinen Informationsvermittlungsprozess weg- und zu Prozessen der Informationssuche und Wissenskreation hinführen. Mit der eigenständigen Planung eines neuen Projektes wird Wissen geschaffen, indem vorhandenes Wissen angewandt und darüber reflektiert wird, um einen Entwicklungsprozess konstruktiv zu gestalten.

4 Umsetzung der didaktischen Konzeption

Das didaktische Konzept der Lerneinheit sieht ein dreiphasiges Vorgehen vor: Nach einer theoretischen Einführung im Rahmen einer Vorlesung mit Übung wenden die Studierenden im Praktikum ein für sie geplantes Prozessmodell an, ehe sie selbstständig den Prozess für ein weiteres Projekt planen und es entsprechend durchführen. So setzen sich die Studierenden aktiv mit den vermittelten Inhalten auseinander. Um die Studierenden bei ihrem aktiven Lernprozess zu unterstützen, wurden Mittel des eLearnings in die konventionellen Veranstaltungsformen integriert. Dabei ist die Lerneinheit in drei inhaltliche Blöcke, so genannte Lernmodule, organisiert, die jeweils auf die speziellen Aufgaben einer Phase des Vorgehens abgestimmt sind.

Bei der theoretischen Vermittlung des UP werden die Studenten durch ein multimediales Lehrbuch unterstützt. Dieses Lernmodul ermöglicht es den Studierenden, das in der Vorlesung vermittelte Wissen selbstständig nachzuarbeiten und durch zusätzliche Materialien zu vertiefen. Um die Studierenden bei der praktischen Anwendung und Vertiefung ihres Wissens über den UP in der Übung und im Softwarepraktikum zu unterstützen, werden in zwei weiteren Lernmodulen Werkzeuge bereitgestellt: ein Prozessmodellierer, mit dem die Studierenden selbstständig einen Prozess für die Durchführung ihres Projekts planen können, und ein Projektutor, der sie bei der Durchführung eines Projekts unterstützt. Nachfolgend werden die drei Lernmodule im einzelnen vorgestellt.

4.1 Lernmodul *Multimediales UP-Lehrbuch*

In dem Lernmodul *Multimediales UP-Lehrbuch* wird den Studierenden neben allgemeinen Vorgehensmodellen der Unified Process mit multimedialen Techniken wie die Verlinkung von Inhalten und der Einsatz von Animationen vorgestellt. Es untergliedert sich in drei Bereiche: einer Einführung in den UP in Folienform, den theoretischen Grundlagen des

UP und einem Praxisbeispiel für ein Softwareprojekt, dessen Planung und Durchführung mit dem UP ausführlich dokumentiert wird.

