

OOP 2008: Mehr Flexibilität im
Geschäftsleben

Komplexität beherrschen

Frank Müller, Kersten Auel

Bereits zum 17. Mal trafen sich Entwickler und Entscheider auf der OOP in München. Zunehmend im Zentrum des Interesses stehen dort neben technischen Fragen solche, die das Management und die Projektführung betreffen.

Für den Erfolg eines Projekts genügt es nicht, kompetente Entwickler zu beschäftigen. Das Beherrschen einer Programmiersprache ist eine Sache, die richtige Wahl der Sprache für ein Projekt eine andere. Davor stehen die Ermittlung der Anforderungen, der Entwurf eines passenden Designs sowie die Entscheidung für ein Vorgehensmodell. Nicht zu vergessen die sogenannten Softskills, die einen Projektleiter dazu befähigen, Mitarbeiter zu motivieren.

Um dieses Themenspektrum abzudecken, boten die Veranstalter der OOP in München den Teilnehmern acht parallele Vortragsreihen, die durch Ausstellervorträge, diverse Keynotes, zwei Tutoriale und diverse Zusatzen-

gebote ergänzt wurden. Mit rund 2000 Konferenz- und Ausstellungsbesuchern konnte die OOP erneut einen Zuwachs verzeichnen – ein Zeichen dafür, dass es auch nach 16 Jahren OOP Bedarf an fachlichem Austausch gibt. Die begleitende Ausstellung konnte im Vergleich zum Vorjahr sogar ein Wachstum von 18 Prozent verzeichnen. Zum ersten Mal seien, so Günter Fuhrmeister, Geschäftsführer von Sigs-Datacom, auch viele Startup-Firmen vertreten gewesen.

Vorträge zu Requirements Engineering, modellgetriebener oder agiler Entwicklung sowie Architekturthemen entsprachen dem Leitthema der diesjährigen Veranstaltung „Foundations for Flexibility“. Seit Jahren versucht man, mit

neuen Paradigmen die Komplexität der Softwareentwicklung in den Griff zu bekommen. Dass das jedoch nicht unbedingt bedeuten muss, beim Design alle denkbaren Fälle abzudecken, betonte Frank Buschmann in seinem Vortrag zum Thema „Patterns and Practices für den Entwurf flexibler Softwaresysteme“. Anstatt eine Anwendung durch die Berücksichtigung aller denkbaren Spezialfälle kompliziert, fehleranfällig und langsam zu machen, sollte man zunächst die Kernfunktionen implementieren und erst von einem stabilen System ausgehend Parameter verändern. Dass Fehler auf diesem Weg dazugehören, wie auch das Lernen aus diesen Fehlern, hob der Architekt in seiner Night School „Aus Fehlern lernen“ hervor.

Einfach Prioritäten setzen

Der Einfachheit dezidiert das Wort sprachen Eric Dörnenburg und Dan North, verwahrten sich aber dagegen, „simpel“ mit „simplifizierend“ gleichzusetzen. Wichtig sei vor allem, sich klarzumachen, für wen man eine Software entwickelt, und Prioritäten zu setzen, das heißt, lieber konsequent eine einzige Sache zu verfolgen als sich zu verzetteln.

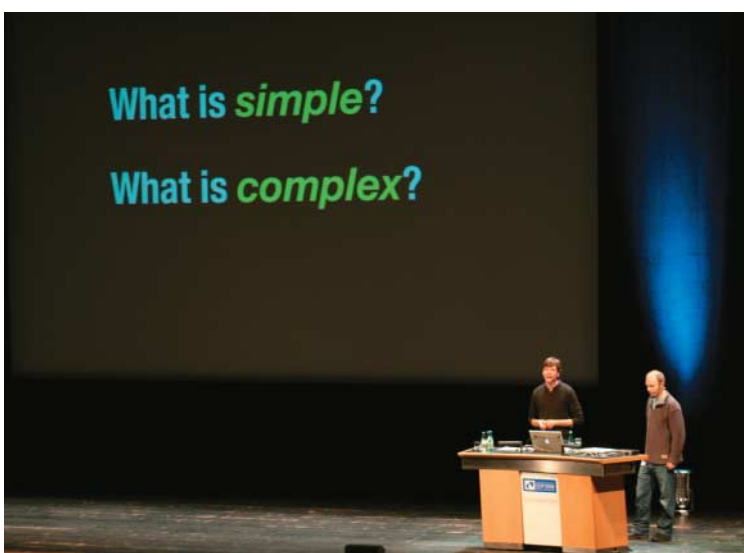
Bereits zehn Jahre sind seit der Einführung agiler Entwicklung vergangen. In seinem „Resümee aus zehn Jahren Praxis“ gelangte Jens Coldewey zusammen mit Vertretern des Publikums zu dem

Schluss, dass testgetriebene Entwicklung auf jeden Fall den Code verbessere. Ein Widerspruch bestehe aber noch bei den Kunden. Diese seien häufig durchaus für agile Entwicklung, bestünden aber auf einem festen Preis und einen festgelegten Rahmen für ein Projekt.

