

IT'S NEXT TOP-MODELL?



Computer auf dem Catwalk? Das Cover dieser Ausgabe von OBJEKTSpektrum zeigt, dass auch unsere anscheinend so rationale Szene hochgradig von Mode und Zeitgeist geprägt ist. Deutlich wird das zum Beispiel, wenn man die Entwicklung der Diskussion in der IT-Gemeinde über Modellierung und Modelle im Laufe der vergangenen Jahre und Jahrzehnte betrachtet. Immer neue Modellierungstrends sind über die Zeit entstanden und teilweise auch wieder verschwunden.

Mit den Datenbanken haben vor langer Zeit Datenmodelle Einzug in die IT gehalten – diese waren aus der real existierenden Anwendungsentwicklung der 80er Jahre kaum wegzudenken. Sogar Unternehmensdatenmodelle wurden propagiert und die IT damit im Gesamtunternehmen strategisch geadelt. Daneben entstand eine erste Softwareguru-Szene mit Persönlichkeiten wie **Tom DeMarco** und **Ed Yourdon**, die Datenflussmodelle der Strukturierten Analyse predigten. Zumindest die Modelle der Strukturierten Analyse sind inzwischen IT-Geschichte.

Die 90er Jahre waren von objektorientierten Modellen und ihren Päpsten, wie **Grady Booch**, **Ivar Jacobson** und **James Rumbaugh**, geprägt. Die *Unified Modeling Language (UML)* war das kristallisierte Ergebnis dieser Epoche und wirkt bis heute nachhaltig auf unsere Arbeit. Die Verwendung der UML ist heute offensichtlich geübte Praxis. Umso wertvoller empfinde ich den Artikel von **Gabriele Taentzer** und **Thorsten Arendt** in dieser Ausgabe von OBJEKTSpektrum, in dem die beiden praktische Tipps für die Qualitätssicherung von UML-Modellen geben.

Einen Höhepunkt der objektorientierten Modellierung haben wir seit Anfang dieses Jahrtausends mit den Ansätzen der *Model Driven Architecture (MDA)* erlebt, die die alte Vision wiederbelebt haben, dass man aus Modellen Software generieren kann. Dieser Ansatz für Softwareentwicklung ist heute noch weit verbreitet, auch wenn nicht ganz so lautstark darüber berichtet wird wie über aktuelle Hype-Themen. **Benedikt von Treskow** hat in diesem Heft einen wirklich guten Erfahrungsbericht hierüber geschrieben und **Daniela Schilling** geht sogar noch einen Schritt weiter, wenn sie ihren Generator-Generator mit dem Ziel der Entmystifizierung des modellgetriebenen Designs vorstellt.

Parallel zu diesen Modellen hat die Modellierung von Geschäftsprozessen immer mehr an Fahrt aufgenommen. Einen ersten Schub bekam das Thema dadurch, dass **August-Wilhelm Scheer** vor den erfolgreichen Einsatz von SAP-Standardsoftware die Modellierung von Prozessketten setzte. Es gibt immer noch Erweiterungen von Prozessmodellierungsansätzen wie den von **Holger Breitling** und **Stefan Hofer** in ihrem Artikel zur exemplarischen Geschäftsprozessmodellierung. Im Sinne der Standardisierung spielt aber gewiss BPMN eine immer größere Rolle, wober **Jakob Freund** in seinem Artikel berichtet.

Der aktuelle IT-Zeitgeist, der ja wesentlich durch die agile Softwareentwicklung beeinflusst ist, schiebt Modelle immer häufiger in die Ecke der zu schwergewichtigen Entwicklung. Ist das der Anfang vom Ende der Modellierung in der Anwendungsentwicklung? Kann man einen roten Faden in dieser Modellgeschichte erkennen? Oder sind es wirklich nur sich mehr oder weniger wiederholende Moden?

Fangen wir einmal mit der Frage an, was Modellierung eigentlich soll. Modelle dienen definitionsgemäß der Abstraktion – sei es der

Abstraktion und Formalisierung von fachlichen Anforderungen und damit der besseren Kommunikation zwischen Anwendern bzw. Auftraggebern und Entwicklern oder der Abstraktion im Sinne einer architektonischen Strukturierung oder sogar der Generierung von Anwendungssystemen. Aber für all diese Ziele sind Modelle nicht unumstritten oder ohne Alternative.

Sowohl die objektorientierte Modellierung als auch die Geschäftsprozessmodellierung haben sich das Ziel einer besseren Kommunikation mit dem Auftraggeber auf ihre Fahnen geschrieben. Die objektorientierten Modelle haben diesen Anspruch nicht immer erfüllt. Genau in diesem Punkt wird die UML-Verwendung in agilen Projektkontexten oft als Verschwendung empfunden, wenn die Modelle keinen direkten Nutzen für die Anwender darstellen, sondern nur als Übergabe-Artefakte zwischen zwei Entwicklungsphasen dienen. Hier wird in agilen Projekten die direkte und persönliche Kommunikation zwischen Fachleuten und Entwicklern in kurzen Iterationen bevorzugt.

Zur Kommunikation zwischen Fachbereich und Entwicklung scheinen Prozessmodelle besser geeignet zu sein. Sie können auch einen fachlichen Rahmen um agile Anforderungen bilden, die in User-Stories formuliert werden und damit nicht nur das System aus Anwendersicht dokumentieren, sondern auch wesentliche Orientierungs- und Strukturierungshilfe geben. **Kim Nena Duggen** und **Stefan Toth** stellen in ihrem Artikel sogar einen leichtgewichtigen Prozessmodellierungsansatz vor, der ohne Werkzeugeinsatz mit kurzen Feedback-Schleifen ideal in agile Entwicklungsmodelle passt. Und in manchen werkzeuggetriebenen Ansätzen wird auch hier Software für automatisierte Workflows aus Prozessmodellen generiert.

Zur Strukturierung (und damit auch Dokumentation) komplexer Softwaresysteme hat sich die UML in ihrer Tradition der Daten- und Funktionsabstraktion wirklich bewährt und es gibt meiner Meinung nach keine wirkliche Alternative zu ihr, auch wenn **Martin Fowler** öffentlichkeitswirksam forderte, dass zur Dokumentation eigentlich guter Code ausreichen sollte. Die zwischenzeitlich fast religiöse Diskussion darum hat er selbst entschärft, indem er den Nutzen von Modellen gerade zur Strukturierung komplexer Systeme einräumt.

Ich habe die Hoffnung, dass sich die IT zunehmend von modischen Trends emanzipiert. Es gibt keine „Weltformel“ für alle Probleme in der Anwendungsentwicklung, sondern wir müssen für die jeweiligen Aufgaben adäquate Verfahren, Artefakte und Werkzeuge wählen. Handwerklich saubere Arbeit – in der Spezifikation, der Modellierung und in der Programmierung – ersetzt mehr und mehr die Tool-Gläubigkeit in der Softwareentwicklung. Gutes Handwerk braucht gute Ausbildung und das Nutzen von Erfahrungen. Wir müssen nicht mit Kanonen auf Spatzen schießen, aber gerade in großen und komplexen Systemen können wir auf die Erfahrungen, die wir in den letzten Jahrzehnten mit Architekturen und Modellen gemacht haben, nicht verzichten.

Ihr Thorsten Janning