

<<解析极限编程>>

图书基本信息

书名：<<解析极限编程>>

13位ISBN编号：9787111357957

10位ISBN编号：7111357957

出版时间：2011-9

出版时间：机械工业出版社

作者：Kent Beck,Cynthia Andres

译者：雷剑文,李应樵,陈振冲

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>

<<解析极限编程>>

内容概要

极限编程 (xp) 是适用于中小型团队在需求不明确或者迅速变化的情况下进行软件开发的轻量级方法学。

本书是xp宣言, 也是第一本有关xp的图书。

这本书介绍了xp背后的思想——它的根源、哲学、情节等。

它将帮助读者选择是否在项目中使用xp时做出明智的决策。

本书的另一个目的是帮助那些已经在使用

xp的读者更好地理解它。

对程序员而言, xp做出的承诺是他们每天能够处理真正重要的工作, 而不必单独面对令人担忧的状况。

他们将能够集中全力来使他们的系统获得成功。

他们将做出最适合由他们来做的决策。

对于客户和管理人员而言, xp的承诺是他们将从每个编程周期中获得最多的利益。

他们将能够在开发的中途更改项目的方向而不用承担太高的成本。

《解析极限编程--拥抱变化(原书第2版)》适合所有软件开发人员、管理人员参考。

<<解析极限编程>>

作者简介

Kent Beck一向挑战软件工程教条，促进诸如模式、测试驱动开发和极限编程等思想。目前属于Three Rivers Institute和Agitar Software，他是许多Addison-Wesley著作的作者。

Cynthia

Andres拥有心理学学士学位，在组织行为学、决策分析和妇女研究方面工作。从XP诞生以来她就跟Beck一起工作于XP的社会方面。她也属于Three Rivers Institute。

<<解析极限编程>>

书籍目录

本书赞誉

推荐序

译者简介

前言

致谢

第1章 极限编程定义

第一部分 探索极限编程

第2章 学习开车

第3章 价值观、原则和实践

第4章 价值观

沟通

简单

反馈

勇气

尊重

其他

第5章 原则

人性化

经济学

.互惠互利

自相似性

改进

多样性

反省

流

机遇

冗余

失败

质量

婴儿步

接受责任

第6章 实践

第7章 基本实践

坐在一起

完整团队

信息工作空间

充满活力地工作

结对编程

结对与个人空间

故事

周循环

季度循环

松弛

10分钟构建

持续集成

<<解析极限编程>>

测试优先编程

增量设计

那么现在

第8章 启程

为实践绘图

第9章 扩展实践

真实客户参与

增量部署

团队连续性

收缩团队

根源分析

共享代码

代码和测试

单一代码库

每日部署

协商范围的合同

依用付费

第10章 完整xp团队

测试员

交互设计师

架构师

项目经理

产品经理

主管人员

技术文献书写员

用户

程序员

人力资源

角色

第11章 约束理论

第12章 计划：管理范围

第13章 尽早测试、经常测试、自动测试

第14章 设计：时间的价值

简单

第15章 增大xp规模

人数

投资

组织的大小

时间

问题的复杂性

解决方案的复杂性

故障的后果

第16章 访谈

第二部分 xp哲学

第17章 xp诞生的故事

第18章 泰勒主义和软件

第19章 丰田生产制度

<<解析极限编程>>

第20章 应用xp

选择教练

什么时候不应该使用xp

第21章 纯度

认证和鉴定

第22章 离岸开发

第23章 永恒的编程之道

第24章 xp和社区

第25章 结语

参考文献

<<解析极限编程>>

章节摘录

版权页：插图：架构师XP团队中的架构师要查找并进行大规模的重构、编写系统级的架构压力测试，并实现故事。

在项目的整个周期，架构师逐步地应用他们的专业知识到项目中。

他们指导着项目架构的进化。

小系统的架构应该与大系统的架构不一样。

对于小系统，架构师要确保系统有适当小的架构。

随着系统的增长，架构师要确保架构的跟进。

将大的架构变化分解成小而安全的步骤，是XP团队的挑战之一。

权力与责任一致的原则表明，给一个人权力做决定而不必亲自承担后果，其他人还不得不服从他的决定，这样是不好的。

架构师们像其他程序员一样签约参加编程任务，但他们同时要注意会带来巨大利益的大的改动机会。

测试能传达架构的意图。

我曾同一个信用卡处理机的架构师交谈，他说在这样一个高性能要求的环境中，你不希望出现任何可能有障碍的架构。

为了达到这个目的，他的团队有完善的压力测试环境。

当他们想要改进架构时，他们会先改进压力测试直至系统崩溃。

然后他们会改进架构直至刚好符合测试。

我向另外一个公司的架构师建议了这种策略。

他抱怨道，他所有的时间都花费在编写规格说明及向开发者们解释这些说明上，他对没有时间去编码感到非常沮丧。

我建议他写一个测试基础设施(testing in-frastructure)，然后用测试来代替规格说明及其解释。

如果他发现了一个设计漏洞，就应该编写一个用于指出那个漏洞的测试。

虽然我没能说服他尝试这个主意，但是我仍认为它是有价值的。

<<解析极限编程>>

媒体关注与评论

“来自经验的新见识和智慧，更清楚地解析极限编程艺术，本书将会帮助许多人实现杰出软件开发之梦。

” ——Joshua Kerievsky, Industrial Logic公司创始人 “XP改变了我们业界思考软件开发的方式。鲜明简单，聚集于执行，坚持基于事实的计划胜过猜测，这些为软件交付设定了新的标准。

” ——David Trowbridge, 微软公司架构师

编辑推荐

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>