

Flotte Softwareprogrammierung darf nicht auf Kosten der Wartbarkeit gehen

Entwicklungserfolg hängt nicht allein vom Tool ab

Werkzeuge für Rapid Development und Softwaremodellierung werden oft unter dem Aspekt der schnellen Anwendungsentwicklung vermarktet. Richard Soley, Chef der Object Management Group (OMG), hält aber die Wartbarkeit der Applikationen für wichtiger. Für OMG-Leiter Soley steht fest: Rapid Application Development (RAD), Anwendungsmodellierung und Tools zur Programmierung von Geschäftsregeln wachsen zusammen. „Alle UML-Tools (Unified Modeling Language) unterstützen bereits RAD“, konstatiert Soley. Auch Geschäftsregel-Tools haben RAD-Charakter, wie Stefanie Peitzker, Marketingleiterin des Business-Rules-Spezialisten Innovations aus Immenstaad am Bodensee, erläutert. Durch die beschleunigte Entwicklung von Business-Logiken lasse sich eine Kostenreduzierung zwischen 30 und 50 Prozent erzielen. Diese Zahl gilt für Java; bei Cobol liegt sie laut Peitzker sogar zwischen 60 und 70 Prozent. Der OMG-Chef warnt jedoch davor, Einsparungen bei der Entwicklung überzubewerten: „Natürlich steht die schnelle Anwendungsentwicklung heute stark im Vordergrund. Aber

wichtiger sollte der langfristige Fokus sein, nämlich auf Integration und Wartung.“ Dort liege der weitaus größte Teil der Kosten einer Applikation.

Wartung generiert die meisten Kosten

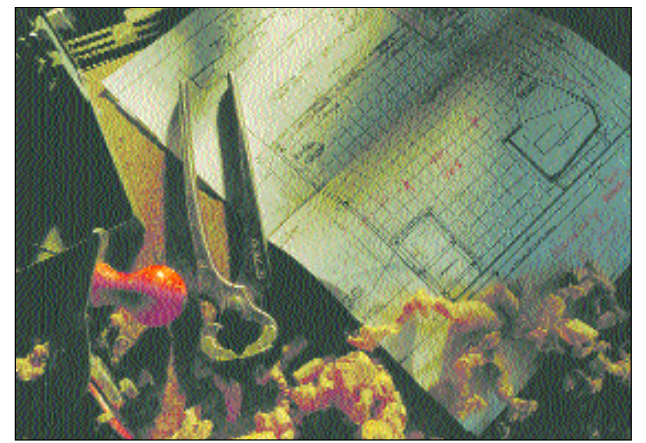
Soley vergleicht die Situation mit dem Bauwesen: „Klar wollen wir ein Haus in der kürzestmöglichen Zeit bauen. Aber wir brauchen einen Bauplan, weil wir erwarten, dass das Gebäude Jahrzehnte hält. Sonst muss man Wände aufbrechen, um zu sehen, wo die Rohre und die elektrischen Leitungen verlaufen.“ Genau dies passiere aber immer wieder bei Software: „Man schaut in den Quelltext und versucht herauszukriegen, was ein Stück Code macht. Wenn wir aber ein Modell hätten, wüssten wir über das Design der Software Bescheid.“

Ein weiterer kritischer Aspekt des RAD-Ansatzes liegt laut Marko Boger, Geschäftsführer des Hamburger UML-Spezialisten Gentleware, in der Systemarchitektur. Bestimmte Entwicklungsschritte, so Boger, ließen sich mithilfe von Assistenten in RAD-Werkzeugen automatisieren. „Das Finden der richtigen Architektur erfor-

dert jedoch auf Seiten der Entwickler viel Erfahrung. Ziel der Entwicklung muss eine skalierbare, robuste Architektur sein.“ Das Augenmerk von RAD-Tools sei jedoch nicht auf die Architektur, sondern auf schnelle Ergebnisse gerichtet. Solche Werkzeuge hätten ihre Berechtigung – etwa wenn es darum gehe, Softwareprototypen für die Kommunikation mit den Kunden zu erstellen. „RAD ist aber eine Sackgasse, wenn eine Anwendung skalieren soll. Dann wird es auf lange Sicht teurer.“

