

2003年10月 第3期

荟 萃 全 球 软 件 智 慧

**[Dr. Dobb's
JOURNAL CHINA]**
月刊 每月10日出版
定价 人民币18元/本

软件开发

项目管理 **CMM** 实战

Hyslop & Sutter 对话

Robert Martin 编程艺术

李维 设计模式的应用

【特别策划】

UML 2.0

软件模式 专题

in association with

**Dr. Dobb's
JOURNAL**

C/C++
Users Journal

software
development

windows
developer

独家授权中文版

Dr.Dobb's Journal

2003年10月

【专题: C/C++ Programming】

C/C++ 编译器比较

by Matthew Wilson

9大流行 C++ 编译器性能、功能和工具大比拼

C++ 字符串的性能

by Lev Kochubeevsky

Mtlib: 内存跟踪库

by Marco Tabini

Mtlib 是一组自动化内存泄漏寻找和跟踪过程的工具。

Java 2D 与网页

by Paul Tremblett

用于应用程序安全的 AutoLogout

by Jonathan Lurie

一种应用级安全技术, 用 C# 实现, 用于 .NET 平台。

流行程序语言量化分析

by Thomas Plum

通过招聘广告中各种语言的出现次数, 分析其流程度。

嵌入式系统

Spy: Windows CE API 截获器

by Dmitri Leman

Spy 是一种调试 Windows CE 程序的工具。

Michael Swaine 专栏

Ed Nisley 嵌入空间专栏

程序员书评

Verity Stob 的新冒险

新闻与评论

新产品

Windows Developer Journal

2003年10月

【专题: Internet】

URL 规范性测试

by Michael J. Hunter

正确处理 URL 的编码方式, 避免安全漏洞。

用 HttpHandlers 扩展 ASP.NET 程序

by Randy Holloway

.NET 许可模型

by Myk Willis

【专栏】

技术点滴

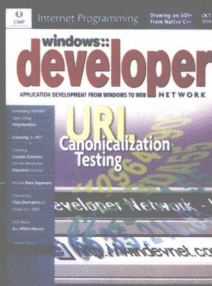
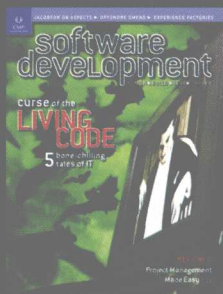
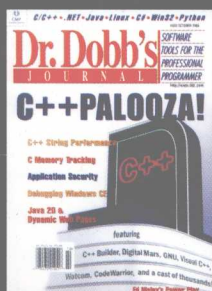
当共享数据段无法共享时
阻止 Visual C++ .NET 中的类派生
do/while 宏问题讨论

深入 .NET

为 WinForms DataGridView 控件创建自定义列
by Dino Esposito

Visual C++ .NET 专家

在原生 C++ 中使用 GDI+
by Richard Grimes



Software Development

2003年10月

【主题: 更好的 beta 版】

活生生的恐怖开发故事

Scott Meyers 等 5 位业界人士回忆了自己在软件开发实践中最恐怖的经历

【质量中心: 失败乃成功之母】

怎样避免犯同样的错误呢?

《New Architecture》杂志的创始主编 Amit Asaravala 提出了建立经验工厂的概念

【产品评论】

Compuware 公司的建模工具 OptimalJ

新版本的 SOAPscope

Borland's JBuilder 9

项目管理工具: Select Process Director Plus

【专栏】

设计中心: AOP 的考察

人们花了 25 年才超出代码一级开始真正实现模型的横切可见性。AOP 能最终不负众望吗?

by Ivar Jacobson

项目管理: 未来的补丁

如何在规划项目的时候考虑未来肯定要出现的 bug?

by Johanna Rothman

敏捷前沿: 杞人忧天?

IT 业的不景气, 面对海外廉价人力资源的竞争, 北美程序员将何去何从?

by Scott W. Ambler

编程艺术之 16

过分的礼节

by Robert C. Martin

C/C++ Users Journal

2003年10月

【专题: 开发过程】

精益软件开发

by Mary Poppendieck

有一种激情有助于项目的成功

以 C++ 兼容的方式启动线程

by Wolfgang Bangerth

BuildConfig: 运行时获得编译配置信息

by Drew Hall

在软件中嵌入配置信息

同一性与相等性: 语法与语义

by Matthew Wilson

两个术语的故事

【.NET 空间】

.NET 到 Java 虚拟机的桥梁

by Jean-Yves Mengant

通过一点 C++ 的帮助, 实现 Java 与 .NET 的互操作性

托管 C++ 中的多线程

by Rex Jaeschke

【专栏】

思维模式

by Miro Samek

名字, 隐喻, 更好的程序设计和语言的政治学

两种性能

by Andrew Koenig & Barbara E. Moo

无论硬件多么高端, 算法还是要考虑的

对话: 头文件的独立性

by Jim Hyslop & Herb Sutter

Conversations - Self-Sufficient Headers

编码标准规定, 每个头文件都应该独立, 事实总是如此吗?



Dr. Dobb's 软件研发 征订单

订阅类别

☐ 单位订阅

单位名称: _____
 联系人: _____
 详细地址: _____
 邮 编: _____
 联系电话: _____
 电子邮件: _____

☐ 个人订阅

姓 名: _____
 工作单位: _____
 职 务: ☐ 开发人员 ☐ 测试人员
☐ 架构师 ☐ 系统分析员
☐ 项目经理 ☐ CTO/CIO
☐ 研究人员 ☐ 系统/网络管理员
☐ 其他 _____

详细地址: _____
 邮 编: _____
 联系电话: _____
 电子邮件: _____

为了便于我们为您提供各种服务, 请准确填写, 字迹清晰。

订阅信息

■ 订阅方式

1. 网站订阅: 可以通过<http://www.china-pub.com>在线订阅。
2. 邮局汇款: 请务必在汇款单附言中注明所需期数。

汇款地址: 北京市西城区百万庄南街1号

邮 编: 100037

收 款 人: 华章公司

电 话: 010-88379622

010-68995259

读者服务部: hzedu@hzbook.com

3. 银行汇款: 请务必将订阅回执单和汇款凭证传真到本刊读者服务部。

汇款账号: 500120105174193

开 户 行: 北京市商业银行甘家口支行

开户单位: 北京华章图文信息有限公司

传 真: 010-68995260

■ 订阅数量

订阅份数: _____

订阅期数: 2003年____月至2003年____月

■ 汇款方式

☐ 邮局汇款 ☐ 银行汇款

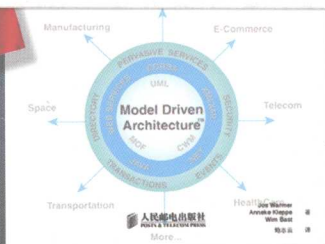
■ 订阅参考

2003年全年8~12月共5期, 每期18.00元, 订阅价90.00元, 免邮费。

回执

书号: 7-115-11559

定价: 29 元



书号: 7-115-11812

预计出版时间: 2003 年 12 月

书号: 7-115-10887

定价: 36 元

回顾与展望

不知不觉地，杂志从创刊至今已经出版到第3期了。十一到了，周围的朋友同事，都纷纷外出云游。而我，除了在家里补一补觉，还得为11月、12月和明年1月的稿子操心，这个国庆看来又是轻松不了了。虽然的确辛苦，但是看着前三期的封面、内容好像一步一个台阶，杂志成长的轨迹清清楚楚，即使明知离真正合格尚有距离，心中还是禁不住充满了敝帚自珍的欣喜之情。

对已经上市的第1期和第2期，不同背景的读者在反馈上差异很大，而且态度都挺旗帜鲜明。问题其实归结起来，还是开发技术和工程管理实践两大方面如何平衡。我们从本期开始采用了“双主题制”，一大一小，大的仍然称为专题，而小的会以“特别策划”的面目出现。这样，两大方面每期均各有一组文章，内容会丰富一些，希望能更具平衡感。如果没有特殊情况，这种安排将在未来几期延续，效果如何，请读者评判。