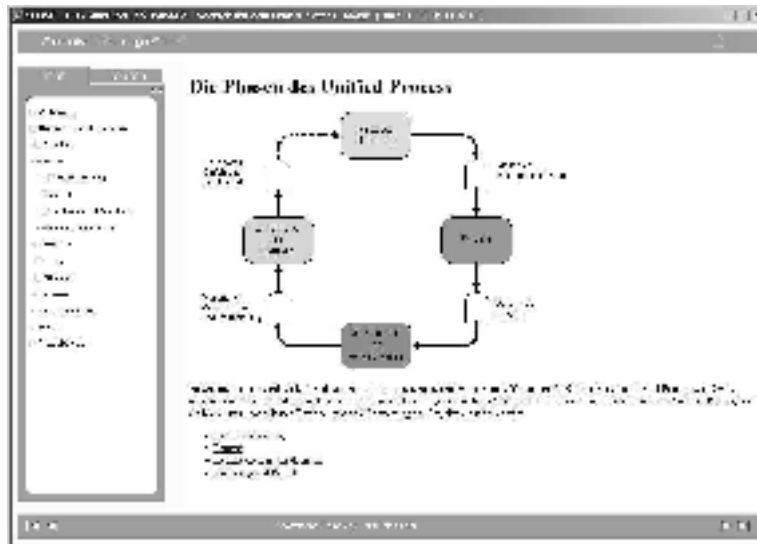


Abbildung 1: Das multimediale Lehrbuch

In Abb. 1 ist beispielhaft eine Seite des Lehrbuchs zu sehen. Es wurde ein Navigationsrahmen entwickelt, in dem die Inhalte des UP-Lehrbuchs eingebettet sind. So können die Studierenden einfach und übersichtlich durch die Inhalte des Lehrbuchs navigieren, unterstützt durch ein (aufklappbares) Inhaltsverzeichnis, hierarchische und sequentielle Vor- und Zurück-Funktionen und eine History. Die drei inhaltlichen Teile des Lehrbuchs, insbesondere die theoretischen Grundlagen und das Praxisbeispiel, sind durch Links inhaltlich eng miteinander in Beziehung gesetzt, um den Bezug der theoretischen Inhalte mit ihrer Anwendung - und umgekehrt - deutlich zu machen. Desweiteren können so genannte geführte Touren definiert werden, in denen ausgewählte Inhalte aus dem Lehrbuch in einer strikt sequentiellen Folge präsentiert werden. So haben die Studierenden mehrere Möglichkeiten, sich den Inhalt anzueignen und zu vertiefen: Sie können selbstständig den Inhalt des Lehrbuchs in einer von ihnen festzulegenden Reihenfolge frei erkunden oder thematisch vorausgewählten Navigationspfaden folgen.

4.2 Lernmodul *Prozessmodellierer*

Für die Definition eines Prozessmodells nach dem UP wird im Lernmodul *Prozessmodellierer* ein grafischer Editor bereitgestellt, mit dem ein Prozess für ein konkretes Softwareprojekt instanziiert werden kann.

Der Modellierer nimmt dabei eine Anpassung des UP an die besonderen Bedürfnisse eines Softwarepraktikums vor: er erlaubt es, einen Prozess zu instanziiieren, der sich auf die fünf

für die Softwareentwicklung wesentlichen Kernarbeitsprozesse konzentriert, von der Erfassung der Anforderungen bis zum Testen. Arbeitsprozesse, die für den Einsatz des UP in einer Lehrveranstaltung weniger von Bedeutung sind, wie etwa die Geschäftsprozessmodellierung und die unterstützenden Prozesse, finden keine Berücksichtigung. Desweiteren kann der Modellierer mit den im Projekt auftretenden Rollen konfiguriert werden.

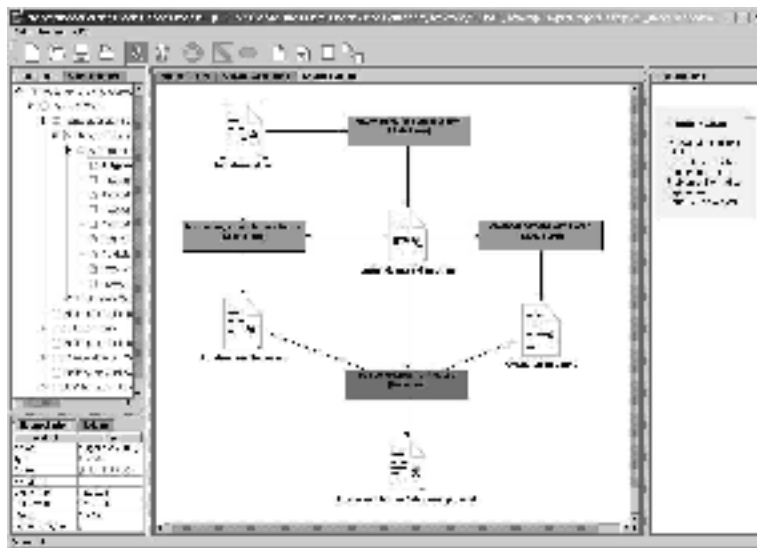


Abbildung 2: Der Prozessmodellierer

Bei der Instanziierung wird der Prozess für das durchzuführende Projekt gemäß den Vorgaben des UP strukturiert und die einzelnen Aktivitäten und zu erstellenden Artefakte in Beziehung gesetzt und zeitlich geplant (s. Abb. 2). Der Prozessmodellierer stellt dabei eine syntaktische Unterstützung zur Verfügung und bietet durch eine übersichtliche Darstellung der einzelnen den UP innewohnenden Hierarchieebenen eine gute Orientierungshilfe. Desweiteren erlaubt der Modellierer es, die einzelnen Elemente des zu erstellenden Prozessmodells zu annotieren. Diese Annotationen werden zusammen mit dem Prozessmodell persistent gemacht.

