

《软件开发沉思录》

图书基本信息

书名：《软件开发沉思录》

13位ISBN编号：9787115213600

10位ISBN编号：7115213607

出版时间：2009-9

出版社：人民邮电出版社

作者：ThoughtWorks

页数：167

译者：ThoughtWorks中国公司

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu111.com

《软件开发沉思录》

前言

本书面世之际，恰逢“敏捷中国2009大会”召开在即，两者可谓相得益彰。从2004年进入中国，ThoughtWorks见证和参与了中国敏捷社区的发展历程：从五年前的筚路蓝缕，到如今的欣欣向荣。更令人欣慰的是，在原则、价值观等“大问题”上，敏捷的实践者们已经基本达成共识，社区的话题更趋于关注实践——这意味着敏捷社区正在步入成熟，社区成员将用他们的知识和技能为各自效力的企业创造更大的价值。我们在这个时候翻译出版这本文集，是希望为社区的发展再尽绵薄之力。作为敏捷方法的积极推动者，ThoughtWorks从多年、多个行业的实践中积累了丰富的经验。本书收录的13篇文章涵盖了编程技术、项目管理、持续集成、测试等方面内容，将带领读者了解ThoughtWorks在软件生命周期各个环节所推荐的工作方式。比较难得的是，这本文集不仅由ThoughtWorks员工撰写，也由ThoughtWorks员工翻译。译者们或是与文章作者素有私交，或是在文章所论述的领域有所专擅，这也使得翻译质量更有保障。感谢这些译者在工作之余的辛勤劳动，他们是韩锴、胡振波、金明、李剑、乔梁、熊节、徐昊、张晓庆和郑晔。

《软件开发沉思录》

内容概要

从编程技术到项目管理，Roy Singham、Martin Fowler、Rebecca Parsons等来自ThoughtWorks的思想领袖通过本书中的13篇美文，将自己多年沉思和实践所得倾囊相授，引领你走向敏捷软件开发的成功之路。

本书内容丰富，涵盖了软件开发的各个阶段，既包含DSL、SOA、多语言开发和领域驱动设计等热门主题，也有对象设计、一键发布、性能测试和项目管理等方面的经验之谈和独到见解。不论你是开发人员还是项目管理人员，都将从本书中获益匪浅。

《软件开发沉思录》

作者简介

ThoughtWorks公司于1993年在美国成立，现已经发展成为具有千人规模，在6个国家具有分公司的全球性IT咨询公司。公司汇集了许多业界思想领袖和众多高素质人才，致力于为客户解决最棘手最紧迫的问题，业务包括向客户交付定制应用软件、提供注重实效的咨询服务、协助企业进行敏捷开发以及开发软件等。

书籍目录

第1章 引言

第2章 走完业务软件的“最后一英里”

2.1 “最后一英里”问题的源头

2.2 理解问题

2.3 解决“最后一英里”问题

2.4 人

2.5 自动化

2.6 针对自动测试非功能性需求的设计

2.7 将设计与生产环境分离

2.8 无版本软件

第3章 一个巢穴，二十种ruby dsl

3.1 巢穴

3.2 使用全局函数

3.3 使用对象

3.4 使用闭包

3.5 执行上下文

3.6 字面量集合

3.7 动态接收

3.8 总结

第4章 语言的盛景

4.1 简介

4.2 样本

4.3 各种各样的分类

4.4 语言的“生命之树”