本期的专题是“软件模式”。当我最初把念头告诉几个朋友时，几乎没有人赞成的，为什么？难度太大了。的确，如今的软件模式，早已超出了GoF的范围，涵盖软件开发各个层次（架构模式、设计模式、惯用法）、各个方面（数据库、界面、反模式、错误模式）、整个生命周期（业务建模、需求分析、用例、测试、维护、再工程、文档……）乃至相关的组织、业务、教学等等环境，而且还在向各个方面渗透。该如何选材呢？作为本刊首次介绍，我们的侧重点还是放在了扩大覆盖面上，希望能够给大家一个比较全面的认识。Alexander不为人知的悲壮经历，相信许多朋友都会感兴趣。《通过重构实现模式》一文对许多唯模式论者恐怕都是很好的清醒剂，对于正在学习的读者也有相当的借鉴意义。由于篇幅有限，通用层面的模式介绍最后只能选择目前讨论比较少但是又非常重要的几种模式：业务模式和分析模式、针对调试的错误模式。此外我们还有文章分别介绍Delphi、C++和J2EE中模式的应用。好在我们有长期的模式专栏，所以感觉不过瘾的朋友，也不必过于遗憾。对了，我们计划在近期采访GoF和《面向模式的体系结构》(POSA)系列的作者Party of Five，正在征集问题，大家集思广益一下。

有一个消息要告诉大家，经过我们网站工作人员的奋战，杂志网站 www.ddjchina.com 就要以全新面貌正式开张了。新网站什么样？卖个关子，请大家密切关注。

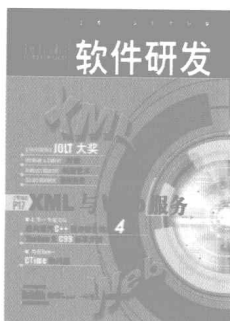
最近，《Dr. Dobb's Journal》的主编 Jon Erickson 向我透露了2004年原版杂志的出版计划。总的看与历年主题大同小异，充分显示了杂志不为流俗所动的风格。从这个单子也可以看出本刊来年的雏形（真是值得期待）。在这里先给大家透个底：本刊下期主题将是“测试与调试”，特别策划是“AOP”。12月主题是“网络与安全”。

editor@ddjchina.com

2003年10月

《Dr. Dobb's Journal》英文版2004年出版计划

1月 最佳实践	5月 优化与性能调整	9月 分布式计算
2月 Web 服务	6月 通信——无线及其他	10月 智能系统
3月 语言	7月 Java	11月 计算机安全
4月 算法	8月 测试与调试	12月 数据库开发



专题 软件模式

12 人物 Christopher Alexander 模式思想先驱

27 通过重构实现模式 Joshua Kerievsky

模式和重构等编程实践都应该根据需求不断变化。

32 分析模式初论 Martin Fowler

分析模式反映的是业务过程的概念架构，建模人员可以从中找到经过检验的解决问题的基本思路。

38 业务模式实例 Hans-Erik Eriksson & Magnus Penker

业务模式捕捉和描述的是业务建模问题及相应的解决方案，以求能够重用。

46 企业级应用架构中模式的应用 Martin Fowler

分层是常用的架构模式之一，Martin Fowler 在介绍模式的同时，还讨论了大量企业级应用开发的心得。

55 C++ Guard 惯用法 Eric D. Crahen

Guard 惯用法提供了比 Scoped Locking 更大的灵活性。

95 从软件质量检验到设计模式的应用 李维

本文中，李维先生以活生生的例子说明了 Dispatcher 模式和 Factory 模式的实际应用。

111 错误模式 Eric E. Allen

错误模式是一个非常有用的概念，它将提高你检测和修正代码中错误的能力。

特别策划：UML 2.0

14 大师圆桌会：定义 UML 内核

17 UML 2.0 与实践者对策 尤克滨

24 UML 2.0 与模型驱动开发实战 *Lars Mats*

专栏

1 编辑寄语

5 软件近事

7 编读往来

8 软件春秋

敏捷革命（下）

软件管理

59 CTO 日记

75% 就够了 *Steve Adolph*

60 项目管理论坛

精益开发方法（下） *Mary Poppendieck*

CMM 实战 *Neil Potter & Mary Sakry*

软件技术

69 C/C++ 专家论坛

Herb Sutter 专栏：迈向 C++0x 标准库【第 2 部分】 *Herb Sutter*

The New C：一切都源于 FORTRAN *Randy Meyers*

80 对话

顺序问题 *Hyslop & Sutter*

84 C++ 技巧

也谈在 STL 中创建索引表 *Sol Foster*

85 每月 Bug++

多重继承 *Jeff Claar*

88 深入 .NET

在浏览器中驻留 .NET 应用程序

Dino Esposito

91 Java 核心技术

EJB 模式 EJB 的设计策略、惯用法和技巧

Floyd Marinescu

101 Borland 技术专栏

Delphi 中 DDL 对象封装和调用技术 刘艺

106 嵌入式系统

Windows CE.NET 测评 *Martin Timmerman*

108 编程艺术

快速减肥 *Robert Martin*

114 技术点滴

使用快捷键自定义键盘导航，第 2 部分

Matthew Wilson

一个灵活的用于窗口居中的函数

Matthew Wilson

在 MFC 扩展 DLL 中使用资源

Thomas Gawehns

Feature: Software Patterns

- 12 People The Pioneer: Christopher Alexander
27 Stop Over-Engineering! **Joshua Kerievsky**
32 Analysis Patterns **Martin Fowler**
38 Business Patterns & UML Applied **Hans-Erik Eriksson & Magnus Penker**
46 Applying Patterns in Enterprise Application Architecture **Martin Fowler**
55 The Guard Idiom **Eric D. Crahen**
95 From Software Metrics to Design Patterns Applied in Delphi **Li Wei**
95 Bug Patterns: An Introduction **Eric E. Allen**

Special: UML 2.0

- 14 Gurus' Roundtable: Define the UML Kernel
Cris Kobryn, Scott Ambler, Peter Coad, Martin Fowler
17 UML 2.0 & the Practitioner's Strategies **You Kebin**
24 UML 2.0 & Model-Driven Development **Lars Mats**

Column

- 1 Editorial
5 News & Views
7 Letters
8 Software History: Agile Revolution Part 2

Management

- 59 CTO Diary
The 75 Percent Solution **Steve Adolph**
60 Project Management Forum
Lean Programming, Part 2 **Mary Poppendieck**
Practical CMM **Neil Potter & Mary Sakry**

Technology

- 69 C/C++ Experts Forum
Sutter's Mill: Toward a Standard C++0x Library, Part 2 **Herb Sutter**
The New C **Randy Meyers**
80 Conversations
Points of Order **Hyslop & Sutter**
84 C++ Tips
Creating an Index Table in STL, Take 2 **Sol Foster**
85 Bug++ of the Month
Multiple Inheritance **Jeff Claar**
88 Inside .NET
Hosting .NET Applications in the Browser **Dino Esposito**
91 Core Java
EJB Design Strategies, Idioms, Techniques **Floyd Marinescu**
101 Borland Corner
Encapsulating and Invoking DDL Objects in Delphi **Liu Yi**
106 Embedded Space
Examining Windows CE.NET **Martin Timmerman**
108 Craftsman
Crash Diet **Robert Martin**
114 Tech Tips

执行主编: 刘江
责任编辑: 贾梅
编辑: 吴怡 陈剑瓯
编辑信箱: editor@ddjchina.com
美术编辑: 陈子平 锡彬
排版制作: 金浩
网站: 周明涛

编委会

中文版: 李 维 潘爱民 钱 岭 孙以义 夏 昕
曹晓钢 左轻侯 李会武 武卫东 温莉芳

英文版:**Dr.Dobb's Journal**

J.Erickson, A.Stevens, B.Schneider, R.Duncan, J.Woehr, J.
Bentley, T.Kientzle, G.Wilson, M.Nelson, E.Nisley, M.
Swaine

Software Development

A.W.Morales, S.Ambler, D.Cline, W.Keuffel, A.Barnhart,
G.Evans, L.O'Brien, R.Wayne, B.Douglass, M.Fowler, E.
Gottesdiener, R.Martin, B.Meyer, S.Meyers,
M.Poppendieck, G.Pour, J.Rothman, G.van Rossum

C/C++ Users Journal

J.Casad, C.Allison, D.Abrahams, B.Karlsson, D.Saks,
H.Sutter, M.Austern, T.Becker, S.Dewhurst, A.Koenig, R.
Meryers, B.Moo, B.Schmidt, R.Ward

Windows Developer Magazine

J.Dorsey, J.Claar, D.Esposito, G.Frazier, R.Grimes,
P.Hesselberg, P.Tomlinson, V.Volkman

读者服务

电子邮件: reader@ddjchina.com
电 话: 68995259
网 站: www.ddjchina.com

版权登记号 图字: 01-2003-1691

图书在版编目(CIP)数据

Dr. Dobb's 软件开发. 第3期 / (美) 萨特 (Sutter, H.) 等著; 刘基诚等译. —北京: 机械工业出版社, 2003.10

ISBN 7-111-13158-4

I. D… II. ①萨… ②刘… III. 软件开发 IV. TP311.52
中国版本图书馆CIP数据核字(2003)第089878号

出版发行

机械工业出版社
(北京西城区百万庄大街22号 邮政编码 100037)
印刷: 中国电影出版社印刷厂
版次: 2003年9月第1版第1次印刷
889mm × 1194mm 1/16 · 7.5印张 16彩页
定价: 18.00元

版权声明

Dr.Dobb's Journal China contains articles under licenses from CMP Media LLC. The articles appearing on pages 14,24,27,55,59,60,69,80,84,85,88,106,108,114 are translated and reprinted by permission of Dr.Dobb's Journal, C/C++ Users Journal, Windows Developer Magazine, and Software Development copyright©2003 CMP Media LLC. All rights reserved.

