

<<软件开发之韵>>

图书基本信息

书名：<<软件开发之韵>>

13位ISBN编号：9787121108167

10位ISBN编号：712110816X

出版时间：2010-5

出版时间：电子工业出版社

作者：雷剑文,陈振冲

页数：295

译者：杨艳

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<软件开发之韵>>

前言

本书帮助我们发现自己的软件方法学，这种方法同时尊重人员与实践的软件开发的双重韵律。

深夜，躺在马拉维湖的康帝海滩，我们仰望晴朗的天空。

无数小星星向我们眨着眼。

有一点疲倦，或许是这遥远神秘的光使人迷幻，我们闭上眼睛，开始倾心聆听：湖水轻柔地拍打着岸，深夜那微小和谐的声音，心好像随着有韵律的、深深的呼吸而悸动。

大自然就是一曲不可思议的韵律合奏。

我们的地球在太阳系的轨道上转动不息，日日夜夜，四季循环，永不休止。

随着自然的韵律，我们醒来学习，入睡记忆，写写改改我们的程序，在完美的韵律合奏中与宇宙间全能的程序融为一体。

从心跳到步伐，韵律是动力十足、充满生机的力量。

这世界如此复杂、好似混沌，我们努力追求缘由并确认关联，但有时只有发现韵律才可以让我们看到普遍存在的规律。

就像人类其他的努力一样，软件开发复杂并充满了泛化与关联，但是没有规则。

我们用规范的软件开发模型和管理学帮助构造软件，但是纷扰的软件开发、不断变化的开发团队、新的需求与任务，意味着根本无法保证上一次成功的方法就能在下一次同样成功。

实际上，有的团队领导看上去并没有采用什么开发方法，因而被别人嘲笑是“即兴开发”，他们反而能够按时完成软件项目。

他们成功的秘密就在于能够理解软件开发的韵律。

面对最让人头疼的软件开发问题，了解韵律将给予我们一个新的视角。

一个团队的成功方法有时并不适合另一个团队，因为即使是最积极肯干的软件团队，也必须在理解一种新方法的韵律后，才可以成功地得以运用。

在管理复杂而多元的软件开发项目时，团队和个人的开发与进程之间的和谐运作是最关键的，然而韵律却是一个经常被忽视的主题。

韵律并不是一种新的开发方法。

当前已经有许多种不同的开发方法，本书并不打算介绍一种新的软件开发和项目管理的框架。

目前最需要的是更多的方法，而是更智慧地使用我们选择的方法。

最好的方式是，理解并和谐地在团队所采用的开发方法的韵律下工作。

如果不理解，不在韵律下和谐地工作，则反而会被选用的开发方法拖累而并非从中受益，会使项目冗长艰难。

本书不是为新手所写的。

实际上，我们认为你可以钓鱼并已经钓到了一些鱼了。

本书是为了那些老手，想改进甚至于想重新发现软件开发的技巧与技术的人而写的。

当我们习惯于只从一个单一的角度看问题时，这些东西很容易被遗忘。

就像正在钓鱼的人，用灵巧的手腕，稳定的韵律甩出鱼线，我们希望能帮助你钓到更多的鱼，并帮助你享受这一过程。

<<软件开发之韵>>

内容概要

《软件开发之韵：和谐敏捷、珠联璧合的开发》是一本关于推荐、推广、推崇敏捷开发的软件方法学教材，这种方法同时尊重人员与实践的软件开发的双重韵律。

全书包括两部分，共9章。

第一部分由三章组成。

第1章介绍软件开发韵律的概念，第2章、第3章分别讨论人与实践，阐明软件开发的一些基本概念并提出几个重要的问题，如：“什么是敏捷价值？”

