

Das Projekt „MuSofT – Multimedia in der SoftwareTechnik“

Klaus Alfert
Lehrstuhl für Software-Technologie
Fachbereich Informatik
Universität Dortmund
alfert@ls10.de

6. Februar 2002

1 Einleitung

Das Projekt „MuSofT – Multimedia in der SoftwareTechnik“ ist am 1.3.2001 gestartet und wird bis Ende 2003 vom BMBF im Rahmen des Programmes „Neue Medien in der Hochschullehre“ gefördert. Acht Universitäten und Fachhochschulen erarbeiten gemeinsam multimediale Lehrmaterialien für das Fachgebiet Software-Technik.

Die Software-Technik ist ein wichtiges Teilgebiet der Informatik und befaßt sich mit der Fragestellung, wie (insbesondere große) Software-Systeme konstruiert werden müssen, um stabil, robust, wartbar und änderungsfreundlich, etc., zu sein [GJM91, Som90, Bal96]. Im Gegensatz zu anderen Gebieten der Informatik, die sich stärker an den Grundlagen von Teilproblemen orientieren, ist die Software-Technik eine Ingenieursdisziplin. Sie stellt mit ihrem Querschnittsangebot einen wesentlichen Teil des Curriculums für Informatiker dar, ist aber auch für „Bindestrichinformatiker“ (z. B. Wirtschafts- oder Medieninformatiker) und Ingenieure (z. B. E-Technik, Lo-

gistik) von zentraler Bedeutung, da auch ihre Aufgaben immer stärker Software-Produktion einschließt.

Die im Rahmen von MuSofT bearbeiteten Themen decken einen großen Bereich sowohl der grundständigen als auch der fortgeschrittenen Lehrgebiete innerhalb der Software-Technik ab. Zu nennen sind:

- Vorgehensweisen in Anforderungsanalyse, Entwurf und Realisierung
 - videogestützte Anforderungsanalyse (Prof. Engels, Paderborn)
 - Entwicklung von Informationssystemen (Prof. Saake, Magdeburg)
 - Software-Architekturen (Prof. Doberkat, Dortmund)
 - Entwurfsmuster (Prof. Seehusen, Lübeck)
 - Algorithmen und Datenstrukturen (Prof. Schürr, München)

- Softwaretechnische Fachdidaktik
 - Dekonstruktion
(Prof. Magenheimer, Paderborn)
 - Software-Engineering in der Lehrerausbildung
(Prof. Magenheimer, Paderborn)
- Prozess- und Projektmanagement in der Software-Entwicklung
 - V-Modell und Qualitätsmanagement
(Prof. Schmidt, Stuttgart)
 - Software-Praktikum mit dem Unified Process
(Prof. Doberkat, Dortmund)
 - Projektmanagement
(Prof. Kelter, Siegen)

Abgerundet werden die Lerneinheiten von übergreifenden Arbeitsgruppen, die sich um die Einhaltung didaktische Standards, um Gleichförmigkeit der verwendeten Beispiele und die Übertragbarkeit der Lernmodule für Nebenfächler und “Bindestrich“-Informatiker kümmern. Die verwendete Notation UML [FS99] und die Programmiersprache Java [GA96] sind für alle Lernmodule gleich und stellen so den Rahmen für die Kombinierbarkeit der Lernmodule auf.

Im folgenden wird der Dortmunder Anteil näher vorgestellt, eine kurze Vorstellung der übrigen Teilprojekte findet sich in [DE02].

2 Software-Architekturen

Software-Architekturen befassen sich mit den Prinzipien und Konstruktionselementen, um Softwaresysteme im Großen zu konzipieren und zu realisieren [SG96]. Zentrale Softwarequalitäten wie Stabilität, Offenheit, Änderungsfreundlichkeit sind unmittelbar von der Software-Architektur berührt. Das Thema der

Software-Architekturen wird im Hauptstudium der Informatik im Rahmen der Veranstaltung „Softwaretechnik“ gelehrt und ist zusätzlich Thema von Spezialvorlesungen und Seminaren.