4.5 这些都很有趣，但我为什么要关心

第5章 多语言开发

5.1 多语言开发

5.2 用groovy的方式读取文件

5.3 jruby和isblank

5.4 jaskell和函数式编程

5.5 java测试

5.6 多语言开发与未来之路

第6章 对象健身操

6.1 九步迈向优秀软件设计

6.2 练习

6.3 总结

第7章 迭代经理是什么角色

7.1 什么是迭代经理

7.2 怎样成为好的迭代经理

7.3 迭代经理不做什么

7.4 迭代经理与团队

7.5 迭代经理与客户

7.6 迭代经理与迭代

7.7 迭代经理与项目

7.8 总结

第8章 项目生命体征

8.1 项目生命体征

- 8.2 项目生命体征与健康状况
- 8.3 项目生命体征与信息指示器
- 8.4 项目生命体征：项目范围增量图
- 8.5 项目生命体征：交付质量
- 8.6 项目生命体征：预算燃尽
- 8.7 项目生命体征：当前开发状态
- 8.8 项目生命体征：团队感觉
- 第9章 消费者驱动契约：服务演化模式
- 9.1 演化服务：一个例子
- 9.2 schema版本
- 9.3 破坏式的变化
- 9.4 消费者驱动契约
- 第10章 领域标注
- 10.1 当领域驱动设计遇到标注
- 10.2 案例分析：leroy的卡车
- 10.3 总结
- 第11章 重构ant构建文件
- 11.1 简介
- 11.2 ant重构列表
- 11.3 总结
- 11.4 参考文献
- 11.5 资源
- 第12章 一键发布
- 12.1 持续构建
- 12.2 超越持续构建
- 12.3 全生命周期的持续集成
- 12.4 第一道门——提交测试
- 12.5 第二道门——验收测试套件
- 12.6 部署准备阶段
- 12.7 后续的测试阶段
- 12.8 让过程自动化
- 12.9 总结
- 第13章 企业web应用中的敏捷测试和瀑布测试
- 13.1 简介
- 13.2 测试生命周期
- 13.3 测试分类
- 13.4 环境
- 13.5 问题管理
- 13.6 工具
- 13.7 报表与度量
- 13.8 测试角色
- 13.9 参考文献
- 第14章 实用主义的性能测试
- 14.1 什么是性能测试
- 14.2 需求采集
- 14.3 运行测试
- 14.4 沟通
- 14.5 流程
- 14.6 总结

参考书目

第2章 走完业务软件的“最后一英里” 2.4 人 敏捷思潮的一大贡献是对于软件开发作为社会行为的认识：有更好的沟通，才会有更好的软件。在采用敏捷开发实践时，很大部分的努力都是用来打破现有组织架构的藩篱，代之以更高效的模式与实践。然而迄今为止，敏捷实践者们关注的焦点几乎都是软件开发者与使用者之间的沟通。他们之间的沟通改善了需求的质量，使大家对业务目标达成共识，但非功能性需求该怎么办呢？谁对这些需求负责？如何在沟通中体现它们？这些问题往往得不到回答。要避免这种“代码一刀切”的软件开发方式，最简单的方式是找出对非功能的、跨模块的需求负责的人，让他们也参与到软件开发的“社交活动”中来。尽早、尽可能频繁地让他们参与沟通。同样，这也可能需要打破现有组织架构的藩篱，代之以更高效的模式与实践。举例来说，软件编写好之后，系统管理员要负责安装和配置。他们还要监控系统在生产环境下的运行情况，确保操作正确无误。如果有问题出现，他们要按照预先制定的流程来恢复系统状态。他们需要规划系统安装之初和规模扩展之后的硬件需求——内存、磁盘、网络、电源、冷却，凡此种种。维护和支持人员需要系统提供有用的错误报告和有效的诊断流程。他们需要知道如何帮助用户解决简单的系统故障，以及在遇到严重的系统故障时如何上报。

《软件开发沉思录》

媒体关注与评论

“内容非常精彩，本领域的必读之作。” ——DZone技术社区

“在帮助客户实施敏捷的过程中，ThoughtWorks人常被问到一个问题：有没有一套标准的‘敏捷模板’可供快速入门之用？作为一种强调持续改进的方法学，自然不会有一套放诸四海而皆准的‘标准流程’；但对于希望采用敏捷方法的组织和个人而言，若有一组普遍适用的最佳实践作为基础，便能少走许多弯路，以期事半功倍之效。本书正好满足了这一需要。” ——ThoughtWorks中国公司总经理 郭晓

《软件开发沉思录》

编辑推荐

13篇美文荟萃软件开发精华，来自软件界思想领袖们的经验心得，为你开启敏捷开发之门。从编程技术到项目管理，Roy Singham、Martir Fowler、Rebecca Parsons等来自ThoughtWorks的思想领袖通过《软件开发沉思录：Thought Works文集》中的13篇美文，将自己多年沉思和实践所得倾囊相授。引领你走向敏捷软件开发的成功之路。《软件开发沉思录：Thought Works文集》内容丰富。涵盖了软件开发的各个阶段，既包含DSL、SOA、多语言开发和领域驱动设计等热门主题。也有对象设计、一键发布、性能测试和项目管理等方面的经验之谈和独到见解。不论你是开发人员还是项目管理人员，都将从《软件开发沉思录：Thought Works文集》中获益匪浅。

