

# Vorwort

Liebe Leserin, lieber Leser,

Softwareentwicklung braucht Planung. Softwareentwicklung braucht Kommunikation. Und Softwareentwicklung braucht Dokumentation. UML ist seit jeher ein passendes Werkzeug, um diese Aufgaben zu bewältigen.

Die immer komplexer werdende Welt der Software, der Komponenten und der Vernetzung im Allgemeinen hat auch anderen Werkzeugen Auftrieb verschafft. So findet man nun wieder mehr Prozessmodellierungen in BPNM, dem *Business Process Model and Notation*. BPNM spezialisiert sich dabei aber auf den Teilbereich der Prozesse, während UML umfassender so ziemlich alle Aspekte eines Softwareprojekts abdeckt.

Einen weniger formalen Ansatz als UML verfolgt das neuere *C4 Model* zur Architekturbeschreibung. Die 4 steht dabei für vier Zoomlevel wie bei einer digitalen Landkarte – angefangen bei Ansicht des Kontextes über die Verbindung der Container zu der Visualisierung der Komponenten. Aber das sind ja erst drei! Denn als vierte Stufe soll der Code selbst dienen, beziehungsweise die daraus generierte oder ohnehin schon vorhandene Dokumentation. Und dafür werden Klassen- und Komponentendiagramme vorgeschlagen oder Sequenzdiagramme eingebunden – und was bietet sich da besser an als UML? UML und *C4 Model* ergänzen sich also. Und was auf den ersten Blick wie weniger Formalität aussieht, entpuppt sich beim zweiten Hinsehen genau wie bei UML nur als ein persönlicher Dialekt, der sich in einem gegebenen Team immer einstellen wird.

Denn bei all den Spezifikationsansätzen – sei es BPNM, *C4 Model* oder UML – stellt sich immer wieder heraus: Die Präzision der Sprache, die alle Projektbeteiligten sprechen, bestimmt die Qualität der Kommunikation und somit letzten Endes die Qualität und den Erfolg des Softwareentwicklungsprojekts. Es ist die Einigung auf eine gemeinsame Notation, die möglichst frei von Ungenauigkeiten sein muss und wenig Spielraum für Missverständnisse lassen darf, die eine gute Spezifikationssprache ausmachen. Und UML leistet genau das. Die Projektbeteiligten brauchen ein gemeinsames »Glossar«, als Zwischenschicht für die Kommunikation untereinander, damit Begriffe, die in unterschiedlichen Bereichen unterschiedlich benutzt werden, eine klare, eindeutige Belegung erhalten. Wenn der Vertrieb von »Preis« redet, dann wissen alle Vertriebsmitarbeiter, was gemeint ist. Sieht der Datenbankdesigner das Attribut »Preis«, kommen gänzlich andere Assoziationen in seinen Sinn. Und im Controlling werden wiederum völlig andere Fragestellungen mit »Preis« verbunden. Ohne die Einigung auf eine klare Bedeutung des Begriffes ist ein Projekt, das alle drei Abteilungen berührt, dem Risiko des Scheiterns ausgesetzt.

Mit UML können Sie das vermeiden. Durch die eindeutige Notation und die Menge der abgedeckten Anwendungsbereiche können die Fachbereiche sich untereinander über die Bedeutung von Begriffen abstimmen. Mit UML kann der gesamte Prozess der Softwareentwicklung begleitet werden. Vom Erfassen von User-Stories über Sequenz- und Klassendiagramme bis zum Deploymentdiagramm.

Denn UML ist eine große Spezifikation. Mit UML 2.5 hat die *Object Management Group* (OMG) Änderungen vorgenommen, die UML ein wenig vereinfacht und somit besser handhabbar gemacht haben. In diesem Buch musste für die Version 2.5 nur recht wenig angepasst werden, da wir uns ohnehin auf die praktikabelsten Notationen beschränkt haben.