4.3 Lernmodul *Projektutor*

Da die Studierenden den modellierten Prozess im Rahmen des Praktikums auch anwenden sollen, wird mit dem Projektutor ein Werkzeug bereitgestellt, mit dem der aktuelle Projektfortschritt verfolgt werden kann. Der Prozesstutor arbeitet auf der Basis des mit dem Prozessmodellierer instanziierten Prozessmodells. Der Tutor legt eine im Rahmen von Lehrveranstaltungen übliche, artefaktzentrierte Sichtweise zu Grunde, denn von den Studierenden werden Artefakte erstellt, abgegeben und von den Betreuern korrigiert.

Der Tutor bietet verschiedene Sichten auf den Prozess an (s. Abb. 3). Neben einer in der



Abbildung 3: Der Projektmentor

Software-technisch übliche Darstellung des zeitlichen Ablaufs des Prozesses in Form eines Gantt-Charts [GJM03] wird zur Auswahl der als nächstes auszuführenden Tätigkeiten ein Aktivitätsbaum angeboten, der eine hierarchische Sicht auf den Prozess liefert. Dabei werden die Abhängigkeiten, die zwischen den auszuführenden Aktivitäten mit ihren zu erstellenden Artefakten existieren, und der aktuelle Projektfortschritt erfasst und visualisiert, und davon abhängig für die einzelnen Prozesselemente Todo-Listen erstellt. Desweiteren steht eine Rollensicht zur Verfügung, bei der den einzelnen Rollen ihre auszuführenden Tätigkeiten zugeordnet sind.

Anhand der Artefakte wird der Projektfortschritt verfolgt. Dabei werden die Studierenden aktiv mit einbezogen: Sie tragen im Tutor ein, in welchem Bearbeitungsstatus (*unbearbeitet*, *in Bearbeitung* und *erledigt*) ein Artefakt ist. Die davon abhängigen Informationen und Visualisierungen werden immer entsprechend aktualisiert. Da dieses manuelle Eintragen von den Studierenden in der hierarchischen Sicht auf das Prozessmodell vorgenommen wird, haben sie immer den unmittelbaren Rückbezug zum Prozess und können ihre ausgeführte Tätigkeit im Projektablauf richtig einordnen. Ziel ist es dabei, die Studierenden zur Reflexion über ihr Handeln anzuregen und dadurch ihr Wissen über den UP zu vertiefen.

4.4 Realisierung der Lernmodule

Bei der Realisierung der Lernmodule sind eine Reihe von Anforderungen zu berücksichtigen: Zum einen ist es wichtig, dass sie plattformunabhängig einsetzbar sind. Der von

ihnen angebotene Funktionsumfang sollte auf die Bedürfnisse der Lernenden zugeschnitten sein und keine überflüssige Funktionalität anbieten, um die Studierenden nicht abzulenken. Um eine angemessene Arbeitsgeschwindigkeit zu gewährleisten, sollten sie keine zu hohen Anforderungen an die Hardwareressourcen stellen. Deshalb wurde vom Einsatz kommerzieller Werkzeuge Abstand genommen, wie etwa Rational Software von IBM für professionelle Softwareentwickler, sondern stattdessen wurden eigene Werkzeuge entwickelt.

Das multimediale Lehrbuch ist als HTML-Anwendung auf jeden Standardbrowser ausführbar. Mit Hilfe von Javascript und der Einbindung von Flash-Animationen konnte alle benötigte Funktionalität integriert werden. Der Modellierer und Tutor wurde mit Hilfe eines am Lehrstuhl für Softwaretechnologie der Universität Dortmund entwickelten Java-Frameworks für Modellierungswerkzeuge [APS04] realisiert, das speziell auf die Bedürfnisse für den Einsatz in der universitären Lehre ausgerichtet ist.

4.5 Einsatz der Lernmodule

Nach der theoretischen Einführung des UP in der Softwaretechnikvorlesung und dem ersten Einsatz in der zugehörigen Übung wurden im zeitlich direkt darauf folgenden Softwarepraktikum zwei Projekte basierend auf dem UP durchgeführt. Für das erste Projekt wurde analog zum bisherigen Vorgehen im Softwarepraktikum [Sc01] eine Planung der Veranstalter vorgegeben. Das Prozessmodell wurde mit dem Modellierer erstellt und mit ausführlichen Annotationen versehen, in denen die vorgesehenen Tätigkeiten und Artefakte erläutert wurden. Außerdem stand den Praktikums TeilnehmerInnen das multimediale UP-Lehrbuch und die UP-Werkzeuge, Prozessmodellierer und Projektutor, zur Verfügung.