本书部分文章翻译自 Dr.Dobb's Journal, C/C++ Users Journal, Windows Developer Magazine 和 Software Development. 由 CMP Media LLC 独家授权。未经出版者书面许可, 不得以任何方式复制或转载部分或者全部内容。版权所有, 侵权必究。

前沿：新一代的搜索技术

9月24日，代号为WebFountain的新一代智能搜索技术亮相。这是IBM Almaden研究中心三年多的成果。新搜索平台融入了大量新技术，包括数据挖掘、模式识别、自然语言处理、贝叶斯概率论等等。它可以在网上搜索诸如“人们对《软件研发》杂志评价如何，为什么”之类问题的答案。目前该技术还处于实验室阶段，但是即将用在IBM的一些大客户项目中。相信过不了多久，Google就会感觉到它的压力了。

人事：Bill Joy 离开 Sun 公司



事情发生在9月9日。这个消息折合到里氏单位，应该有六七级以上。也许从Sun目前的公司业绩来说，这不会有太大影响，然而，对于开发人员来说，Sun公司真正的精神领袖是Bill Joy。Bill Joy何许人也？他曾被《财富》杂志称为“Internet的爱迪生”，是美国工程院院士，美国艺术与科学学院院士，1982年创办Sun的四个人之一。Bill Joy离职前是Sun公司的首席科学家。要说他的成就，可以列出惊人的一大堆。使他成名的当然是BSD即Unix的Berkeley版本。他还是Unix平台上人见人爱的vi以及csh的设计者。在Sun公司，他设计了NFS协议，领导了Java、Sparc处理器、Solaris操作系统的开发。近年来他的主要精力放在Jini和JXTA技术的开发上。目前，尚不清楚他下一步的动向。

语言：Perl 6 与 Parrot

Perl语言近来似乎有些冷清，风头已经被Python、Ruby等后起之秀抢去了不少。这一点从相关图书的出版势头就能够看出来。关注开源软件的读者也许还记得，2001年的愚人节时O'Reilly公司的网站上曾经发布了一条消息，说Perl之父Larry Wall和Python之父Guido van Rossum达成了一致，决定将两种语言合并，推出一种新的语言Parrot，并宣布O'Reilly公司将出版一本由Larry和Guido合著的《Programming Parrot in a Nutshell》。当时网页上封面中可爱的木

版画鹦鹉还真的让我信以为真了呢（其实书名本身就透露出玩笑气息了：Programming in a Nutshell本就是O'Reilly公司的两个系列）。可是更令人吃惊的是，本月Parrot真的横空出世了（0.0.11版）。原来，在Perl从5向6的升级过程中，Larry Wall和其他核心开发人员决定对Perl进行大手术，开发一个全新的虚拟机（也就是解释器），基于寄存器（这一决策相当大胆，因为目前主要语言的虚拟机都是基于栈的，Miguel de Icaza对此就大不以为然），能够支持多种语言。这就是Parrot。

还有一件事儿很有意思，Perl 6的开发还八字没有一撇呢，可是O'Reilly公司早已出版了一本《Perl 6 Essentials》，也算史无前例。副标题说明了真正的意图：The Community Rewrite of Perl。



奖项：JavaWorld 编辑选择奖

本刊上一期介绍了本年度Jolt大奖的获奖图书和产品，许多读者表示了很大兴趣。确实，从这些权威评奖中可以看到国外的趋势，有很大的借鉴意义。今年的JavaWorld编辑选择奖分为10类，分别是：最佳Java数据访问工具Oracle 9iAS TopLink（最后入围的还有CocoBase Enterprise O/R 4.5和Hibernate 1.2.4），最佳Java IDE IntelliJ IDEA 3.0，最佳Java性能监视和测试工具JUnit 3.8.1，最佳Java应用服务器BEA WebLogic Server 8.1，最佳Java设备开发工具J2ME WTK，最佳Java/XML工具Xerces2 Java Parser 2.4，最佳Java安装工具Java Web Start 1.2，最佳Java图书《Patterns of Enterprise Application Architecture》（最后入围的还有《Java Development with Ant》、《Java Performance Tuning, 2/e》），最有价值的由Java社区开

发的技术Apache Ant 1.5，最具创新Java产品和技术AspectJ 1.0.6（最后入围的还有Eclipse 2.1，JavaServer Faces）。可以看到整个获奖情况与Jolt奖暨生产效率奖有很多重合，这说明专家们的看法是一致的。

观察：XML 使计算机

回归冯·诺伊曼架构

本刊上期《基于XML的编程系统》一文曾展望了XML在未来程序设计系统中的中心角色。无独有偶，最近一家英国的小公司hyfinity申报了一项专利，提出了一种基于XML的全新计算机架构。我们都知道，因为数据和程序都是二进制的位，所以能够共享内存，这是冯·诺伊曼架构的基本原理之一，这种经典架构从计算机诞生以来一直延续至今，占据主流地位。但是计算机的发展正在逐渐偏离这种架构，比如面向对象技术就是一方面将数据和操作绑定在一起，一方面存储两者的方式却又完全不同。XML的兴起使人们开始想到，这可能使计算机回归到最初的形式，因为数据和程序现在都可以用XML保存了，XML甚至还能用同一种形式存储输入和显示。比如可以用ebXML表示数据，BPEL4WS作为程序，而XForms作为显示。我们可以进一步开发出XML机器，用XML代替现在形形色色的各种计算机结构、程序、数据，这样机器只需要能高效地处理XML就行了。而程序开发的工作量就会小得多。hyfinity公司的专利正是由此出发的，他们用Java开发了一种称为Morphyc的内核，相当于冯·诺伊曼架构中的控制进程，它将输入/输出，内存和算术/逻辑单元组合起来。这其中算术/逻辑单元都可以是普通进程，比如XML解析器、XPath和XSLT处理程序等。这种架构已经显示了威力：由于内核极小，非常简洁，可重用性极高，这家只有8个全职雇员的公司已经基于此内核开发了可以用于快速Web程序设计的MVC（是XForms的扩展）；开发了P2P集成软件包PxP；以及用于开发这些软件的开发环境xStudio。而且

已经有一些客户在使用这些软件。

趋势：AOP

众所周知，语言和开发工具的选择最终往往会影响操作系统和服务器软件的销售。VB的流行，对于微软的成功就居功至伟。而Java对于非微软阵营的意义也是不言而喻。针对C#的凶猛攻势，Sun公司的应对措施是Rave（意为咆哮，Sun最近的项目代号都怒气冲冲的，比如Mad Hatter，国内媒体译为“狂仇”）计划，直接针对VB，大有围魏救赵的意思。而IBM则和最近加入了JCP（Java Community Process）的JBoss公司开始展开合作，探讨将AOP（Aspect-Oriented Programming，面向方面程序设计）技术在Java语言中标准化，以图在技术上领先微软。

