

<<敏捷持续集成>>

图书基本信息

书名：<<敏捷持续集成>>

13位ISBN编号：9787121065576

10位ISBN编号：7121065576

出版时间：2008-6

出版时间：电子工业出版社

作者：罗时飞 著

页数：315

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>

<<敏捷持续集成>>

内容概要

在并重“理论与实践”思想的指引下，作者踏上了持续集成及CruiseControl的写作行程。全书共分为5部分内容：第1部分（初涉持续集成）由持续集成概述（第1章）、实施持续集成（第2章）构成，这部分内容的着眼点在于阐述相关的背景知识及初步认识持续集成；第2部分（敏捷持续集成）由基于Subversion的持续集成实战（第3a章）、基于CVS的持续集成实战（第4章）、基于VSS的持续集成实战（第5章）、基于ClearCase的持续集成实战（第6章）构成，这部分内容将实战摆在了首当其冲的位置；第3部分（持续集成权威指南）由持续数据库集成（第7章）、持续单元及集成测试（第8章）、持续评审（第9章）、持续部署（第10章）、持续功能及负载测试（第11章）、持续反馈（第12章）构成，研究实战背后的理论知识成为了这部分内容的重点；第4部分（CruiseControl高级特性）由分布式集成构建（第13章）、CruiseControl内核研究及扩展（第14章）构成，这是属于CruiseControl自身的盛宴；第5部分由维护CC配置文件（附录A）构成，它是查找CruiseControl配置文件相关知识点的权威去处。

《开源技术专家·敏捷持续集成（CruiseControl版）：高效研发之道》适合作为CI集成人员、Java/JavaEE开发者、架构师、项目及产品开发经理的参考书，同时，《开源技术专家·敏捷持续集成（CruiseControl版）：高效研发之道》非常适合于高校相关专业的学生，以及对持续集成有兴趣的各类读者。

作者简介

罗时飞，独立JavaEE顾问、西安交通大学硕士。
将近10年的JavaEE开发、架构经验，理论和经验并重。
对JavaEE平台、敏捷方法及开源技术有深入研究，出版图书多本，曾被评为电子工业出版社的“优秀译者”。
他一直在依托网站专职从事企业级Java的推广工作，其近期代表作品有：《精通EJB3.0》（第4版，译）、《精通Spring2.0》、《敏捷Acegi、CAS——构建安全的Java系统》。

<<敏捷持续集成>>

书籍目录

第1部分 初涉持续集成第1章 持续集成概述1.1 敏捷软件开发1.1.1 提炼重复任务—实现可回归性的重要前提1.1.2 敏捷性—奔向成功的致胜法宝1.2 何谓持续集成1.2.1 驱动CI的引擎—CI服务器1.2.2 敏捷配置管理1.3 CruiseControl初探1.4 小结第2章 实施持续集成第2部分 敏捷持续集成第3章 基于Subversion的持续集成实战第4章 基于CVS的持续集成实战第5章 基于VSS的持续集成实战第6章 基于ClearCase的持续集成实战第3部分 持续集成权威指南第7章 持续数据库集成第8章 持续单元及集成测试第9章 持续评审第10章 持续部署第11章 持续功能及负载测试第12章 持续反馈第4部分 CruiseControl高级特性第13章 分布式集成构建第14章 CruiseControl内核研究及扩展附录A 维护CC配置文件

章节摘录

第1部分 初涉持续集成 第1章 持续集成概述 1.1 敏捷软件开发 在以往几年的Java EE软件架构、咨询工作当中，作者一直将“可回归性、敏捷性”摆在最为重要的位置，它们在衡量一项工作的完成质量和速度方面起到了至关重要的作用。

诸多开发者都应该有这种经历，自己编写了大量的测试代码，然而每次手工运行它们之前，或者要手工往数据库添加一些测试数据（种子数据，Seed Data），或者将现有RDBMS中的~些测试数据删除，这些已存数据可能是执行上次测试代码期间的“成果”。

与此同时，在每次手工运行完这些测试代码后，有可能会将现有RDBMS中的数据彻底删除或破坏掉等情况发生。

可以看出，这类测试代码不具有可回归性。

由于受到上述多方面因素的干扰，使得测试代码的运行结果具有不确定性。

最终，使得用于完成单元或集成测试目的的测试代码没有真正用起来，进而宣告当前项目的代码测试工作失败。

软件开发期间，不具有可回归性的场景举不胜举。

再比如，作者看到大量的中小型软件研发团队经常是直接基于本地的集成开发环境手工打包WAR，并直接将打好的WAR包部署到生产环境中，Eclipse、NetBeans等IDE工具就能够承担这一“重任”。

这类研发团队是否考虑到：不同开发者的本地配置都会存在这样或那样的差别，比如JDK版本、Eclipse中项目属性设置的不同。

更有甚者，这类WAR包的打包工作没有由专人负责。

一旦日后生产环境反映研发团队提交的WAR包存在问题时，比如缺少某个属性文件，如何追溯并解决这类问题呢?后果不堪设想。

如果要手工完成，至少要在专门的机器上进行吧?或者，最好有一份详细的打包文档，从而使得Web应用的打包工作能够流程化。

可回归性，即尽可能将重复劳动交给计算机去完成，也就是说研发人员（软件从业人员，包括CI集成人员）需要最大化计算机的价值。

自从人类发明计算机以来，它们最擅长的，就是完成重复劳动，而且毫无怨言地工作。

与此同时，我们要善于将软件开发期间涉及到的各种重复劳动提炼出来，并将这类劳动托付给计算机完成，进而使得我们能够尽可能地把宝贵的时间投身于智力劳动上。

这也体现了现阶段计算机与人类间社会分工的差异性。

如果能够改善测试代码的质量，无论数据库中存在的的数据是什么，测试结果都是确定的，这也说明，“运行测试代码”这项工作已经是一种具有重复性质的体力劳动。

这就是一种可回归性的体现，因为测试代码的运行结果不再依赖于某某人、某某先决条件。

类似地，如果借助于Apache Ant或Maven完成Web应用的编译、打包工作，则这也不愧是一种可回归能力的体现。

<<敏捷持续集成>>

媒体关注与评论

持续集成及CruiseControl的权威之作 CI理论与实践并重 如期交付高质量复杂软件产品的
比由之路

<<敏捷持续集成>>

编辑推荐

围绕持续集成及CruiseControl进行讲解。

CI是当前应用最广泛的敏捷实践之一，而CruiseControl是功能最强大、历史最悠久的CI服务器，基于CruiseControl实施CI工作意义不寻常。

全书从阐述可回归性、敏捷性开始，揭开研究CI及CruiseControl的序幕，这便是“初涉持续集成”部分；“敏捷持续集成”部分专注于CI实战，带领读者深入Subversion、CVS、VSS、ClearCase等主流SCM工具的持续集成实践中；研究CI实战背后的理论知识成为“持续集成权威指南”部分的重点，由持续数据库集成、持续单元及集成测试、持续评审、持续部署、持续功能及负载测试、持续反馈构成；“CruiseControl高级特性”部分是属于CruiseControl自身的盛宴，由分布式集成构建、CruiseControl内核研究及扩展构成；附录A深入而细致地阐述了全家福。

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>