Das zweite Projekt sollte basierend auf den Erfahrungen des ersten Projekts von den Praktikums TeilnehmerInnen selbst geplant werden. Als Aufgabe wurde ein Spielprogramm gewählt. Im ersten Zyklus wurde die Mensch-zu-Mensch-Version des Spiels realisiert, im zweiten Zyklus wurde das Spiel um einen simulierten Computergegner erweitert.

5 Evaluation, Ergebnisse und Erfahrungen

Die Einführung des UP in einer frühen Phase des Studiums ist mit Risiken verbunden, da der UP üblicherweise nicht zum Lehrstoff des Grundstudiums gehört. Deshalb ist es besonders wichtig, für unser Vorgehen einen ständigen Verbesserungsprozess zu etablieren, der auf einer wissenschaftlich fundierten Evaluation zur Überprüfung unseres didaktischen Konzepts basiert. Da für die Evaluation von eLearning-Materialien jedoch bisher keine für unsere Zwecke geeigneten methodischen Ansätze bekannt sind, wurden wir vom Hochschuldidaktischen Zentrum der Universität Dortmund unterstützt [KMGSD04]. Es wurde ein dreistufiges Vorgehen gewählt: zuerst eine Vorevaluation (Pretest), danach eine Evaluation in der Vorlesung Softwaretechnik im Grundstudium und abschließend eine Evaluation

im Softwarepraktikum im Grundstudium.

Ziel des **Pretests** war, die Lernmodule schon im Vorfeld der Lehrveranstaltung zu evaluieren, um gravierende Fehler insbesondere bei der Funktion der Werkzeuge (Prozessmodellierer und Tutor) auszuschließen und genügend Zeit für u.U. aufwändige Korrekturen zu haben. Dieser Pretest wurde mit Studierenden in 3 Gruppen mit 2-3 Personen an zwei Evaluationstagen durchgeführt: Die Evaluation wurde anhand einer konkreten Projektaufgabenstellung zur Modellierung eines Entwicklungsprozesses vorgenommen. Anschließend wurde die Projektdurchführung simuliert, indem die TeilnehmerInnen die zu erstellenden Artefakte in der geplanten Reihenfolge als erledigt kennzeichneten. Die Evaluation erfolgte durch Beobachtung der Studierenden bei der Arbeit und einer anschließenden Befragung der Gruppen. Die Studierenden konnten während der Nutzung der Lernmodule jederzeit Feedback geben und Fragen stellen.

Obwohl beim Pretest die Durchführung eines Projektes nur simuliert wurde, konnten durch die Erkenntnisse bei der Nutzung der Werkzeuge einige funktionelle Verbesserungsvorschläge identifiziert werden. Generell wurden die Werkzeuge von den Studierenden überwiegend positiv eingeschätzt.

Die **zweite Stufe der Evaluation** wurde während des Einsatzes des Lehrbuchs und des Prozessmodellierers im Wintersemester 2003/2004 in der Softwaretechnikvorlesung und -übung im Grundstudium durchgeführt. Der Unified Process wurde in der Vorlesung vorgestellt und als Übung anhand einer konkreten Projektaufgabenstellung ein Prozess modelliert. Die Evaluation erfolgte durch die Auswertung eines umfassenden Fragebogens. 287 ausgefüllte Fragebögen liegen vor.

Die Antworten bestätigen, dass sich die Konzepte des UP in Form einer Vorlesung nur schwer vermitteln lassen. Während immerhin 39 Prozent der Studierenden angeben, den Nutzen des UP verstanden zu haben, ist 45 Prozent die Anwendung des UP nicht klar geworden. Daher wird auch die Übungsaufgabe nicht gut bewertet. Viele Studierende hatten die Aufgabenstellung nicht richtig verstanden, was sich darin äußert, dass 44 Prozent der Befragten Vorlesung und Übung zum UP nicht gut aufeinander abgestimmt finden. Weniger die Erklärungen in der Übungsgruppe, als vielmehr das eigenständige Arbeiten mit dem UP scheint den Studierenden die Funktionsweise des UP klar gemacht zu haben.