Was dieses Buch ausmacht, ist, dass es die Brücke zu konkreten Implementierungen schlägt, die sich in Beispielen in Java und C# widerspiegeln. Damit decken wir noch einen weiteren Anwendungsfall von UML ab: Mit einigen Werkzeugen lassen sich nämlich Diagramm und Code direkt miteinander verknüpfen. Sie können Ihrer Lieblings-IDE parallel also das Diagramm aktuell halten oder andersherum in der Diagramm-App mittels Doppelklicks direkt zum Quellcode gelangen. Denn UML wurde mit dem Ziel des computergestützten Softwareengineerings (*Computer Aided Software Engeneering*, CASE) erdacht. Sie bekommen in diesem Buch also wenn möglich immer eine praxisorientierte Sichtweise auf UML gezeigt.

Nun wünsche ich Ihnen viel Erfolg beim Einsatz des Buches.

**Torsten T. Will**

## **Vorwort zur ersten Auflage**

Bei meinem ersten Kontakt mit der Unified Modeling Language (UML) in der Softwaretechnik-Vorlesung konnte ich deren Bedeutung und Nutzen noch nicht abschätzen. Ich hatte bis dato nur an kleineren Softwareprojekten gearbeitet und sah die Notwendigkeit einer Modellierungssprache für Software und deren Nutzen auf den ersten Blick nicht.

Zu Beginn meiner Diplomarbeit merkte ich jedoch sehr schnell, dass meine bisherigen Methoden des »Programmierens im Kleinen« das Projekt nicht zu einem erfolgreichen Abschluss führen würden. Am besagten Projekt sollten drei Software-Teams an drei unterschiedlichen Standorten beteiligt sein und unterschiedliche Komponenten derselben Software entwickeln.

Durch die Modellierung mit der UML gelang es, sowohl mit dem Auftraggeber als auch mit allen Entwicklerteams sehr präzise über die zu erstellenden Komponenten zu kommunizieren und die benötigten Schnittstellen festzulegen, sodass die Implementierung sehr zügig und erfolgreich durchgeführt werden konnte.

Seither zähle ich mich zu den bekennenden Fans der UML, propagiere bei jeder Gelegenheit ihren Nutzen und freue mich, mit diesem Buch auch Ihnen die UML als eine universal einsetzbare Modellierungssprache näherbringen zu dürfen.

Dieses Buch behandelt die UML »von Anfang an« und führt Sie bis in ihre »Tiefen« hinab. Bei jeder Gelegenheit werden zu den vorgestellten Notationselementen anschauliche Beispiele und Realisierungsmöglichkeiten in Java und C# geboten, um Ihnen das Verstehen zu erleichtern, damit Sie das gerade Erlernte sofort in der Praxis anwenden können.

Bei der Erstellung dieses umfangreichen Werkes haben mich einige Menschen sehr unterstützt, denen ich an dieser Stelle meinen großen Dank aussprechen möchte.

Zunächst würde ich gerne Galileo Press<sup>1</sup> und speziell meine Lektorin Frau Stevens-Lemoine nennen, die mich vorbildlich bei der Arbeit unterstützt und dieses Buch überhaupt erst ermöglicht haben.

Meine Kollegin, Frau Klaudia Kern, war sozusagen an der »vordersten Front«. Sie hat die allerersten Versionen der Buch-Kapitel über sich ergehen lassen müssen und half mir mit ihren Korrekturen sehr.

Einen unschätzbaren Wert für das Buch hatten die Beiträge und Verbesserungsvorschläge von Prof. Dr. Convent, Prof. Dr. Juen und Prof. Dr. Kaiser. Als Professoren haben sie mich bereits zu meiner Hochschulzeit an der FH Gelsenkirchen, Standort Bocholt, mit ihrer Fachkenntnis und Erfahrung immer wieder in Erstaunen versetzt. Ihre Ideen tragen einen großen Anteil an der Qualität des Buches.

**Christoph Kecher**

---

<sup>1</sup> Alter Verlagsname des Rheinwerk Verlages (1999–2014)