”“从开源软件开发中我们能学到什么”等。

第二部分包括其余的六章，都是关于开发韵律的。

软件开发韵律是一个强大的比喻，可帮助我们分析何时更好地采用一种软件开发的方法，使软件开发实践更加和谐，软件的质量也得以提升。

另外，《软件开发之韵：和谐敏捷、珠联璧合的开发》以软件开发实践中的点滴作为出发点展开讨论，描述了一些项目片段和工业实例，注重用事实说话。

全书行文深入浅出，亲切自然，并配以很多有趣的漫画来阐述书中的概念，值得读者细细品读，定当回味无穷。

适合阅读《软件开发之韵：和谐敏捷、珠联璧合的开发》的，不仅仅是处在软件行业第一线的程序员；各个软件开发单位的团队领导、项目主管、高层管理人员，以及人力资源经理、文档撰写人员、程序开发工具的设计者、程序开发语言的设计者，甚至所有其工作与程序开发有关的人，都能从《软件开发之韵：和谐敏捷、珠联璧合的开发》中得到启发。

<<软件开发之韵>>

作者简介

KIM MAN LUI博士，是一位大学（Hong Kong Polytechnic University）电子计算学系的访问教授。
他是Oracle认证数据库管理员和Sun认证、Java程序员。

KEITH C.C.CHAN博士,是香港理工大学电子计算学系的教授和系主任，他曾经是IBM设在多伦多的IBM加拿大实验室（IBM Canada Laboratory）的高级分析员。