《软件开发沉思录》

精彩短评

精彩书评

- 1、的确是认真看了两遍才起笔的，总想写很多，但是，又都是不成体系的，就吐点槽好了 > ...同时发布在:ZqreadThoughtWorksAnthology - openbookproject - 图灵·样书申请~软件开发沉思录 -Project Hosting on Google Code<http://code.google.com/p/openbookproject/wiki/ZqreadThoughtWorksAnthology>应图灵俱乐部列表的倡议,申请了样书 "软件开发沉思录";想就此进一步整理 敏捷中国2009 中的见闻;= Thought Works 文集 =整体上是本包罗万象的薄书;167页的容量包含了14篇,由 Thought Works 成员在日常工作中总结出来的真实体验,非常不寻常;也非常真诚;由于本人涉及软件开发时间不长,经验也比较狭窄,并没有对所有章节有所共鸣,特此评论其中非常有触动的:== "最后一英里";==这是 Thought Works 创始人的,果然只说问题域,不给答案的;-)这是所有高层在输出文字时的通病,不过,这也正是现实世界的真实一面,"没有银弹";回想俺在过往所有公司项目中的糟糕体验,对比所有自由软件社区项目中的自在体验;也逐渐明白,掺合了利益的项目,都无法轻灵起来的,这是不可能的;但是,我们可以选择自个儿的"态度";正如,这篇短文中最后给出的对策:* 深刻理解了问题本质后* 勇敢的给出自个儿的应对,并坚持下去,坚信可以成功就好,就好 ...== 对象健身操 ==Jeff Bay 的这篇短文,则是标准的技术成员风格:* 给出解决方案* 给出案例操作/自我训练流程* 给出现实应用案例统计完全 的分享即学即用技巧的格局;非常习惯,也非常受用,而且非常认同;这些在使用 OOP 语言时,精练出来的原则,其实在各个公司,各种技术框架上都多少有些;但是,可以总结到这种程度,以及可以内部推广坚持到的地步,实在叹为观止;准备内部也精简结合相关静态代码检测工具,进行推广!-) * PS: 这篇和"重构Ant脚本";那篇都是非常非常地道的技巧分享,只是俺没有什么机会用 Ant 就没有共鸣了>;...== 迭代经理 ==这是 Thought Works 自创的一种角色吧,非常有启发性,这其实是给所有项目中的热心大妈正了名,给了一个"*经理"的名头;* 不过,的确这样将以前多个人自觉配合完成事儿,定一专人统一进行效率要高很多;* 但是,这真的有非常苛刻的先决条件的哪! 1. 团队是敏捷的! 1. 项目是敏捷的! 1. 文化是敏捷的! 否则,迭代经理在开发/技术/项目/产品经理之间,纯粹是个概念>;...== 项目生命体征 ==咳!这篇就是一位成功的迭代经理的经验分享!当然也同样适用其它各种经理;-)给出了5种核心实用体征,以及图表的建立和使用方法;嗯嗯嗯,对俺感触最大的就是"团队感觉";在我的团队中也在部分使用,只是:* 这东西真的要坚持才可用!* 真的要习惯才管用!* 真的要诚实才能用!* 真的要认同才要用!真的是是真的用才可以的...看过太多各种 KIP/PPC 等等对程序员的绩效考核,都考不到点儿上,但是"团队感觉";非常容易作到是真实的,但是,非常难以和真实的绩效结合起来...这也只能是在有足够空闲的情况下的一种参考...== 一键发布 ==果然是位技术主管的文章,这种全流程的自动化,没有高层的支持是绝对作不下来的;每次看各种自动化全流程的方案时,都在流口水,其实我的团队主要是 Python 为主进行开发的,和 Thought Works 的核心开发语言 Ruby 是同类的动态脚本语言,照理,这种没有编译过程的动态脚本系统,应该非常好进行"一键发布";的;但是,真正的作了才发现很难要很大的决心和毅力:1. 先建立配置管理规范 1. 再建立测试主机环境 1. 再确立版本发布策略 1. 然后明确一切都是理解并执行的 1. 最后才是"一键发布";的配置而已...前提条件的各种团队建设的事儿,只要超过3人,就非常容易进入中国的江湖状态,难以理清楚了... * 羡慕从一开始就竖立规范的团队,* 或是有大神力,大法力的技术主管,可以整合所有人的思想和行为,快速打造出这一流程>;...= 小结 =书名翻译的好:* Thought Works 文集* 被翻译为 "软件开发沉思录";* 的确,所有文章没有真实的项目经验支撑的话,是写不出来的* 而且 Wought Works 作为专业的敏捷文化推广者和受益者,可以说一直处在对自我和项目的不断反省中只是,这么小的书这么贵有点难以下决心买...
- 2、覆盖各阶段的论文集，可能不是每一章都适合读者，我比较喜欢关于领域标注的那篇文章，理论实际相结合，Ruby和Ant的基本跳过了，那篇消费者驱动看了但是没有看懂，就仅提供一种思路吧。最后几章比较系统的介绍了各种测试方法，不过这种理论的东西看过估计记住的不多。InfoQ上有下载精简版，挑选文章都是方法论的。 <http://www.infoq.com/cn/minibooks/thoughtworks-anthology>
- 3、从编程技术到项目管理，Roy Singham、Martin Fowler、Rebecca Parsons等来自ThoughtWorks的思想领袖通过本书中的13篇美文，将自己多年沉思和实践所得倾囊相授，引领你走向敏捷软件开发的成功之路。本书内容丰富，涵盖了软件开发的各个阶段，既包含DSL、SOA、多语言开发和领域驱动设计等热门主题，也有对象设计、一键发布、性能测试和项目管理等方面的经验之谈和独到见解。不论你是开发人员还是项目管理人员，都将从本书中获益匪浅。
- 4、这本书我一共读了三遍。不过说实话，我没看出太大的营养，也许是我的水平所限吧。在我看来