Das UP-Lehrbuch wurde häufig zum vertiefenden Selbststudium der Inhalte beim Lösen der Übungsaufgabe verwendet: 66 Prozent der Studierenden nutzten das Praxisbeispiel im UP-Lehrbuch häufig, 22 Prozent immerhin einmal. Dabei wurde das Praxisbeispiel im UP-Lehrbuch umso hilfreicher zum Lösen der Übungsaufgabe empfunden, je intensiver die Studierenden das UP-Lehrbuch verwendet haben. Generell bewerten es 38 Prozent der Studierenden als hilfreich; investierten sie jedoch mehr als zwei Stunden Zeit, waren es sogar 56 Prozent, dagegen nur 30 Prozent bei den Studierenden, die weniger als eine halbe Stunde investiert haben. Zum Erstellen eines Prozessmodells für ein Entwicklungsprojekt wurde der Modellierer allgemein als nützlich bewertet (11 Prozent: sehr guter Nutzen, 33 Prozent: guter Nutzen, 27 Prozent: zufriedenstellender Nutzen).

Generell zeigt sich, dass Studierende im Grundstudium, die niemals zuvor ein halbwegs komplexes Softwareprojekt durchgeführt haben, anscheinend überfordert sind, die Inhalte zu verstehen und diese eigenständig anzuwenden. Insbesondere der generische Charakter

des UP macht es den Studierenden schwer, eine Anschauung zu entwickeln.

Im anschließenden Praktikum mit zwei Projekten kamen wieder das multimediale UP-Lehrbuch und die beiden UP-Werkzeuge Prozessmodellierer und Projektutor zum Einsatz. In dieser **dritten Stufe der Evaluation** wurden wieder Fragebögen und Interviews mit ausgewählten Studierenden verwendet. Am Software-Praktikum nahmen 72 Studierende teil, die in neun Projektteams aufgeteilt waren.

Unserer didaktischen Konzeption folgend war im ersten Projekt der Ablauf vorgeplant. Es waren 5 Iterationen vorgesehen: zwei in der Phase *Konzeptualisierung* und jeweils eine Iteration in den weiteren Phasen. Die PraktikumssteilnehmerInnen konnten sich mit Hilfe des Prozessmodellierers Einblicke in das vorgegebene Prozessmodell verschaffen. Im Modell sind für jeden Arbeitsprozess detailliert die Abhängigkeiten zwischen den einzelnen Tätigkeiten und Artefakten beschrieben. Abb. 2 zeigt z.B. den Anforderungsworkflow in der Phase Konzeptualisierung. Bei der Durchführung bietet der Projektutor (vgl. Abb. 3) mit seinem Gantt-Chart den TeilnehmerInnen einen guten Überblick über die Zeitplanung des Projekts und verdeutlicht durch das Abhaken der erstellten Artefakte im Aktivitätsbaum gut den Projektfortschritt, was zur Motivation der TeilnehmerInnen beiträgt.

Das zweite Projekt wurde von den TeilnehmerInnen selbstständig unter Anleitung der Betreuer geplant. Mit dem Modellierer wurde ein entsprechendes Prozessmodell erstellt, das, wie in der Aufgabenstellung vorgegeben, aus zwei Entwicklungszyklen (zunächst ein Mensch-gegen-Mensch-Spiel, danach das Spiel mit simuliertem Computergegner) bestand. Innerhalb eines Zyklus orientierten sich die Studierenden weitgehend an den Vorgaben des Prozessmodells des ersten Projekts. Unterschiede zum ersten Projekt ergaben sich z.B. in der Integration von Testmethoden in die Implementierung, dem Zeitpunkt der Erstellung des Benutzungshandbuchs und insbesondere in der Länge der vorgesehenen Zeiten für die Aktivitäten.

Unser Ansatz, für das erste Projekt ein Prozessmodell vorzugeben und es im zweiten Projekt von den TeilnehmerInnen selbstständig erstellen zu lassen, um so schrittweise die Fähigkeit einer selbstständigen Planung eines Softwareprojektes zu vermitteln, war erfolgreich. 68 Prozent der TeilnehmerInnen fanden es sehr hilfreich für das zweite Projekt, dass beim ersten Projekt die Prozessmodellierung vorgegeben war, weitere 24 Prozent teilweise hilfreich. Entgegen den Erfahrungen mit der Übungsaufgabe fühlten sich 89 Prozent der TeilnehmerInnen mit der eigenständigen Planung im zweiten Projekt nicht überfordert.

