

Vorwort

Liebe Leserin, lieber Leser,

die methodische Entwicklung von Software, auch Software Engineering oder Software-technik genannt, behandelt nicht nur die Erstellung der Software im engeren Sinne, sondern hat auch Fragen der Qualitätssicherung, des Einsatzes und der Wartung sowie des Managements zum Gegenstand. Ein weiterer wichtiger Aspekt besteht darin, dass große Softwaresysteme von heterogen zusammengesetzten Teams entwickelt werden, so dass interdisziplinäres Wissen und soziale Kompetenz der Beteiligten für den Projekterfolg ebenfalls eine zentrale Rolle spielen.

Diese Themenbereiche in einem Buch abzuhandeln heißt, Gefahr zu laufen, völlig oberflächlich zu bleiben. Die Alternative, sich auf wenige Teilaspekte der Softwareentwicklung zu beschränken, die man dann mit einer angemessenen Tiefe behandelt, birgt andererseits die Gefahr, ein zu einseitiges Bild zu provozieren.

Inhalt des Buches und Vorkenntnisse. Das vorliegende Buch gibt daher im ersten Teil einen Überblick über das Gesamtgebiet Software Engineering und behandelt dann in den weiteren Teilen ausschließlich grundlegende Informatikaspekte der objektorientierten Softwareentwicklung. Im Zentrum steht MOOS – eine moderne, UML- und komponentenbasierte Vorgehensweise zur methodischen Entwicklung objektorientierter Software, die den Weg zu neuen, modellgetriebenen Entwicklungstechniken weist, aber dennoch die fundierten Grundlagen des Software Engineering nicht vernachlässigt.

Aus der Sicht des Software Engineering besteht ein wichtiger Vorteil der Objektorientierung darin, dass im Vergleich zu herkömmlichen Vorgehensweisen eine größere Durchgängigkeit durch alle Tätigkeiten der Softwareentwicklung – von der Anforderungsermittlung bis hin zur Implementierung – erreichbar ist. Dies spiegelt sich auch in der Unified Modeling Language (UML) wider, die sich als De-facto-Standard etabliert hat und deren Konstrukte in Anforderungsermittlung und Entwurf gleichermaßen verwendet werden. Das Buch stellt daher zunächst die wichtigsten UML-Elemente (UML-Version 2.0) für die Modellierung der Struktur, der Funktion und des Verhaltens von Softwaresystemen vor.

Objektorientierte Softwareentwicklung ist mittlerweile ein so weites Feld, dass sich auch ein Buch zu diesem Thema – leider – auf wenige Kernbereiche beschränken muss. Zunächst einmal ist deshalb der Anwendungsbereich auf kommerzielle Softwaresysteme eingeschränkt. Darüber hinaus stehen lediglich die grundlegenden Konzepte im Fokus, die für die *Anforderungsermittlung* und den *Entwurf* relevant sind, wobei die

UML als Modellierungssprache benutzt wird. Dabei wird als zusätzliche konzeptuelle Maßnahme die *Spezifizierung* der *Software* als weitere Aktivität zwischen Anforderungsermittlung und Entwurf eingeschoben. Auch Fragen der *Vorgehensmethodik* bei der Anforderungsermittlung, der Softwarespezifizierung und beim Entwurf werden beantwortet. Insgesamt kommen die technischen Kernaktivitäten mit Ausnahme der Implementierung und des Tests ausführlich zur Sprache.

Vorwissen. Unabdingbare Voraussetzung für das Verständnis der Buchinhalte zum Entwurf sind Grundkenntnisse in einer objektorientierten Programmiersprache, möglichst in Java. Alle anderen Teile können mit grundlegenden Informatikkenntnissen gelesen werden.

Gebrauchsanleitung. Grundsätzlich können die einzelnen Teile des Buches sequenziell, von vorne nach hinten, durchgelesen werden. Teil I kann unabhängig von den anderen Teilen gelesen werden und gibt einen Überblick über das Gesamtgebiet des Software Engineering. Der eher an der UML interessierte Leser wird mit den Teilen II und III beginnen, um sich dann entweder in Teil I das Gesamtgebiet des Software Engineering anzusehen oder aber in den Teilen IV bis VIII den Einsatz der UML in den Aktivitäten der methodischen objektorientierten Softwareentwicklung zu verfolgen. Der mit der UML und dem Software Engineering vertraute Leser kann auch direkt in die Teile IV bis VIII einsteigen und die Teile II und III lediglich bei Fragen zu den verwendeten UML-Elementen konsultieren.