译者简介：杨艳，长安大学信息工程学院讲师。

研究方向包括：智能交通系统、交通安全、驾驶员心理学和软件工程管理。

主要研究兴趣：检测评价车内外驾驶环境，为科学地制定交通策略提供理论依据，并致力于利用改善车内外软硬件环境以提高交通安全。

丁大江，大学毕业后就一直做软件编程，迄今已有十余年，有完成任务的喜悦，更多的是陷入几乎绝望的烦恼。

好的编程方法可以大大缩短编程过程，希望这本书能对大家有所帮助。

倪神昭，福建泉州籍人。

2005年毕业于厦门大学自动化系，同年进入英国伦敦大学研读智能系统硕士学位。

2010年博士毕业于南安普顿大学计算机系，主修机器学习与机器翻译。

王天骄，本科毕业于北京交通大学交通运输学院，取得学士学位。

至本书截稿时，就读于英国南安普敦大学土木与环境工程学院交通运输研究组，攻读博士学位。

研究兴趣包括：道路使用者行为，交通建模与仿真，智能交通系统等。

杨军，总后直属某部助理工程师，管理学、军事学学士，后勤管理信息化硕士。

研究方向：后勤信息化理论与实践、基于多Agent的信息系统协同。

张靖，2004年毕业于长安大学，2005年就读于南安普顿大学交通规划与工程硕士专业，2006年毕业后继续在该校攻读博士学位。

<<软件开发之韵>>

书籍目录

第一部分：基本概念第1章 程序员不死 21.1 开发软件与修建隧道相比 31.1.1 美好的旧时光 31.1.2 情况越变化，他们越相同 41.1.3 软件产品的背后 51.1.4 成交或不成交 81.2 哆来咪哆来咪 101.2.1 迭代模型 121.2.2 编码后修复模型 141.2.3 混沌 151.2.4 重要的方法 191.3 软件开发韵律 221.3.1 五线谱示例 231.3.2 博弈理论 261.3.3 启动 - 结束示图 (In-Out Diagram) 281.3.4 精通 - 培训示图 291.3.5 不用数学 301.3.6 去哪里探索韵律 31参考文献 32第2章 了解程序员 342.1 个性及智力 362.1.1 编程高手 372.1.2 了解你的团队 382.1.3 招募程序员 402.2 外包程序员 422.2.1 本土化的程序员 432.2.2 程序员，文化及团队 442.3 经验式管理 452.3.1 对待因果关系不严谨 462.3.2 谨慎借用经验 472.3.3 从现在做起 49参考文献 51第3章 从开源做起 523.1 流程和实践 553.1.1 项目的四个P 573.1.2 敏捷的价值 603.1.3 零起点合作 613.2 开源软件开发 623.2.1 软件克隆 633.2.2 软件质量 643.2.3 启动流程 653.2.4 开源开发团体 663.2.5 用户程序员 673.2.6 参与者角色 683.2.7 快速发布 693.2.8 黑盒编程 723.2.9 OSS实践 743.3 类OSS开发 743.3.1 敏捷实践 753.3.2 近邻交流 763.3.3 松耦合和紧耦合 773.3.4 同一地点的软件开发 783.4 结论 79参考文献 80第二部分：韵律第4章 抄袭编程 844.1 抄袭 864.1.1 已有的代码 874.1.2 社交网络分析 884.1.3 被抄袭 894.1.4 让人人成为程序员 924.1.5 模式语言 964.1.6 软件团队能力 984.1.7 粗线条设计 1014.1.8 培训不是解决方案 1024.2 抄袭最快 1034.2.1 不道德 1044.2.2 无先例的代码 1054.2.3 人际关系网 1064.2.4 抄袭的韵律 1074.2.5 工作中抄袭 1104.3 抄袭的生意与韵律 1124.3.1 15分钟的商业报告 1134.3.2 市场调研 1154.3.3 聊天机器人 1174.3.4 老歌新唱 122参考文献 125第5章 结对编程 1275.1 艺术与科学 1285.1.1 最佳搭档 1295.1.2 喧闹的程序设计 1305.1.3 仅仅是培训 1315.1.4 付费给观众 1315.2 两个世界 1325.2.1 没钱的世界 1335.2.2 金钱引导的世界 1355.2.3 经济学 1365.2.4 虚构的质量——时间关系 1365.2.5 加速运行时间 1375.2.6 关键路径法 1385.2.7 为什么是两个结对而不是三个：反组织现象 1415.2.8 软件的需求是个拼图 1425.3 程序设计任务需求 1445.3.1 $2+4=6$ 1445.3.2 $2+4=4$ 1455.3.3 $2+4=3$ 1465.3.4 $2+4 \geq 2$ 1475.3.5 $2+4=?$ 1485.4 结对编程不仅仅是程序设计 1495.4.1 用代码设计 1505.4.2 结对设计 1525.4.3 韵律结对编程 1545.5 结对编程团队指导 156参考文献 158第6章 重复编程 1616.1 结对编程的争议 1646.1.1 编程是一项特殊的工作吗 1646.1.2 三个脑袋是否比两个好 1656.1.3 不可重复的实验 1666.2 重复编程 1676.2.1 相反的结果 1716.2.2 原理 1736.2.3 三人一组编程的效率不高 1746.3 旋律：结对-单独-结对-单独 1766.3.1 持续性 1776.3.2 联系 1796.3.3 动机 1836.4 证明布鲁克斯法则的一个特例 1866.4.1 士气低落 1886.4.2 沟通的成本 1896.4.3 适用于延误项目的旋律 191参考文献 193第7章 敏捷组队 1967.1 项目团队 1977.1.1 自组织团队 2017.1.2 团队中的团队 2027.1.3 项目团队的组成 2047.1.4 团队生命周期与学习曲线 2057.2 生产力 2087.2.1 生产力的错觉 2087.2.2 集体代码所有权 2097.2.3 责任、职责和透明度 2107.3 问题与出问题的人 2117.3.1 旋律：困难——重组 2137.3.2 组队原则 2157.4 拯救即将失败的项目 2177.4.1 项目红绿灯报告 2187.4.2 一个商业案例 2197.4.3 指导委员会会议 2197.4.4 敏捷组队发挥作用 2217.5 提防Iago (埃古) 222参考文献 223第8章 增量设计 2258.1 建模和计划 2268.1.1 敏捷计划 2278.1.2 使用功能性模块进行设计 2308.1.3 简洁设计 2318.1.4 总体成本的概念 2328.2 返工还是复用 2358.2.1 无法避免的返工 2368.2.2 即兴创作 2378.2.3 预先设计 2398.3 即时的软件开发 2408.3.1 CMM的旋律 2418.3.2 一次工厂参观 2448.3.3 走来的工人 2458.3.4 即时软件开发 2478.3.5 增量式设计 2488.4 需求复杂性 2508.4.1 遗漏的需求 2538.4.2 冲突的需求 2548.4.3 迅速改变的需求 2548.4.4 需求和设计 2568.5 重构 2568.5.1 重构活动 2598.5.2 通过挑战进行重构 2608.5.3 为了设计模式进行重构 2638.5.4 故意制造错误 264参考文献 264第9章 测试驱动开发 2679.1 逆向瀑布 2709.1.1 设计-编码-测试 2709.1.2 测试-编码-设计 2719.2 测试优先编程 2729.2.1 测试和验证 2729.2.2 断点测试 2739.2.3 支撑实践 2759.3 韵律：测试-编码-重构 2769.3.1 简单的案例 2789.3.2 自动操作 2799.3.3 意识革命 2819.3.4 用来合作的测试案例 2849.4 快速的软件过程升级 2869.4.1 培训程序 2869.4.2 项目规划 2879.4.3 项目跟踪 2889.4.4 软件质量 2899.4.5 软件配置 2909.4.6 人员纪律 291参考文献 291尾声 各种乐声的混合 293开发旋律和您 294适用于具有更多重复性编程任务的开发旋律 295适用于具有挑战性的任务的开发旋律 295

