

► chefredakteur



„Compilerbau ist die am besten durchdrungene Disziplin in der ansonsten doch noch recht unreifen Informatik!“ So oder so ähnlich hat sich mein Doktorvater Prof. Nagl gern ausgedrückt, um seine Freude über diese gereifte Mischung aus Theoretischer Informatik, Programmiersprachenkonzepten und Best Practices in der technischen Umsetzung auszudrücken. Und daran musste ich während der Entstehung dieser Ausgabe von OBJEKTSpektrum immer wieder denken.

Wenn wir hier und heute über moderne Programmierpraktiken sprechen, dann sind wir in genau dieser guten Tradition des Compilerbaus, nur 25 Jahre nach dem oben genannten Zitat meines Lehrers. Heute wird nicht mehr über die Vor- und Nachteile modularer Programmiersprachenkonzepte diskutiert und auch die Welle objektorientierter Sprachen wie C++ und Smalltalk ist über uns hinweg geschwappt. Und wenn auch in der heutigen kommerziellen Anwendungswelt eher über Ruby, Java und ObjectiveC diskutiert wird, wie in dem unterhaltsamen Artikel von **Holger Bohlmann**, **Manuel Küblböck** und **Henning Spille**, so hat doch jede dieser vergangenen Wellen deutliche Spuren bei den modernen Praktiken hinterlassen. **Erich Gamma** war beispielsweise mit seinen Design Patterns auch der Pate für deren kleine Brüder, die Idiome, die **André Janus** und **Jens Jäger** ebenso beschreiben wie die schmutzigen kleinen Brüder der Anti-Pattern, die Code Smells.

Der wesentliche Sinn vieler modernen Praktiken liegt wie der vieler Konzepte aus der Vergangenheit in der losen Kopplung von Komponenten und der sauberer Kapselungen. Aber es geht eben auch um mehr als eine logische Strukturierung: Es geht um immer mehr Flexibilität in der Konstruktion von Software, selbst bei sehr komplexen Anwendungsstrukturen. Und Flexibilität bedeutet eben auch, dass man sich von einer zentralen Steuerung endgültig verabschiedet und Kontrolle zunehmend dezentralisiert, wie **Eberhard Wolff** in seiner Bilanz über die Dependency Injection und aspektorientierte Programmierung herleitet.

Und damit bilden die Praktiken ein praktisches Hilfsmittel für eine konkrete Umsetzung flexibler Services in einer sauberen Architektur im Sinne der SOA-Evangelisten früherer Jahre. Auch **Veit Köppen** und **Liane Will** beschreiben übrigens in ihrem Artikel über die praktische Umsetzung beim professionellen ITIL-basierten Betrieb von SOA-Anwendungen eine besondere Herausforderung beim Einsatz von SOA.

Komme ich aber zurück auf die „neue Flexibilität“ unserer Programmierpraktiken. Sie bildet auch die Basis moderner flexibler Management-Disziplinen. **Marko Schulz** und **Andreas Havenstein** betonen daher in ihrem Artikel über den Certified Scrum Developer: „Scrum erfordert flexible Strukturen, damit das Team jederzeit auf geänderte Anforderungen reagieren kann, die der Product Owner (PO) einbringt. Ein CSD muss Entwurfsprinzipien kennen und anwenden können, die zu entkoppelten Strukturen führen.“ Aber auch der CSD ist mit dem Drang nach Flexibilität immer in der Gefahr, eine saubere Architektur zu vernachlässigen.

Damit haben wir auch schon den Bogen gespannt von den Programmierpraktiken zu den Programmierern und den anderen Menschen im Entwicklungsprozess, die allen meinen Gesprächspartner im OOP-Special dieses Heftes so sehr am Herzen liegen. **Matthias Bohlen** zieht die Parallelen zwischen dem „Flow im Lean“ und dem „Flow im Team“, **Johanna Rothman** gibt Tipps, um die richtigen Leute einzustellen und wie man mit Hilfe dieser Menschen seine Arbeit erledigt bekommt. **Diana Larsen** umschreibt die Softwareentwicklung als einen komplexen Tanz von dem Ganzen (dem organisatorischen System) und seinen Teilen (den betroffenen Menschen).

Ich finde es wirklich auffällig, dass so viele Technik-Freaks irgendwann den Weg hin zur Arbeitsorganisation finden, die einen über Kanban, die anderen über systemische Betrachtungen von Teams. Oft hatte ich ja den Verdacht, dass der Grund darin zu suchen ist, dass man mit zunehmendem Alter nicht mehr der doch sehr schnellen technischen Entwicklung standhält (Autoren und Leser des OBJEKTSpektrums sind hier natürlich ausgenommen. ☺) Aber in diesem Heft ist mir der Zusammenhang zwischen diesen scheinbar weichen Themen und den eher technischen Themen wieder so richtig bewusst geworden: Wir bauen immer komplexere Systeme und merken, dass eine zentrale Kontrolle dieser Komplexität nicht mehr Herr wird. Sowohl die Programmierpraktiken als auch die Managementpraktiken versuchen, Ordnung im Chaos zu halten, indem sie Kontrolle dezentralisieren. Oder wie **Diana Larsen** sagt: „Es ist eigentlich paradox, dass wir erst dadurch mehr Kontrolle über komplexe Ergebnisse bekommen, dass wir auf Kontrolle über die Teile verzichten.“

Ihr Thorsten Janning

Meet the Editor @ OOP 2012
am SIGS DATACOM-Stand

Di, 24.01.2012 15.45–16.15 Uhr

Mi, 25.01.2012 10.30–11.00 Uhr

OOP 2012
SOFTWARE MEETS BUSINESS