AOP是JBoss 4.0的重要卖点，而IBM则延续了AOP技术发源地施乐PARC AspectJ的香火，公司本身还有一个计划叫做HyperJ。IBM和JBoss两者的合作当然是不可小觑。但是微软也不是吃素的。微软从MTS开始，到COM+再到.NET企业级服务，历代技术中其实都已经大量采用了AOP技术。像平台服务中安全、事务等方面的处理就是采纳了“关注点横切（concern crosscut）”的概念，只不过微软没有拿这个说事儿罢了。何况，出身于PARC的微软前任首席架构师、杰出工程师Charles Simonyi可是AOP的先驱之一，虽然他在微软研究院的Intentional Programming项目没有得到最终的认可，但是他离开微软创办Intentional软件公司可是得到盖茨大力支持的。这家公司主攻的其实就是AOP，而且AOP之父Gregor Kiczales是公司的合伙人。Intentional公司与微软签有优先权合同，如果研发有实质进展，摘桃子的肯定是微软。鉴于AOP如此火热，本刊下一期即以此作为“特别策划”内容。

关注：国际程序设计竞赛

不知道大家是否注意到了，近日在IEEE计算机学会主办的第四届国际设计竞赛（CSIDC）激烈的竞争中，来自我国台湾省的台湾大学代表队以一个“基于

Tablet PC的在线学习系统NEWS”从170多支代表队中脱颖而出，并在最后举行的决赛中力拔头筹，并夺得微软多媒体奖。当然，佳绩的取得决非偶然，早在2000年第一届CSIDC上台湾大学就获得过亚军。二到五名的代表队分别是：罗马尼亚布加勒斯特Politehnica大学，美国北卡罗来纳大



台湾大学带队教练赖飞黑教授和队员
郭美辰、黄宝莹、郑先廷、林柏均

学，巴西Engenharia军事学院，加拿大DeVry Calgary大学。获得微软软件工程奖的是巴西Engenharia军事学院。遗憾的是，在获奖名单中没有出现大陆院校的名字。看来CSIDC在国内还需要进一步的推广。

IEEE CSIDC虽然举办时间不长，但是由于别具特色，注重工程实践（参赛内容是真实的项目），隐隐然已经开始与有20多年历史的ACM主办的国际计算机程序设计竞赛（简称ICPC，比赛形式是现场解答题目）并驾齐驱了。今年的ICPC决赛是3月份进行的，冠军被波兰的华沙大学获得，他们解出了全部9道题。来自中国的清华大学、上海交大、中山大学分别获得了第5、6和8名。去年上海交大代表队勇夺冠军，着实令人激动。这两个大赛之间的微妙的竞争关系不是偶然的，它们后面站着的分别是微软和IBM两大巨人。据本刊分析，CSIDC很可能是ICPC被IBM包办以后，微软另辟蹊径运作的产物，这无疑对于广大学子是一件大好事。

Google也将举办一个程序设计比赛：Google Code Jam 2003。这实际上是Google选拔人才的一种方式。头名奖金1万美元，前几轮比赛都在网上进行，而且中国是9个有资格参赛的国家之一，机会非常难得。遗憾的是，当你读到本期杂志的时候，比赛的注册期10月1日已经过了。不过，还是可

以访问比赛的网站，了解题型，明年再来。

专利：不可小视的问题

最近闹得沸沸扬扬的欧盟专利法案修正案9月24日在欧洲议会投票通过。修正案主要修改的是这样一条与计算机发明有关的条款：“数学方法、方案和智力活动的规则，业务方法和计算机程序都不能申请专利。”而按照新的条文，法官将很难拒绝算法和Amazon网站著名的“One Click Shopping”这样的业务方法专利申请请求。欧盟做出此项修改的目的是效仿美国，促进软件业的发展。但很多人认为此举对于自由软件、开源软件和中小企业影响很大，所以引起了网上和真实世界中声势浩大的抗议浪潮，Richard Stallman, Linus Torvalds和Alan Cox都对此发表声明表示反对，许多经济学家也批评了“软件专利能够促进经济发展”的概念。投票因此从月初，推迟到月末，对原提案做出了一些相应的修改（比如排除了计算机使用方面的专利申请许可），但是总的方向没有变。有兴趣的读者可以访问 <http://swpat.ffii.org/index.en.html>。

软件专利的威力有多大？让我们再来看一条新闻。虽然微软应对诉讼已经身经百战，而且取得了各种辉煌胜利，但是她也不总是那么幸运。美国法院最近裁定，微软的IE产品侵犯了Eolas公司的专利，赔偿5亿2100万美元。虽然微软还要上诉，最后结果还不能预料，但是这件事情却提醒我们要特别关注软件中的专利问题。你能想象得到微软侵犯的是什么专利技术吗？在网页中嵌入各种插件！也就是说，你在软件中设计加入applet, scriptlet, 以及ActiveX等等插件都是要交钱的！这场官司如果微软最后败诉，Eolas还有一长串对象可以照此办理呢。Eolas何许人也？她是专利持有人Mike Doyle创办的公司。Mike当年在加利福尼亚大学做研究时，获得了多项专利，其中就包括网页插件。此外，该公司还曾是@符号的持有者，早先时候被IBM出高价买去。

（更多新闻和评论，请访问www.ddjchina.com）



期刊与我

——我对《软件研发》的期待与期许

在科技界讨生活，不能不看科技期刊；在瞬息万变的IT界尤其不能不看期刊，否则无异提早为自己的职场生涯划句号。DDJ、CUJ、WDJ、SDM……（注），就像我的老朋友，我阅读她们已经超过15年——打从我进入计算机领域开始。（注：DDJ、CUJ、WDJ、SDM分别是Dr. Dobb's Journal、C/C++ Users Journal、Windows Developer Journal、Software Development Magazine的缩写）

这些期刊当中，以DDJ历史最久，定位也最超拔广泛，计算机技术领域（软件、硬件、编程语言、编程技术、操作平台……）无所不包。CUJ和WDJ主攻明确，分别探索C/C++领域最前沿的思想和Windows平台上的实用技术。这些刊物虽没有纯学术的耀眼冠冕，但也绝非一般杂志（magazine）可比，而是实用技术性极强的期刊（journal）。

15年的阅读岁月中，这些刊物成了我生活的一部分。日常工作繁重的我，不一定时时能够充分享受这些食粮，每次出国旅游，我总要挑选10来本这些刊物，在飞机上、在平静的异地夜晚，安心愉快地享受这些丰富大餐。朋友乃至家人，屡屡奇怪为什么我在旅游之中还要读书。他们不能明白，对一个享受技术的人而言，不受干扰地沉浸在这些美妙的文章与思维中，是一场多么美好的飨宴。

我曾经为了搜罗这些期刊的过期版本（back issues）致信出版公司或上网洽购。令人惊讶的是，它们的过期版本比当期版本贵上许多（印象中几近两倍）。一方面固然由于订阅与零售待遇不同，

一方面也看得出来，出版公司对于这些刊物有着十足的信心，确信其价值不因短暂时光流逝而有所损耗。事实上这些刊物的确值得这些信心——三两年内的文章往往还如吉光片羽，仍能予读者以实用或启发。究其原因，这些刊物谈的是技术，做的是学问，不是浮光掠影的报导与简介。

这些期刊的专栏主笔人皆是领域专家，技术领先业界，思想与行动多有超前。任何人到达相当水平，最欠缺最需要的，正是超前于现实的“启发”。我尝说，工厂与实验室，缺一不可，这些刊物很多成分正是扮演了“实验室”的前导功用。区区小数，便换来一流专家的思想结晶和呕心沥血的文章，我实在觉得世上没有更高报酬率的买卖。：)

日前欣闻华章公司取得DDJ、CUJ、WDJ、SDM的中文版权，为国内读者推出《软件研发》刊物。华章公司在外版书、外文期刊方面的认知、魄力和行动力，在在“敢为天下先”，诚可敬也。工业界的发展与未来，完全倚重厚实教育培养出的扎实根苗；IT教育最需要的，是大批高质量、接轨全球的读物——书籍如此，期刊杂志亦如此。能够在世界一流的工业期刊基础之上发展，事半功倍，非常值得庆幸与期待。当然，外文刊物所取得的荣耀，属于外文刊物。我们冀望，以这些一流外文刊物为底蕴的《软件研发》，也做出属于自己的荣光，庶几不负所有读者的期待与期许。

侯捷 于台湾新竹



收到了第2期，觉得和第1期比，内容的确明显增多，呵呵，这是一个好现象。

我觉得《软件研发》的主题非常鲜明是一个很大的优点。在章节处理上有一些问题，因为在每个栏目板块之间应该有明显的标记，比如空一页或者什么的，不然我一路看起来感觉主题混乱。技术上呢分布比较广，可能是想抓住所有的技术读者的缘故。