Ähnlich den Elementen des Programmierens im Kleinen, die im Grundstudium gelehrt werden, ist es notwendig, den konstruktiven Umgang mit Software-Architekturen einzuüben, um Architekturen anwenden zu können. Daher wird neben dem Vermitteln der eher theoretischen Erkenntnisse über Software-Architekturen, großer Wert darauf gelegt, die erworbenen Kenntnisse praktisch anzuwenden. Dafür wird ein grafischer Editor realisiert, der die Modellierung von Software-Architekturen erlaubt. Er ist sowohl für den Übungs- und Praktikumsbetrieb konzipiert, ist aber auch als Visualisierungswerkzeug in der Vorlesung oder für das Selbststudium geeignet. Für letzere sind Animationen und Hypertexte vorgesehen, die in den Editor integriert werden.

3 Software-Praktikum mit dem Unified Process

In Software-Praktika wird die Durchführung von Software-Projekten im Team gelehrt, an der Universität Dortmund geschieht dies für Informatik-Studenten im Grundstudium sowie für Nebenfächler im Hauptstudium. Das Software-Praktikum wendet dabei die Kenntnisse aus der Programmierung im Kleinen und der Datenstrukturen in mehrwöchigen Praktikumsaufgaben an. Es bietet damit einen ersten praktischen Einblick in das Gebiet der Software-Technik. Das verwendete Vorgehensmodell ist zumeist – einer Einführung in die Software-Technik angemessen – schlicht gewählt [Sch01].

Der Unified Process ist ein internationaler Industriestandard, der die Durchführung von Software-Projekten beschreibt, in dem der

Software-Lebenszyklus in einen Zyklus von Phasen strukturiert wird [JBR98]. Die Tätigkeiten der Projektmitarbeiter in den verschiedenen Phasen werden durch die Definition von Rollen beschrieben. Als Besonderheit gegenüber anderen Phasenmodellen wird beim Unified Process die Adaption des Modells auf die Projektspezifika betont.

Auf der Basis des Unified Process soll das Software-Praktikum durchgeführt werden, wobei neu hinzukommt, dass die Gestaltung des Software-Prozesses elementarer Bestandteil des Praktikums wird und so Reflektionen über die Tätigkeiten der Praktikanten eingefordert werden. Die Lerneinheit bietet, neben einem lehrbuchartigen Hypertext über den Unified Process, die Möglichkeit Software-Prozesse grafisch zu modellieren. Darauf aufbauend wird eine Prozessverfolgung integriert. Auf dieser Basis können die Studenten ihr eigenes Vorgehen im Praktikum zu evaluieren und bekommen so ein unmittelbares Feedback ihrer eigenen Arbeitsplanung.

Literatur

- [GA96] GOSLING, JAMES und KEN ARNOLD: *Java: The Language*. Addison-Wesley, 1996.
- [GJM91] GHEZZI, CARLO, MEHDI JAZAYERI und DINO MANDRIOLI: *Fundamentals of Software Engineering*. Prentice Hall, Englewood Cliffs, NJ, 1991.
- [JBR98] JACOBSON, IVAR, GRADY BOOCH und JAMES RUMBAUGH: *The Unified Software Development Process*. Addison Wesley, 1998.
- [Sch01] SCHMEDDING, DORIS: *Ein Prozessmodell für das Software-Praktikum*. In: LICHTER, HORST und MARTIN GLINZ (Herausgeber): *SEUH 7. Software Engineering im Unterricht der Schulen*, Seiten 87–97, Zürich, Februar 2001. dpunkt-Verlag.
- [SG96] SHAW, MARY und DAVID GARLAN: *Software Architecture*. Prentice Hall, Englewood Cliffs, NJ, 1996.
- [Som90] SOMMERVILLE, IAN: *Software-Engineering*. Addison-Wesley, 4th Auflage, 1990.
- [Bal96] BALZERT, HELMUT: *Lehrbuch der Software-Technik: Software Entwicklung*. Lehrbücher der Informatik. Spektrum Akademie Verlag, Heidelberg Berlin Oxford, 1996.
- [DE02] DOBERKAT, ERNST-ERICH und GREGOR ENGELS: *MuSoft – Multimedia in der Software-Technik*. Informaik Forschung und Entwicklung, 17, 2002. Im Druck.
- [FS99] FOWLER, MARTIN und KANDAL SCOTT: *UML distilled: A brief guide to the standard object modeling language*. Addison-Wesley, Reading, MA, 2nd Auflage, 1999.