，本书不是每一章都适合于读者的，我在读这本书的时候跳过了Ant和一键发布的部分，而关于消费者契约和领域标注，我承认我没有读懂，因此打算等过段时间学了SOA和领域驱动设计的书之后再来看这两章。但是从这本书其他部分来说，我认为很多话都是类似于废话，很多东西明明是软件工程中很基础的理论，还有一些就像<反模式>那本书一样，提到的都是我们大家都能想到的办法。最后，还是那句话，我不质疑，也没有权利，没有能力去质疑这些我所尊敬的大牛们的看法和文章，也许是我水平所限。等到过段时间我再回来读下吧，希望会有新的收获。

5、首先承认我读得比较快。在一个炎热的午后，等着钟点工阿姨帮我把家里打扫干净的4个小时里，我把这本167页的小册子读完了。收获不大。个人认为，在这本小册子里，除了比较大而宽泛的方法论叙述外，稍微有些营养的主要集中在：第7章：迭代经理是什么角色；第8章：项目生命体征；第11章：重构Ant构建文件；第12章：一键发布；第13章：企业Web应用中的敏捷测试和瀑布测试；以及第14章：实用主义的性能测试。6/14，大概是50%不到的有效内容。而我对第5、6章尤其不敢苟同。在第5章中，作者提出可以在统一的JVM平台上利用多语言优势，如JRuby, Groovy等共同开发。殊不知，在公司规模较大，代码拥有数量较多(>1M行)的条件下，降低维护开销的最佳实践之一就是减少使用语言的数量。每引入一种语言，就意味着：1) 公司内要形成一个语言的使用者社区，不然有问题无法共同探讨，而唯一的几个工程师离开后就会形成无人能懂，无人可维护的“死代码”，2) 要建立与之对应的一套构造工具和自动化测试系统，3) 公司的公共架构，比如存储、消息传递、缓存、RPC等，都要为每种语言撰写一套client。这些都是非常大的开销。事实上，对于绝大多数公司而言，有Java/C++，Python，PHP/Ruby，JavaScript/ActionScript，就足够应付日常的工作了。Google的前端是Java做的，Facebook是PHP，豆瓣是Python。这些都是很好的例子。在第6章中，作者非常推崇“极小”，即将类和方法的逻辑和代码行数控制在最小的范围内。这原是一个good practice，无可厚非，因为它可以将代码读者对于上下文和程序逻辑限制在可理解的范围内。然而作者对它的热爱到达了偏执的地步：方法只适用一级缩进、拒绝使用Else关键字、类的长度不超过50行、类中实例变量不超过2个。这样除了把代码变得支离破碎外我看不出有任何的好处。观点可商榷，原教旨主义的实践我着实不敢恭维。瑕不掩瑜，全书中我认为比较出彩的是最后三章，值得一读。最后对于全书定价(¥39)，大笑一声，不予置评。；)

6、这本书非常薄，里面的作者却不少，来自thoughtworks的各个层面，其中心就是围绕敏捷这个东西在说事儿，没有必要从头看到尾，各取所需吧。比如对我来说，只有“对象健身操”这一章勉强是看完了的。

7、全部评论的地址：http://blog.sina.com.cn/s/blog_58f26c070100glhn.html推荐给做软件的朋友们。这是一本技术比较强的书，涉及了许多软件方面的术语和知识，甚至需要读懂程序代码，才能看明白某些章节。建议看此书之前，先对软件项目的敏捷开发有一定的了解。……

《软件开发沉思录》

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:www.tushu111.com