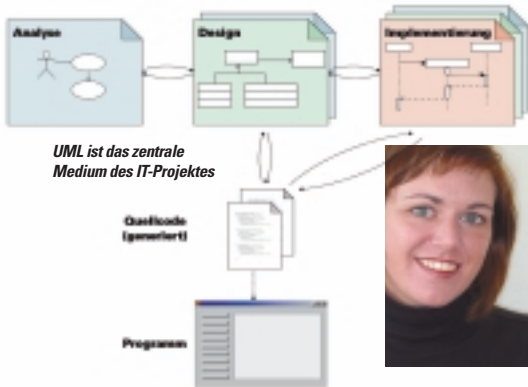


Lingua franca der IT-Welt

Mit UML wurde ein Standard geschaffen, um Systemdesign und -implementierung zu spezifizieren.

Elizabeth Graham und Jesco von Voss



Die Softwaretechnik nähert sich mit zunehmender Reife dem Stand anderer Technikfelder an. Zwei Drittel aller Software-Abteilungen setzen bereits UML ein. Im gesamten Verlauf eines IT-Projektes kann UML die Kommunikation mit Anwendern und im Team verbessern, Anforderungskonformität sichern, Wartbarkeit und Qualität erhöhen und schnellere Projektergebnisse bewirken. Ein UML-Modell enthält Geschäftsprozesse, Programmdesign, Sourcecode und Deployment-Daten in knapper und eindeutiger Form. Die Arbeit mit UML bedeutet das Zeichnen von Diagrammen, die bestimmten Syntaxregeln unterliegen. Elemente eines objektorientierten Systems werden grafisch dargestellt, ihre Beziehungen untereinander eingezeichnet und der logische Ablauf der Interaktion von Komponenten in Grafiken dargelegt. Um das Zeichnen von UML-Diagrammen zu erleichtern, werden in der Regel Software-Werkzeuge eingesetzt, die teilweise kostenlos erhältlich sind. Da UML im Gegensatz zu einzelnen Programmiersprachen nicht von verfügbarer Technik abhängt und mittlerweile schon in den schulischen Informatikunterricht einzieht, ist sie die einzige zukunftssichere Lingua franca von IT-Arbeitern rund um die Welt.



Elizabeth Graham



Jesco von Voss

Über den gesamten Projektzyklus hinweg kann UML eingesetzt werden. Analysten können Geschäftsprozesse grafisch modellieren und Anforderungen dokumentieren. Systemarchitekten modellieren dann das Software-System grafisch; die resultierenden Designdokumente werden genutzt zur Instruktion von Programmierern im Haus oder in Off-shore-Unternehmen, zur Schnittstellendefinition zwischen Teams, zur zukünftigen Wartung, bei der Wiederverwendung von Teilsystemen oder um Anforderungen des Auftraggebers an Systemdokumentation zu erfüllen. Die Programmierer können nun – ohne Medienbruch – das UML-Modell des Systems weiter ausfüllen, indem sie Code für die eingesetzte Technologie (n-Tier, EJB, .NET, SOA, ...) und für die Erfüllung der Geschäftsprozesse ergänzen oder weiter UML verwenden, um diese Aspekte darzustellen und zu simulieren, was im Embedded-Systems-

Umfeld schnellere Entwicklung und Einsparungen bei der Zahl der Hardware-Prototypen bedeutet. Da UML sich auf jede moderne Programmiersprache abbilden lässt, kann aus dem UML-Modell Sourcecode für eine beliebige Plattform generiert werden.

Systemspezifischer Programmcode

Unter dem Stichwort Model Driven Architecture (MDA) wird dieses Vorgehen aus Modellierung und Übersetzung in systemspezifischen Programmcode diskutiert; MDA-Projekte sparen Studien zufolge ein Drittel der Entwicklungszeit ein. Umgekehrt kann – zur Wartung von Legacy-Systemen – existierender, schwer verständlicher Sourcecode in UML übersetzt werden. Ein geeignetes UML-Werkzeug muss Code in verschiedenen Programmiersprachen einlesen und generieren können, Dokumentation des Modells mit allen Diagrammen produzieren, Änderungen des Modells verfolgen und Teamarbeit unterstützen. Außerdem sollte es möglichst unabhängig vom eingesetzten Entwicklungsprozess sein und alle UML-bezogenen Standards erfüllen, um Vendor-Lock-In zu vermeiden. UML ist jedoch nicht die Lösung aller Probleme, denn sie stellt nur eine einheitliche Sicht auf die Artefakte des Entwicklungsprozesses dar. Projekt- und Ressourcenplanung, Risikomanagement, Qualitätssicherung, Installation, Abnahme und Bugtracking liegen außerhalb ihrer Ausdruckskraft. Der Standard UML bietet einen das ganze Projekt umfassenden Formalismus, der universell einsetzbar ist, denn weder Einsatzgebiet noch Technologie sind vorgeschrieben. Kurz- und langfristige Kostenersparnis erzwingen geradezu den Einsatz von UML. ■

Die Autoren Elizabeth Graham und Jesco von Voss sind Mitarbeiter der Firma Gentleware.

Gentleware
www.gentleware.com
nicole.fischer@gentleware.com