Insgesamt konnten wir beobachten, dass die Studierenden in der Lage waren, einen Entwicklungsprozess zu planen, ihre Planung zu präsentieren, ein Projekt gemäß der Planung durchzuführen und ihre Erfahrungen zu reflektieren. Dies deckt sich mit der eigenen Einschätzung der Studierenden über die im Softwarepraktikum erworbenen Kenntnisse. 68 Prozent der TeilnehmerInnen schätzten ihre Kenntnisse, Softwareprozesse zu modellieren, als gut ein, weitere 6 Prozent als sehr gut. 61 Prozent sind der Meinung sich gute Kenntnisse in der Entwicklung konkreter Projekte angeeignet zu haben, 14 Prozent sogar sehr gute Kenntnisse. Ihre Fähigkeit, ein Softwareprojekt zu organisieren, schätzten 56 Prozent als gut und 11 Prozent als sehr gut ein.

Die Beobachtungen und Befragungsergebnisse bestätigen deutlich unser didaktisches Konzept, dass Prozessmodellierung als Inhalt in der Softwaretechnikausbildung nur Sinn macht,

wenn Prozessmodellierung mit Softwareentwicklungsprojekten verknüpft wird. Eine Übung im herkömmlichen Sinn kann nur sehr begrenzt Erfahrungswissen vermitteln, da die Planung eines Projekts ohne seine tatsächliche Durchführung die Schwachstellen der Planung nicht deutlich werden lässt. Erst die eigene Projektdurchführung in der Praktikumsgruppe ermöglicht den zielgerichteten Einsatz des Wissens. Dabei boten die eingesetzten Werkzeuge durch die explizite Darstellung des Entwicklungsprozesses eine gute Unterstützung für die Lernenden und Lehrenden. Anhand der übersichtlichen Darstellung im Modellierer und Tutor konnte gut über den Prozessablauf und den Fortschritt kommuniziert werden. Die Werkzeuge unterstützten die Studierenden aufgrund ihrer angebotenen Funktionalität bei der Anwendung des vermittelten Wissens.

6 Zusammenfassung und Ausblick

Das Ziel, mit dem UP einen komplexen Lehrinhalt aus der Softwaretechnik im Grundstudium zu vermitteln, ist neu und stellt einen hohen Anspruch dar. Um dies erfolgreich umzusetzen, wird in dem vorgestellten didaktischen Konzept ein neuer Ansatz gewählt, bei dem zum einen der Lerninhalt den Studierenden lehrveranstaltungsübergreifend vermittelt wird. Zum anderen soll die Wissensvermittlung durch aktives Lernen erfolgen, um so den Studierenden den Unified Process nachhaltig näher zu bringen. Zur Unterstützung wird dabei auf die Möglichkeiten von eLearning zurückgegriffen, indem verstärkt Werkzeuge zur Wissensvermittlung in traditionelle Lehrveranstaltungsformen wie Vorlesung, Übung und Praktikum eingesetzt werden.

Um die Tragfähigkeit des didaktischen Konzepts zu überprüfen und Verbesserungspotenziale zu entdecken, ist es notwendig, den Einsatz der Lerneinheit zu evaluieren. Da der Einsatz von eLearning in der Literatur bisher noch nicht systematisch untersucht wurde, ist in Zusammenarbeit mit dem Hochschuldidaktischen Zentrum der Universität Dortmund ein Vorgehen zur Evaluation entwickelt worden.

Die Ergebnisse der Evaluation geben sowohl dem lehrveranstaltungsübergreifenden Ansatz als auch dem Einsatz von Werkzeugen Recht. Alleine das Lehren der theoretischen Konzepte des UP im Rahmen einer Lehrveranstaltung reichte bei den meisten Studierenden nicht aus, um ihr Wissen selbstständig im Rahmen einer Übungsaufgabe anzuwenden. Erst durch die Anwendung des Wissens im anschließenden Softwarepraktikum konnte bei den Studierenden nach deren Selbsteinschätzung und unseren Beobachtungen ein Verständnis für die Prozessmodellierung erreicht werden. Dabei wurde die Werkzeugunterstützung durch die drei zur Verfügung gestellten Lernmodule *multimediales UP-Lehrbuch*, *Prozessmodellierer* und *Projekttutor* von den Studierenden als sehr hilfreich angesehen. Mit Hilfe der Werkzeuge wird die Kommunikation im Entwicklungsteam und das Verständnis gefördert, was zur Verbesserung der Qualität der Lehre beiträgt.