我想从技术上讲，软件文章可以分成几个档次：

1. 语言级 (Java、C++、嵌入式……)
2. 思想级 (Design、Analysis、Modeling)
3. 软件工程级 (RUP、XP、Agile)
4. 管理级
5. 新技术 (Web服务也算半新的技术吧)

我认为如果走高端路线，就该侧重：

- 技术型，国外偏向这一类多，他们喜欢思想级，软件工程级
- 管理型，国内这一类不少，喜欢软件工程级，管理级
- 技术文化，历史和新闻的大家都喜欢(我最喜欢这些)

所以我想贵杂志应该以新技术为主线，附带技术型或者管理型喜欢的文章，关于语言的文章应该减少，但不能没有。这样才能吸引我们的视线。这一期反映语言的部分，比如C++的文章太多，连我这个比较喜欢C++的读者，都觉得太多。

—— Jacob Mee

编辑的回复：非常感谢侯捷先生的鼓励，还有Jacob读者如此高屋建瓴的建议。我们的内容来自世界级的杂志，当然不能在新技术方面逊色，我们会不懈地努力！但是也应该看到，技术的发展其实万变不离其宗，比较一下《Dr. Dobb's Journal》杂志几年前和现在的全年编辑计划、目录和文章，你就会感觉到这一点，许多技术、方法其实本质上的东西并没有变（这些是为什么DDJ的文章过刊反而卖高价，几年后仍然可以以文章为单位销售的原因），只不过表现形式、应用环境和层次等等发生了变化了而已。另一位读者Lebrun就告诫我们，不要盲目地陷入hype和buzzword里面去。这一点我们将谨记于心。

敏捷革命 (下)

上回只因主编有令,小史不得不跑题去搜罗一些Web服务发展历史中的重要故事。闲话少叙,我们言归正传,本回继续讲述敏捷十七雄2001年雪鸟之会的盛况。看官们可以从中了解到敏捷方法的各种思想根源:微软公司的实践、RAD、耐克公司产品设计过程、化学过程解释、软件工程思想。还能读到奇才 Jeff De Luca 的传奇经历。

却说2001年2月的犹他州Wasatch山中的Snowbird度假胜地,许多软件开发界的重量级人物聚会于此。会议伊始,6位近来声名显赫的XP(极限编程)方法领导人先行介绍了XP的发展历程。负责主讲的Ward Cunningham本来就极善言谈而且思维跳跃,又有业界著名的演说家和博学之士Martin Fowler和Robert Martin的帮衬,更是口若悬河,如滔滔江水奔流不止了。眼看着他的话题已经开始转到大谈特谈心爱的WikiWikiWeb,时而又回忆起自己在普度大学时与英年早逝的Krulinski^①如何在机房里纵意驰骋。主持人Kent Beck看看时间渐晚,只能笑着将其打断:“许多人称我为XP之父,然而,我必须承认,如果说这些年来我对社区还算有一些贡献的话,那么我的所有贡献中,都离不开Ward对我的影响,我相信在座的很多人也已经从Ward的众多天才创造中获益匪浅。”会议室内此时响起一片掌声。

“下面还是由我来说说这次会议的吧。”Kent Beck继续说道:“事情是这样的,去年春天,在我家里,俄勒冈州乡

下的一个大庄园,开了一次会,与会的有许多XP倡导者,当然也有Jim(指Jim Highsmith)、Alistair(指Alistair Cockburn)和Dave(指Dave Thomas)等几位XP圈子外面的同道。大家都感到,许多从实践中得出来的开发方法,其实都是多年来整个业界同仁共同的经验积淀,有很大的共性,而要使这些方法被更多人接受和采用,应该团结起来。而且去年开始,已经有很多文章把我们的东西合称为‘轻型’或者‘轻量级’(lightweight)方法了。但是,那次会议上的很多人,我相信在座各位也一样,都不太喜欢这样的称呼。”这时候有人插话了:“对,我是反应最强烈的一个。这个名字有诬蔑性,将我的方法学称为轻型的,也许不是不能接受,但是我肯定不想自己被人说着说着,就变成了参加轻量级方法学者会议的轻量级的家伙。这听上去好像是说一群个头小而且弱智的家伙凑在一起,互相询问:‘喂,老兄,今天几号啦?’”座中一片笑声。这个长着络腮胡子,显得非常憨厚的人就是Crystal方法的创始人Alistair Cockburn。

“是的,”一位坐在前排的老者接着说,“我本来并不觉得轻型、轻量级有什么太大的问题,自己有时候也这样用。可是接到Alistair的信以后,我的看法改变了。”等他站起来,大家顺着灯光看去,原来说话的此人乃是ASD(自适应软件开发)方法的创始人Jim Highsmith。座中又有几个人发表意见,表示对lightweight的不满。

“所以,”Kent Beck接着说,“本次会议的第一个议题就是为我们的方法取一个共同的名字。对了,有一点我要说明,本次会议的始作俑者应该是Bob Martin。Bob在去年9月份给我们中的很多人发了一封邮件,提议今年一二月份在芝加哥召开一次小型会议,把各种轻型方法的研究者召集起来共商大计。”“Bob还很快为此在我的Wiki上设立了网页,”好久

注1: Krulinski是广受欢迎的《Visual C++技术内幕》一书的作者,酷爱极限运动,1997年4月遭遇滑翔机事故,随风而逝。



Jon Kern在成为独立咨询顾问之前,曾担任TogetherSoft公司研发和专业服务业务的总监兼副总裁,是Together Control Center的主要负责人。他早年曾为美国国防部工作,在巡航导弹引擎测试项目中担任过工程师。1995~1998年他自己创办的公司曾为IBM开发过几百万行的大型系统。Jon与Peter Coad合著了《Java Design》一书,在特性驱动开发(FDD)方法的创立过程中做出了很大贡献。他还经常担任网上期刊Coad Letter的编辑。



Alistair Cockburn是软件工程界尤其是用例方面的权威,以见闻广博和经历丰富著称。Ward Cunningham称Alistair是软件工程界真正的方法学家。他的父亲是一位联合国工作人员,所以从小就游历世界各国,从斯里兰卡、孟加拉到南非。他先后在美国、瑞典和苏格兰读书,曾在IBM苏黎世研究院、IBM咨询集团和挪威中央银行供职。而长期的顾问工作更给了他观察世界各种项目的机会。例如,1997年在挪威中央银行工作时,公司同时有几种类型的项目在进行:35人基于COBOL的2000年问题项目,两个人的

Java/CORBA项目和一个人的SQL信息恢复项目,等等。这些项目显然无法运用同一种理论,需要因地制宜,然而政策却不允许这样做。多年的经验积累,Alistair从中总结出软件开发是“一种发明与交流的合作性工作”,强调人与人的相互影响、团队、技能、才智和交流的重要性,同时给出了每个项目一种方法学的概念,这些最终演变为一种三维框架式的轻型方法——“Crystal”系列方法。他的《编写有效用例》(中文版由机械工业出版社出版)在国内产生了较大影响。即将出版的《敏捷软件开发》也是敏捷运动的基本文献。

没有说话的Ward Cunningham接过了话头,“大家对会议的讨论很热烈。我没有想到的是,争得最多的居然是开会的地点。我估计Martin压根儿就没诚心。一二月份的芝加哥,那么冷,连玩儿的地方都没有。(大伙儿都忍着不笑。)有人提议去加勒比海上的安瓜拉,晒晒太阳,游游泳,多舒服。可是在座的很多都是老板呐,名人啦,哪里有那么长时间跑到那么远。最后得票最多的,就是这里啦。虽然冷,但是很好玩儿呀。滑雪嘛。多有趣,至少有一个人可以学着用头滑雪。”大家都看着Martin Fowler上午滑雪摔了跤以后更加突出的秃头,再也忍不住了,纷纷暴笑起来。

“我还是觉得应该去暖和一些的地方,”有人说,“至少下一次应该换一个地方。”说话的是来自TogetherSoft公司的Jon Kern。因为长期生活在大西洋边温暖的北卡罗来纳州,他可不怎么喜欢冬天。

Kent Beck笑着说:“Jon,知道么,你能来我们都感到非常荣幸。大家都清楚TogetherSoft公司和Peter Coad先生在软件工程界的地位。从大项目中产生的FDD方法,对我们都有很大的启发。我提议Jon先把FDD方法介绍一下。”