Dass das Webprotokoll REST im Kommen ist, machte Stefan Tilkov nicht nur an dem großen Zuspruch fest, den sein Vortrag fand – in früheren Jahren sei das Thema noch auf deutlich weniger Interesse gestoßen. Er betonte vor allem die Vorzüge von REST für Webanwendungen, da es seine Wurzeln im Internet habe, während Webservices eher aus dem Unternehmensumfeld kommen und Techniken wie CORBA oder EJB nutzen, die ihren Ursprung dort haben. Die Diskussion um die Bedeutung des Protokolls setzte sich im Panel „Was kommt nach SOA?“ mit Nicolai Josuttis und anderen fort. Einer der hier diskutierten Begriffe war auch SOA 2.0. Die Sprecher waren sich über die Unsinnigkeit dieses Begriffs einig, da nicht mal ein klares Bild von SOA 1.0 existiert. Dennoch zeigte sich in den weiteren Sessions, dass SOA in der Praxis angekommen ist. Der Hype ist vorbei, nun gilt es, mit den Alltagsproblemen zurechtzukommen. Vorträge aus der Praxis gaben hierbei Einblicke in die Erfahrungen etablierter SOA-Infrastrukturen.

Smalltalk macht von sich reden

Während immer wieder neue Programmiersprachen im Gespräch sind – dieses Jahr etwa gab es einen eigenen Track zu Ruby und Ruby on Rails –, behaupten andere ihren Platz. Zwar bei Weitem nicht so verbreitet wie Java, C# oder C++, erfreut sich auch dieses Jahr Smalltalk nicht nur einer treuen Fangemeinde, sondern in letzter Zeit auch wachsenden Zuspruchs. Das mag nicht zuletzt der Beliebtheit neuerer dynamischer Sprachen wie Ruby oder Groovy zu verdanken sein. Diese sprechen in ihrer Struktur Entwickler an, die in den traditionellen Sprachen erfahren sind. Die verminderte



Eine einfache Message vermittelten Eric Dörnenburg und Dan North in ihrer Keynote: Um die Komplexität einer Software in den Griff zu bekommen, muss man nur zur rechten Zeit die richtigen Entscheidungen treffen.

Komplexität der dynamischen Sprachen erlaubt eine schnelle und unkomplizierte Realisierung. Und so versprach dann auch die Einladung zum traditionellen Smalltalk-Abend: „Wenn Sie Ruby on Rails mögen, werden Sie Smalltalk lieben!“ Ergebnis war die bisher größte Anzahl an Teilnehmern seit Bestehen der OOP. Ihnen wurden vier Sessions zu Seaside, dem Vergleich dieses Web-Frameworks zu Ruby, dem Einsatz mit GemStone sowie den Neuerungen von VisualWorks 7.6 geboten. Den Abschluss bildete wieder ein lockerer Abend mit viel Raum für interessante Gespräche. Doch nicht nur dieser Abend, sondern die gesamte Ausstellung war laut Aussage von Georg Heeg, der einen Gemeinschaftsstand mit Cincom und GemStone präsentierte, die mit dem bisher größten Zuspruch.

Ob nun der Sieg von Thorsten Seitz beim „Dynamic Language Shootout“ auf den Einsatz des Smalltalk-Web-Frameworks Seaside oder allein seine Programmierfähigkeiten zurückzuführen ist, bleibe dahingestellt. Jedenfalls konnte er sich mit seinem in VisualWorks entwickelten Scrabble-Spiel gegen 11 Konkurrenten durchsetzen, die an dem vom Javaspktrum im Herbst letzten Jahres ausgeschriebenen „Duell der dynamischen Sprachen“ teilgenommen haben (www.sig-datacom.de/sd/kongresse/oop_2008/index.php?cat=dls_competition). Neben dem intelligenten Lösungsansatz konnte besonders die Kür mit einer ansprechenden Oberfläche und der parallelen Auswahl unterschiedlicher Strategien die Jury überzeugen. Diese bestand aus Stefan Tilkov, Michael Stal, Christian Albrecht, Georg Heeg, Michael Weigend, Joachim Baumann und Aristoteles Pagaltzis. Platz 2 belegte Mirko Stocker mit einer unter Ruby und JRuby lauffähigen Anwendung inklusive GUI, Platz 3 Robert Brandner mit einer schnellen und kompakten Lua-Lösung. Sie wies leider kein GUI auf, erzielte dafür jedoch die höchsten Punktestände. Die weiteren Kandidaten wurden in Groovy (2x), Haskell, Scheme, Perl, Python, Scala, erneut Ruby und Squeak realisiert. Es bleibt spannend, wie sehr diese Sprachen ihren

Weg in professionelle Projekte finden werden.

Vor der UML steht die DSL

Eine in der Welt der großen Software-Entwicklungsprojekte bereits angekommene Technik ist die Generierung von Quellen direkt aus dem fachlichen Modell heraus. Bisher hat sich das

Model-Driven Development dabei vielfach der UML als der Lingua Franca für die Modellierung bedient. Im Rahmen der Kommunikation mit dem Fachanwender erweist sich die Unified Modeling Language inzwischen jedoch oftmals als zu komplex. Diesem Problem begegnen die Werkzeughersteller durch die Ergänzung ihrer Programme um grafische, domänenspezifische Sprachen. Deren

Ausprägung richtet sich nach dem jeweiligen Fachgebiet und führt erst später zu Verfeinerungen in der UML oder direkt zur Generierung des Codes. Optionale Simulationen von Prozessen im Modell erleichtern das Nachvollziehen. Ein weiterer Trend, der der Beherrschung der Komplexität dient.

Die nächste OOP findet vom 26. bis 30. Januar 2009 in München statt. (ka)