Aufgrund dieser positiven Erfahrungen ist im nächsten Veranstaltungskanon der Softwaretechnik bestehend aus Vorlesung mit Übung und anschließendem Praktikum wieder vorgesehen, mit Hilfe unserer Lerneinheit den UP zu vermitteln. Dabei werden die durch die Evaluation eruierten Verbesserungsmöglichkeiten in die Lernmodule integriert.

Weiterhin ist angedacht, die entwickelten Lernmodule auch in Veranstaltungen des Hauptstudiums, wie etwa Projektgruppen, einzusetzen. Gerade durch den wiederholten Einsatz der Lernmodule im Curriculum rechtfertigt sich der Aufwand, der in ihre Erstellung und in die Einarbeitung bei ihrer Benutzung investiert werden muss. Die Lernmodule stehen auch Interessenten anderer Hochschulen im Portal des *MuSoft*-Projekts (<http://www.softwartechnik.de>) kostenlos zur Verfügung.

Danksagung Die Autoren danken der Arbeitsgruppe von Prof. Sigrid Metz-Göckel vom Hochschuldidaktischen Zentrum der Universität Dortmund für ihre Unterstützung bei der Evaluation und Prof. Ernst-Erich Doberkat für seine hilfreichen Anmerkungen.

Literatur

- [APS04] Alfert, K., Pleumann, J., und Schröder, J.: Software engineering education needs adequate modeling tools. In: Horton, T. B. und Soebel, A. E. (Hrsg.), *17th Conference on Software Engineering Education & Training (CSEE&T 04)*. S. 72–77. IEEE Computer Society. March 2004.
- [DE02] Doberkat, E.-E. und Engels, G.: MuSoft - Multimedia in der Softwaretechnik. *Informatik: Forschung und Entwicklung*. 17(1):41–44. 2002.
- [GJM03] Ghezzi, C., Jazayeri, M., und Mandrioli, D.: *Fundamentals of Software Engineering*. Prentice-Hall. 2. 2003.
- [JBR98] Jacobson, I., Booch, G., und Rumbaugh, J.: *The Unified Software Development Process*. Addison Wesley. 1998.
- [Ke02] Kelter, U.: Bericht des MuSoft-Teilprojekts 3.4 über das erste Projektjahr. In: Doberkat, E.-E. und Engels, G. (Hrsg.), *Ergebnisbericht des Jahres 2001 des Projektes "MuSoft-Multimedia in der SoftwareTechnik"*. Software-Technik Memo No. 121. Universität Dortmund. February 2002.
- [KMGSD04] Kamphans, M., Metz-Göckel, S., Schöttelndreier, A., und Drag, A.: Der Unified Process im Test. Evaluationsergebnisse zum Einsatz des UP in der Informatik-Lehre. MuSoft-Bericht Nr. 7. Lehrstuhl für Software-Technologie, Universität Dortmund. 2004. im Druck.
- [MS01] Mandl-Striegnitz, P.: Qualifizierte Software-Projektmanager durch simulationsbasierte Ausbildung. In: Lichter, H. und Glinz, M. (Hrsg.), *SEUH 7. Software Engineering im Unterricht der Schulen*. S. 23–37. Zürich. February 2001. dpunkt-Verlag.
- [Ri99] Richter, R.: Projektarbeit und Projektmanagement in Informatikprojekten. In: Dreher, B., Schulz, C., und Weber-Wulff, D. (Hrsg.), *Software Engineering im Unterricht der Schulen SEUH'99*. S. 69–78. Wiesbaden. February 1999. Teubner.
- [Sc01] Schmedding, D.: Ein Prozessmodell für das Software-Praktikum. In: Lichter, H. und Glinz, M. (Hrsg.), *SEUH 7. Software Engineering im Unterricht der Schulen*. S. 87–97. Zürich. February 2001. dpunkt-Verlag.
- [vG98] von Glasersfeld, E.: *Radikaler Konstruktivismus. Ideen, Ergebnisse, Probleme*. Suhrkamp. 1998.