Literatur. Hinweise auf verwendete oder weiterführende Literatur finden sich im Buch als eingestreute Referenzen auf die Literaturliste am Ende des Buches. Der Literaturliste vorangestellt habe ich Anmerkungen zu einigen ausgewählten Werken, mit denen interessierte Leserinnen und Leser ihr Software-Engineering-Wissen erweitern und vertiefen können, insbesondere hinsichtlich der nicht im vorliegenden Buch behandelten Aspekte. Grundsätzlich kann das Buch jedoch ohne zusätzliche Literatur gelesen werden.

Entstehungsgeschichte. Das Buch entstand auf der Grundlage des in den Jahren 1997 bis 2002 an der FernUniversität Hagen von Hans-Werner Six und mir entwickelten Fernstudium-Kurses »Software Engineering I – Objektorientierte Softwareentwicklung« [SixWin02], bei dessen Erstellung u. a. meine damaligen Kollegen Henrik Behrens, Bernd-Uwe Pagel und Stefan Jungmayr mitgewirkt haben (Bernd-Uwe Pagel schrieb die erste Version von Teil II, Stefan Jungmayr wirkte an Kapitel 4 und mit Henrik Behrens zusammen auch an den Kapiteln 30 und 31 mit, Henrik Behrens darüber hinaus auch an Kapitel 42). Für die lebhaften Diskussionen und die gute Zusammenarbeit den besten Dank! Zahlreiche Studierende im Studiengang Informatik der FernUniversität haben in den Sommersemestern der Jahre 1999 bis 2002 durch konstruktive Beiträge, Fragen und Rückmeldungen ihren qualitätssichernden Beitrag geleistet.

Nachdem der Kurs so im Jahre 2002 eine gewisse Reife erlangt hatte (und ich an die FH Köln berufen wurde), beabsichtigten Hans-Werner Six und ich, unsere Ideen einem weiteren Auditorium zugänglich zu machen. Nicht zuletzt unsere antizipierten Erwartungen des erweiterten Leserkreises hinsichtlich aktueller Entwicklungen z. B. auf den Gebieten komponentenbasierter Architekturen und modellgetriebener, generativer Vorgehensweisen führten jedoch zu so grundlegenden Überarbeitungen und Erweiterungen des Materials, dass Hans-Werner Six seine Co-Autorenschaft dann leider aufgeben

musste – und damit auch unsere gemeinsame Vision von Teilen beispielsweise zur Entwicklung webbasierter Anwendungssysteme. Für die langen, interessanten und tief schürfenden Diskussionen in der ersten Phase der Überarbeitung sowie die großzügige Erlaubnis, das Projekt alleine weiterführen zu dürfen, schulde ich ihm tiefen Dank.

Dass das Buch also nun – wenn auch mit bescheidenerer Zielsetzung – trotz dieses herben Rückschlages erscheinen kann, ist nicht zuletzt dem aufmunternden Zuspruch und der tatkräftigen Unterstützung der Lektorin des dpunkt.verlages, Frau Christa Preisendanz, zu verdanken. Besonderen Dank schulde ich auch den Kolleginnen und Kollegen aus Hochschule und Industrie, die das Manuskript einem Review unterzogen haben; namentlich bekannt sind mir Markus Völter, dessen kritische Anmerkungen insbesondere die Einordnung modellgetriebener Vorgehensweisen im Buch verbesserten, sowie Falk Fraikin, der mich neben zahllosen konstruktiven Hinweisen dazu ermutigte, das Buchprojekt überhaupt zu Ende zu führen. Von unschätzbarem Wert während der Vollendung des Buches war auch das direkte Feedback vieler Studierender der Fachhochschule Köln, die in meinen Vorlesungen und Praktika zur Softwaretechnik eher unfreiwillig zu Beta-Testern von Teilen des Stoffes wurden.

Das Copy-Editing besorgte mit professioneller Gründlichkeit Frau Ursula Zimpfer. Ihre Anmerkungen sorgten nicht nur für erhebliche redaktionelle Verbesserungen, sondern steigerten auch an vielen Stellen die inhaltliche Qualität. Alle verbliebenen Fehler, Unvollständigkeiten und Inkonsistenzen sind allein mir zuzuschreiben – um entsprechende Rückmeldungen bitte ich Sie hiermit ganz herzlich: winter@gm.fh-koeln.de, Stichwort: MOOS.

Last but not least ein großes »Dankeschön!« an meine Familie: Meine Frau Petra, die trotz eigener Arbeiten immer ein offenes Ohr für die Sorgen und Nöte eines (ver-)zweifelnden Autors hatte, und meine Kinder Mischa und Jonas, die ihren Papa in der letzten Zeit leider allzu oft nur im Arbeitszimmer, nicht aber bei den Legosteinen oder in der Kletterwand finden konnten.

Mario Winter
Wuppertal, Juni 2005