Jim Highsmith写的关于ASD的同名书籍(中文版由清华大学出版社出版)获得了本刊英文版之一《Software Development》颁发的2000年Jolt大奖。有意思的是,当10多年前,Ward Cunningham和Kent Beck开始极限之旅的时候,Jim却在实施最传统的一种开发方法——数据结构化系统开发。这并不奇怪,Jim从事软件开发这一行已经30多年了,他的第一项工作是测试用于跟踪阿波罗宇航飞船返回的软件和设备。此后还攻读了管理学硕士。27岁时被提升为财务经理。此外,担任过咨询、程序员、项目经理、开发经理、销售经理和方法学专家,其间经历过原始方法学、信息工程、RAD。经验极为丰富。20世纪90年代早期,他在微软做了几年咨询工作,发现微软并没有使用通常的严格方法。与一般方法学注重前期详细的需求、架构、建模不同,他们非常注重实际,注重项目周期的结尾:编码和测试、代码检查等等,需求往往通过交谈、使用白板进行,也采用了迭代里程碑、特性驱动等技术,整个开发过程中经常回顾,而且强调人们之间

的紧密合作。此后,他与Sam Bayer合作,后者的RAD背景对Jim的影响也很大。90年代中期,Jim在耐克公司工作,耐克公司在高效的产品设计过程中,采用了“阶段和闸门开发”的概念,在关键节点上进行评估,节点之间的创造性活动很少讨论如何做,只看结果。这种方法不需要太多的过程和文档,非常注重人本身。这些使Jim从原先的传统方法学转向了自适应软件开发。

目前Alistair和Jim Highsmith正在协同工作,通过将通用的理论与项目团队面临的特定情况结合起来,致力于发展“Crystal”与“适应性理论”(Adaptive ideas),并将它们推荐给对敏捷软件开发系统的建立。此次会后,Jim Highsmith和Martin Fowler合撰的《敏捷宣言》(参见本刊第1期,2003年8月)成为敏捷方法的基本文献,而2002年出版的《敏捷软件开发生态系统》(中文版即将由机械工业出版社出版)一书更是最全面介绍了敏捷方法的圣经式著作。2002年出版的《敏捷软件开发生态系统》(中文版即将由机械工业出版社出版)一书更是最全面介绍了敏捷方法的圣经式著作。



没有参加 Snowbird 大会的 Jeff De Luca 是敏捷方法社区中最具传奇色彩的人物。他的成功故事完全可以拍成电影，足以

激励着更多人去勇于实现自己的梦想。谁能想到这位如今已大名鼎鼎的 IT 高级管理和技术专家的 IT 生涯却是从 IBM 澳大利亚公司的收发室开始的——1981 年，他 17 岁时因为家境原因失学，成了临时的邮件收发员。虽然大学的门一天都没有进过，但是他的天才很快显露出来——他重新设计了公司的邮件系统，结果自己原本要 8 小时完成的工作，只需 50 分钟就干完了，这使他成了 IBM 的正式员工。Jeff 在 IBM 公司浓厚的信息技术氛围中耳濡目染，渐渐也对计算机发生了兴趣。有一天，他读到了公司高层阅读的杂志《Think》上的一篇文章，决心此生应该成为一名系统工程师。现实很快让 Jeff 碰了壁：公司规定，工程师需要有大学学位。然而，Jeff 并不气馁，他利用业余时间

自学系统技术。由于自己设计的邮件处理系统效率很高，他能够在上班时间内很快干完本职工作，经常跑去帮程序员干活，并从中学习。他的灵气很快得到了认可，他得到了一个操作员的职位，负责管理事宜和系统硬件。在此期间，Jeff 还开始修改操作系统，编写一些小程序，解决工作中的问题。IBM 的高层很快注意到这个小伙子有着发明创造的非凡才能，送他到美国培训 6 个月。这一下不得了了：Jeff 被看作是具有巨大潜力的系统工程人员，一步升了 32 级（IBM 的级别够多的！），成为助理系统工程师。这时候，他仍然没有学位。此后，Jeff 就像坐上了直升梯：他到了著名的罗彻斯特城堡为操作系统开发网络和系统软件，发明了一项编程技术，设计了领先的各种系统之间联网的 Object Distribution Facility/36。到了 25 岁的时候，他的人生梦想实现了——他成了系统工程师。事情还没有结束：Jeff 又迅速地连续提升为各领域的负责职位，成为项目管理人员。他经手的项目都非常成功，曾不可思议地连续五次获得 IBM 一等奖。两次获得公

司最高奖——杰出创新成就奖。1993 年，Jeff 感觉在 IBM 已经学不到什么东西了，在高级系统策略师的职位上提出辞职。这一职位通常需要花 18 年才能达到，而 Jeff 只用了 11 年。

离开 IBM 以后，Jeff 创建了自己的咨询公司 Nebulon。在担任顾问的所有项目（电信、银行、保险等行业）中，保持着百分之一百的成功率。



Peter Coad 是面向对象技术的先驱之一，著名的 Coad 方法（又称 Coad/Yourdon 方法）创始人。他于 1981 年获得南加州大学

计算机硕士，1986 年创办 Object 国际公司，后改名为 TogetherSoft，是仅次于 Rational 的软件工程工具厂商。他被人称为世界上最有经验的对象建模大师，曾经开发过几百个模型，涉及几乎所有行业和领域。近年来创造的彩色建模方法和 FDD 均独树一帜。TogetherSoft 并入 Borland 以后，他出任高级副总裁兼首席策略官。

Jon Kern 讲述的 FDD 故事

FDD 是在著名的新加坡项目中最后成型的。这个项目是一家大银行的大型商业贷款系统，当时东南亚地区最大的一个 Java 项目之一。原来承包项目的一家知名大型系统集成公司在两年之后做不下去了，而交付的结果却很具有代表性：3500 个用例，一个对象模型（内含数百个类），数千个属性（没有方法），一行代码都没有。银行只好请来东南亚和澳洲一带有名的项目救火队员——墨尔本 Nebulon 公司的 Jeff De Luca。而 Jeff 看出这个项目大有病入膏肓之象，而且问题领域极为复杂，建模工作难度巨大，非盖世名医不足以妙手回春。他很快联系了 TogetherSoft

公司的总裁 Peter Coad，请 Peter 来担任项目的首席架构师。自己则承担了项目经理的职责。由 50 人组成的这个世界顶级开发团队开始了抢救行动。15 个月以后，项目成功交付了 2000 项特性，为银行带出了一只训练有素的队伍，而且实现了初期定下的目标：客户、经理和开发人员皆大欢喜，银行 CEO 发来了热情洋溢的贺信。

当然，FDD 的形成并非一蹴而就。在新加坡项目之前，Jeff 就已经总结出一种流线型、轻过程的开发框架，这里面凝聚了他自己多年的开发和项目管理经验，也融入了过去几十年业界的思想和实践，包括 Harlan Mill 的“外科手术式团队”，Fred Brooks 的

“保持概念整体性的前提下使项目规模具有可伸缩性”，Jerry Weinberg 的将软件构造视作人类活动，Tom DeMarco 对项目本质的深入认识，还有 Michael Fagan 的软件审查。而 Peter Coad 也带来了自己的长期经验结晶——以客户价值为中心的“特性”概念。这两种思想和相应技术的撞击，产生了许多创新，除了 FDD 外，还有富于想像的彩色建模方法。这些成果在两人合著的《Java Modeling Color with UML》一书中有充分的反映。

此后，FDD 开始应用于众多项目。包括另一个大银行项目，动用了 250 人，18 个月。此外，著名的 Together Control Center 就是在俄罗斯采用 FDD 方法开发的。

“谢谢 Jon 的回忆。FDD 的故事让我们看到了软件开发中是人而不是过程占据着主导地位，这一点我想大家已经形成了共识。” Jon Kern 的介绍结束后，Kent Beck 接过了话头，“我们此次会议的中心议题应该是寻找共识。如果能够最终形成一些共同原则的话，将极大地推动我们整个运动。当然，有些人可能认为这么多方法学者在一起，不可能达成什么共识。”他看了一眼 Alistair Cockburn（“对，” Alistair 笑了，“我是这么想的。”），然后接着说：“这样好了，大家分别提出自己认为比较重要的原则，然后再共同讨论。可以先介绍自己的方法，再提出原则。SCRUM 方法的三位倡导者都来了，哪一位给我们介绍一下？我要说明的一点是，XP 中的计划游戏实践是从 SCRUM 借鉴过来的。”