Schnelligkeit ist allerdings das A und O für den Business-Rules-Spezialisten Innovations. An-

wendungen, die mit klassischen RAD-Tools wie Microsofts Visual Studio oder Suns Java Studio Creator erzeugt werden, weisen meist „eine hart verdrahtete Geschäftslogik“ auf, berichtet Marketingleiterin Peitzker. Der Ansatz der Immenstaader beruht jedoch darauf, die Business-Logik getrennt vom übrigen Applikationscode zu halten – gewissermaßen als oberste Anwendungsschicht, die häufig an geänderte Marktgegebenheiten angepasst wird. Jedes Werkzeug hat aber seine Grenzen, und Anwender tun gut daran, nicht alle Behauptungen der Hersteller für bare Münze zu nehmen. Verspre-



Gute Werkzeuge und Pläne können die Entwicklung beschleunigen. Entscheidend sind jedoch kompetente Architekten. Foto: IBM

chungen, nach denen Business-Analysten am Bildschirm Geschäftsprozesse skizzieren und dann per Knopfdruck Software generieren können, bezeichnet OMG-Mann Soley als Hype. „Bei RAD gab es die Annahme, dass wir die Entwicklungszeit auf beinahe Null reduzieren

könnten, wenn wir nur bessere Assistenten hätten. Auch das ist überzogen.“ Ebenso wie die Behauptungen mancher Anbieter von Modellierungswerkzeugen, die eine fehlerfreie Architektur und die automatische Generierung des gesamten Applikationscodes versprechen. fg

Abstraktion für Geschäftsprozesse gesucht

„Wir brauchen keine weitere Sprache“

Laut dem Chef der Object Management Group (OMG), Richard Soley, herrscht bei den Beschreibungssprachen für Businessprozesse geradezu Kakophonie. Er strebt in einem gemeinsamen Ansatz eine verbesserte Interoperabilität an.

Sehen Sie die Unified Modeling Language (UML) auch als Tool zur Abbildung von Geschäftsprozessen?

Die OMG hat derzeit keine Standards zur Geschäftsprozessmodellierung. Hier gibt es bereits viele Sprachen. Die von IDS Scheer gehört zu den verbreitetsten. Es gibt auch Geschäftsregel-Tools von Ilog und anderen – nicht zu vergessen die XML-basierten Ansätze wie BPEL. Wir haben hier eine regelrechte Kakophonie. Was wir brauchen, ist nicht eine weitere Sprache, sondern eine Abstraktion, die Ansätze für Geschäftsregeln beschreibt und es erlaubt, Beschreibungen von der einen auf die andere Sprache abzubilden. Wir arbeiten an einem Tool für dieses Business Process Metamodeling. Aber diese Arbeit steht erst am Anfang.

Also haben Kritiker recht, die sagen, UML sei derzeit zur Geschäftsprozessbeschreibung ungeeignet? Absolut. Wir behaupten nicht, dass UML die einzige brauch-



OMG-Chef Soley hält UML bei der Geschäftsprozessbeschreibung nicht für optimal.

Foto: OMG

bare Modellierungssprache der Welt ist. Aber ich denke, dass wir auf dem richtigen Weg sind. Schauen Sie sich die Firmen an, die Ansätze für die Beschreibung von Geschäftsprozessen definieren – sie sind alle OMG-Mitglieder. Und sie beteiligen sich an dem Projekt, ein gemeinsames Metamodel für Businessprozesse zu finden.

Können Modellierungstechniken den Wandel zu serviceorientierten Architekturen (SOA) unterstützen?

Das ist so, als ob man sagen würde: „Jetzt bauen wir Gebäude aus Stahl anstatt aus Ziegelsteinen. Brauchen wir da noch Baupläne?“ Mit Bauplänen geben wir Informationen über das Design eines Hauses weiter. Modellierungssprachen sind ein Weg, den Entwurf zu kommunizieren, SOA dagegen ein Satz von Vorschriften für das Design – der davon ausgeht, dass wir Systeme als Gruppen von Diensten entwerfen, die untereinander kommunizieren und kooperieren. fg

4/275
EAZ