<<软件开发之韵>>

章节摘录

最终而言，历史和市场可能会消除这些差异，但是在项目实施的当下，对每个项目管理者来说这都是需要解决的问题。

管理者需要了解他们要外包的地区信息，而这些外包地区的人员及文化有着一系列你所不熟悉的、他们本土化的标准。

我们不能因为当今众多事物的全球化，因为有时你不需要跟你的团队成员面对面接触，就想当然地认为团队成员的性格与你无关了。

对于一个良好的团队工作来说，用人得当非常必要。

要是你对发展中国家影响到程序员行为的地区约束缺乏了解，那你就必须把自己的最好水平准备到位，当然这来自于你培训或带领过的各地软件开发团队的经验。

2.2.1 本土化的程序员 在欧洲，制造业已经移到了东欧；在美国，从北部移到了南部，甚至跨过国界移到了墨西哥。

在亚洲，制造业在曾经的农村地区找到了其适应的场所。

制造业一直在这么做——那就是转移到地价和劳动力更便宜的地方，转移到那些当地政府一心想提供资助、打算新建的基础设施上。

同样的道理，像制造业的进发一样，无数小型的当地软件开发团队也以同样的方式出现了，他们或是用自己的团队或是将软件项目转包出去，从而对外提供系统解决方案。

这些软件开发团队往往会受到当地环境的约束。

他们大多数是由当地人组成的，各自为了不同的目的，队伍中也缺乏精英人才。

在这种地区中，一个人如果想组建软件开发团队，除了在大型公司总部及顶级研究型大学之外，都一定要意识到在欠发达城镇和发达城市中团队的业务水平和工作态度往往有着巨大的差别——即使这两个地区的地理位置非常接近。

而当政府着力扶持某个城市或地区时（比如在中国），这个地区与其他地区之间的差异会更加迅速地持续扩大。

<<软件开发之韵>>

编辑推荐

《软件开发之韵：和谐敏捷、珠联璧合的开发》以提高软件质量和效益为目的，针对敏捷开发实践的灵活性，从一种易于接受的、创新的视角来看待软件开发。当敏捷开发在你的组织中运行得并不像你所期望的那样顺利的时候，或是你在敏捷开发和严格开发方式之间犹豫不决的时候，往往就需要停下来思考软件开发的韵律了！

敏捷软件开发是一种十分流行的开发方式，这种开发方式一直在影响着人们对于敏捷实践和传统的严格开发方式之间的联系的认识。

在《软件开发之韵：和谐敏捷、珠联璧合的开发》中，两位作者LUI和CHAN向读者阐述了，在敏捷实践的基础上，如何采用其中一种开发方式并使它与另一种开发方式相结合，以实现一种“协同作用”，这种协同作用也被作者定义为软件开发的韵律。

作者展示了这些开发旋律如何能够互相协同，以实现一种合力作用，从而使它们在协同工作方式下所发挥的效用比单独使用更加强大。

《软件开发之韵：和谐敏捷、珠联璧合的开发》还使用比喻的方式向读者展示了如何解决一些典型的软件管理中的争议性问题，并介绍了如何应对在敏捷软件管理中常见的一些困难。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>