“好啦，下面应该轮到来自欧洲的 DSDM 方法的代表发言了。DSDM 可能是我们这些方法中历史最为悠久的，在欧洲各国都有很大影响。” Kent Beck 说：“大家可能还不认识，有请 Arie van Bennekum 先生。他是今天坐飞机从荷兰刚刚赶到的。”

一头金发非常高大英俊的 Bennekum，操着一口北欧腔的英语开始讲话。

“我们还高兴地看到有来自软件测试界的同道 Brian Marick、面向对象的先驱 Steve Mellor 和大家都很熟悉的 Pragmatic Programmers 双人组 Andy 和 Dave 参加我们的会议。我相信他们的不同背景和丰富经验能够极大地帮助我们，并把我们的方法扩展到其他领域，比如说测试、实时系统等等。”会议告一段落，Kent Beck 建议休会。可是大家显然意犹未尽，于是亲朋好友地三个一群两个一伙自发地开始了分组讨论。有些人说着说着，还找到一块白板，写写画画。天色已暗，会议室里的灯光并不很亮，似乎能看见思想的火光在四处溅射。

两天的会议结束的时候，各种方法有了共同的名字——敏捷，大家都对这个上口的名字非常满意，因为这个词体现了最重要的概念：能够对变化做出迅速反应。只有 Martin Fowler 除外——他们英国人对这个词的读法和大家都不一样。这期间，17 个人求同存异，居然在许多关键问题上都取得了一致，找到了不

少普遍认同的价值观。最后甚至有了一份由四条原则组成的宣言（参见本刊第 1 期），大家都在宣言上签了字。之所以名为宣言，是因为大家都感到应该对整个开发社区大声疾呼，应该对现状发出战斗的口号。后面的目的是接下来热烈的 Wiki 讨论和频繁电子文件交换的结果。

2001 年 10 月的 OOPSLA 会议上，17 个人中大部分又得以重聚，他们得以有机会评估过去几个月中的进展。不知是谁提议就此成立一个组织，这得到了大家的一致响应，名字很快也有了——Agile Alliance。Ward Cunningham 主动承担了获取域名 www.agilealliance.com 并建设网站的任务，网站上放有宣言的内容，还有 Jim Highsmith 写的一篇短文，和签名者的链接。

不久以后 Jim Highsmith 和 Martin Fowler 合作，撰写了一篇文章，发表在 2002 年的《Software Development》杂志上（参见本刊第 1 期），敏捷会议也很快召开，从此，敏捷运动开始逐步进入了主流。

Dr. Dobbs
JOURNAL CHINA

SCRUM 的发展历程

SCRUM 是 Jeff Sutherland 和 Ken Schwaber 合作的产物。1993 年 Sutherland 在 Easel 公司任主管对象技术的副总裁时开始试验许多 SCRUM 的基本思想，第一个项目是开发具有正反双向工程（round-trip engineering）功能的面向对象设计和分析工具。与此同时，已经用传统方法开发和帮助别人开发了几十年软件的 Ken Schwaber 开始对自己的行业产生了疑惑，因为原有的方法似乎总是无法适应客户的需求，进度和质量都很难尽如人意。一个偶然的机会，他将这一问题向一位杜邦公司的化学过程专家请教，专家的解答使 Schwaber 茅塞顿开：软件开发和化学反应类似，是一种非线性的过程，它从理论上讲是无法精确预测的，只有不断根据获取反馈信息，调整参数，才能最大程度地逼近正确值。

1995 年，Easel 被 VMARK 兼并，Sutherland 也随之加入，就是在这里他遇到了在公司做咨询的 Ken Schwaber。此后两人又先后在 Individual 公司和 IDX 公司（世界最大的医疗软件公司之一）继续合作。SCRUM 就此诞生了。2001 年，Mike Beedle 和 Ken Schwaber 一起撰写了《Agile Software Development with SCRUM》一书（Prentice Hall，2001 秋出版），使 SCRUM 进入了公众视野。

DSDM 简史

20 世纪 90 年代，快速应用开发（Rapid Application Development, RAD）曾经是 IT 业的时尚。它所要解决的也是传统瀑布式方法不能适应快速变化的商业环境的问题。然而，各种 RAD 过程出自各种背景，没有共同的定义，各厂商和顾问都有自己的解释和方法。

针对这种现状，1994 年 1 月，16 个厂家组成了 DSDM (Dynamic Systems Development Method, 动态系统开发方法) 协会，旨在共同开发和推动统一的 RAD 框架。RAD 的发展已经为协会提供了足够的经验，很快在同年 3 月的会议上，高层框架被 36 名成员一致通过了。自那以后，虽然框架在一直发展，但是基本的概念保持不变。目前的版本已经达到 4.1。DSDM 在英国是最常用的 RAD 方法，而且它已经逐渐取得了国际声望，在欧洲逐渐推开。虽然 DSDM 最初是针对 IT 业的，但是现在它已经推广到各个行业。



Christopher Alexander

曾是《Residential Architect》
的封面人物

Christopher Alexander

模式思想先驱

大凡讲述软件模式的文章和书籍，总少不了把模式的源头追溯到建筑学家 Christopher Alexander，广大软件开发者也因此熟悉了这位思想者，许多人逐渐成为他的思想信徒（国内有不少学习设计模式的开发人员都购买了他 1979 年写的《建筑的永恒之道》和《模式语言》中文版，我相信这肯定是出版社学建筑的编辑始料未及的），甚至有不少对他敬若神明。那么，Christopher Alexander 到底是个怎样的人呢？

Christopher Alexander，公元 1936 年 10 月 4 日出生于奥地利维也纳，英国国籍，父母亲都是考古学家。1938 年德军入侵时随父母逃往英国。他毕业于剑桥大学，获得的是建筑学学士学位和数学硕士学位。在剑桥大学，有一次的作业是设计一所房子，他按照课堂上讲授的形式主义理论故意交上去一份戏作：一个玻璃大盒子，被巨大的砖墙切割开来。这完全是一个抽象、缺乏意义的概念。不料，怪事情发生了：系主任把他叫到办公室，高兴地说：“祝贺你，我的孩子！这正是我们想要的。” Alexander 当时想：“天哪，我是不是进了疯人院？”

带着对英国建筑教育的失望之情，1958 年他远渡重洋，在哈佛大学攻读建筑学博士学位，1964 年他的博士论文正式以书的形式出版，这就是《Notes on the Synthesis of Form》（形式合成论）。其中他提出了严格而且至少超前 10 年的建筑设计方法，采用了算法和计算机辅助分析技术。这使他声名鹊起，获得了美国建筑学会第一个研究金奖。

Alexander 曾于 1962 年在印度工作过一年，期间印度政府曾请他帮助重建一座大坝移民村落，但是最后的效果并不令人满意。他发现，其中的主要原因是整个设计和建设过程中村子里的人没有积极地参与进来。印度的经历使他开始思考人与建筑的关系。他针对建筑设计的日益形式化，提出建筑的设计不应该由设计师主观地自顶向下地决定一切，而应该由房屋的主人自己掌握规律，做出决策。这很像建筑设计 DIY 的概念。其中的理论基础是，建筑学与其他科学一样，也存在客观的规律性，比如和谐 (harmony)、浑然一体 (wholeness)，他称之为“无名特质” (quality without a name)。

数学背景使 Alexander 能够以与众不同的角度看待世界，一方面他的思考经常呈现出哲学化甚至神秘化的形态，另一方面，他采用科学推理的方法，试图找到一组规律（这种规律就是 he 所说的模式），能够决定一个城市、一座建筑、一间房屋的结构，这与他的同行以时尚、意识形态和个人喜好为基础的非科学方法截然不同。

1963 年，Alexander 到加州大学伯克利分校任教。1965 年，Alexander 发表了题为“城市不是树”的论文，以独特的视角讨论了城市规划的一些问题，至今仍然是此领域的经典文献。

此后他根据自己的理论，在 5 位合作者、同事和学生的帮助下，经过 10 多年时间，收集了 253 个模式，从“建筑不能超过四层”（模式 21）到“应该面朝东方睡眠”（模式 138），无所不包，而且按 Alexander 的说法，这些模式 500 年来就一直深深植根于事物本质之中。1977 年，成果由剑桥大学出版社正式出版，这就是著名的《A Pattern Language》（模式语言），用很薄的纸印刷而成，将近 1200 页，外观、分量和威严的口气，都和圣经很像。

当时，建筑学界正处在现代主义和后现代主义之间的过渡期。Alexander的书揭示了当时占主流但是已经有些日暮西山的现代主义建筑的弱点，对于普通大众而言也是一本讲授如何自行设计和建造房屋的指南，因此取得了极大的关注，书非常畅销。直至今日，据说这本书每年还能卖出1万本。

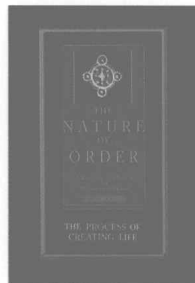
1980年 Alexander 被选为瑞典皇家学院（这可是评选诺贝尔奖的机构）院士，声望达到了顶峰。也正是在20世纪80年代早期，计算机科学家们开始热烈地讨论 Alexander 的著作，并且似乎找到了解决软件设计问题的方案。1995年受《A Pattern Language》启发的《设计模式》一书出版后，软件设计模式逐渐成为主流甚至必备的技术，成千上万的程序员开始朝夕与之相处。Alexander 在软件开发界的地位也如日中天，近乎于神。除 Sun 公司的 Gabriel 之外，“Sims”系列游戏的开发者 Will Write 也多次说过，Alexander 是对他影响最大的人。

然而，随着无视规则的解构主义和后现代主义的兴起，整个现代建筑的趋势与 Alexander 已经大异其趣，加上自己没有太多拿得出手的作品，Alexander 在建筑界的地位急剧下降了。Alexander 的

方法本来就有片面强调客观的倾向，而他本人到了后期更进一步将理论弄得有些玄学化，宗教意味非常重，引起了许多人的反感。他的理论本来就没有得到太多同行的认同，这下更被人视为激进和异端。糟糕的是，他似乎不善于处理人际关系，与周围的环境此时也开始格格不入。他和同事们的关系越来越差，不断发生冲突。到了1985年，他甚至写信给校方说自己和学生受到了迫害，要求给予学术自由。事情最终不了了之。1998年他在伯克利退休。之前还算幸运，获得了美国艺术与科学院院士的称号。

虽然已经被主流建筑学界视为边缘，但是 Alexander 仍然我行我素，每天早上2点钟就起床，继续自己宏大的乌托邦式的工作。后来，他和自己在剑桥大学的编辑也翻了脸，于是新著转由自己创办的公司出版。好在他的拥趸很多，而且还包括查尔斯王子这样的人。Sun 公司对他也有不少资助。今年早些时候，经过漫长的27年的辛苦劳作，他的四卷本巨著《The Nature of Order》（秩序的本质）终于问世，全书一共2150页。书的副标题为：关于建筑艺术和宇宙本质的随笔，主题是生命本身。“生命不是应用于自我繁殖的生物机器的一种有限的机械概念，”他在

序言中说。“他是一种存在于空间本身的特质，可以应用于每一片砖，每一块石头，每个人，每个出现在空间中的任何种类的物理结构。每个事物都有生命。”



《秩序的本质》第二卷的封面

《纽约时报》最近对这位名人进行了采访，文中是这样描写的：“Alexander 先生有着熊一般的身材，有一双忧郁的蓝眼睛，穿着皱皱巴巴的卡其布服装，看上去饱受风霜，不像传说中那样脾气暴躁。”

Christopher Alexander，曾经雄心勃勃地改变世界，最终却被大多数人看成是一次堂吉珂德式的尝试。



Christopher Alexander 近照，背景是他自己设计的英国西苏塞克斯的 West Dean 花园游客中心，他在此长大。

建筑界对 Christopher Alexander 理论的评论

Alexander 的亲密战友，德克萨斯大学数学教授建筑专家 Nikos A. Salingaros 这样说：“他（Alexander）的发现实际上宣判了大多数执业建筑设计的无效性。因此，建筑界将模式语言视为严重的威胁，进行压制。对其进行公开的攻击，只会让这种思想传布更广，因此在建筑学院、专业会议和出版物上提到它的时候，总会立即被故意地斥为无关紧要，而一笔带过。”

《The Nature of Order》这本书我在 Berkeley 上他课的时候正处在最后定稿阶段。我们整班就成了他的小白鼠，试看了

他的教材。真的是很失望。如果说《A Pattern Language》还有很多可取的东西，那他的新作真的可取的地方不多。Christopher 在以前的《A Pattern Language》中就列出了好的城市设计和建筑需要具备的条件，其中真的有很多是很好的例子。但建筑不是开中药，按药方，这样加些，那样加些，就可以配到一道治病的良药。正因为他是学数学的，所以就希望通过逻辑的方法把好的建筑的条件都归纳起来，以后把这些条件都拿出来一齐用就好了。他和他的崇拜者对于他的理论和他的《A Pattern Language》就像是人对宗教一样。新的《The

Nature of Order》更是如此，他希望通过系统地归纳，把自然界的美都归纳起来。但这次的归纳更加随意和唯心，追求的是心灵的认同，精神上的慰藉。全书的说明缥缈虚无，全都是 Alexander 个人感觉为中心的理论。最近一位建筑理论评论家在《Architecture》杂志上把他的书批得一文不值，Alexander 在 Berkeley 教书30年的生涯已经结束，Berkeley 确实也该换换新血了。”

——留美中国建筑学者

大师圆桌会: 定义 UML 内核

面对越来越复杂的系统和越来越紧张的时间,新一代的建模人员需要采用标准建模语言来工作和交流。现在,UML 2.0 标准已经被 OMG 正式批准为推荐采用标准,正处在最后审查 (finalization) 阶段,新的标准分成四个部分,总计 1000 多页。面对如此庞大的标准,分析人员应该如何看待和使用呢?最近,《Software Development》杂志召开了一次圆桌会,邀请了业界几位重量级的人士参加,主题是讨论 UML 的发展方向 and 评价 UML 2.0 中内核的定义。

主持:《Software Development》杂志

大师圆桌会出席者:



Cris Kobryn, Telelogic 公司的 CTO, OMG UML 修订任务组和 OMG 分析设计平台任务组两主席之一



Martin Fowler, ThoughtWorks 公司首席科学家,软件工程专家,《UML Distilled》作者



Scott Ambler, Ronin 国际软件咨询公司总裁,《The Element of UML Style》作者



Peter Coad, Borland 公司副总裁兼首席策略官,《Java Modeling In Color With UML》一书的作者

《Software Development》(SD): Kobryn 先生,作为标准工作组的负责人,能够我们介绍一下 UML 新版标准,还有它的开发过程吗?

Cris Kobryn: 我们知道, UML 1.x 受批评最多的地方,就是它太大、太复杂了,而且因此在 UML 2.0 中我们定义了一个内核 (kernel),它可以看成是基本的 (substantive) UML,使用这一部分就可以对 89% 的常见问题建模。

任何优秀的设计者都会这样设想:新一代 UML 应该做减法,而不是加法。也就是说,设计完成时,应该刚刚好,不多也不少。我们尽力遵循这种优秀设计的古训,因为大多数人没有时间在元层次建模 (metamodeling)。

我们认为 OMG UML 标准开发的过程是非常高效的,事实已经证明了这一点。UML 在许多应用领域都有了大量的成功运用。

整个 UML/OMG 标准的开发过程可以这样描述:一组优秀的方法学家不仅梳

理了语义,而且定义了很好的开发路线图。1.3 版其实已经很成熟了,1.5 是 1.4 和 2.0 之间的过渡。而 2.0 中最关键的就是一个精简的内核,增强了可定制性和对基于构件开发 (比如 EJB、CORBA 和 COM+) 的支持。2.0 版从架构上对内核进行了精简,并使之与其他关于元数据集成和储存的 OMG 规范 (例如元对象设施 MOF) 更加一致。

此外,对模型管理的支持有所改进,包括分层、多视图和构件框架。模型版本管理不属于新版 UML 的范围。图的交换利用了 OMG 的另一个结构规范 XML。

问题是,开发商们对修订会支持到什么程度?很显然建模者都希望 UML 易学易用,我常常看到人们因为缺少工具支持而经常混淆工具带来的局限和 UML 规范本身的局限,这使得他们很难明智地、准确地批评 UML。UML 开发应该是市场驱动的,如果用户要求支持 UML 2.0,开发商就应该提供。

Scott Ambler:在我看来 UML 就是一堆