

Join the discussion! p2p.wrox.com



Wrox Programmer to Programmer™

Real-World Solutions for Developing High-Quality
PHP Frameworks and Applications

开发高质量PHP框架 与应用的实际案例解析

[美] Sebastian Bergmann 著
Stefan Pribsch 译
刘文瀚 刘海燕

清华大学出版社

开发高质量PHP框架与 应用的实际案例解析

[美] Sebastian Bergmann 著
Stefan Pribsch
刘文瀚 刘海燕 译

清华大学出版社

北 京

Sebastian Bergmann, Stefan Pribsch

Real-World Solutions for Developing High-Quality PHP Frameworks and Applications

EISBN: 978-0-470-87249-9

Copyright © 2011 by Wiley Publishing, Inc.

All Rights Reserved. This translation published under license.

本书中文简体字版由 Wiley Publishing, Inc. 授权清华大学出版社出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

北京市版权局著作权合同登记号 图字: 01-2011-3826

本书封面贴有 Wiley 公司防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

开发高质量 PHP 框架与应用的实际案例解析/(美)伯格曼(Bergmann, S.), (美)普瑞斯克(Pribsch, S.)著;
刘文瀚, 刘海燕 译.—北京: 清华大学出版社, 2012.5

书名原文: Real-World Solutions for Developing High-Quality PHP Frameworks and Applications
ISBN 978-7-302-28526-7

I. ①开… II. ①伯… ②普… ③刘… ④刘… III. ①PHP 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2012)第 066237 号

责任编辑: 王 军 于 平

装帧设计: 牛艳敏

责任校对: 邱晓玉

责任印制: 王静怡

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 邮 编: 100084

社 总 机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者: 清华大学印刷厂

装 订 者: 三河市溧源装订厂

经 销: 全国新华书店

开 本: 185mm×260mm

印 张: 20.75

字 数: 531 千字

版 次: 2012 年 5 月第 1 版

印 次: 2012 年 5 月第 1 次印刷

印 数: 1~4000

定 价: 49.00 元

产品编号: 041152-01

作者简介

Sebastian Bergmann拥有计算机科学学位，是PHP项目中质量保证领域的先驱者。他的测试框架PHPUnit已经成为约定俗成的标准。他积极活跃在PHP开发领域并创建了多种开发工具。Sebastian Bergmann是一位在国际上深受欢迎的专家。他具有写作书籍和文章的经验，并且时常在一些国际会议上发言。

Stefan Pribsch参与创办thePHP.cc公司，并且是该公司的首席顾问。他拥有计算机科学学位，出版了多部书籍，发表过多篇科技论文。作为顾问，他帮助客户改进开发过程，使PHP能更好地发挥作用。他专注于软件体系结构、OOP(Object Oriented Programming，面向对象的程序设计)、设计模式及工具。Stefan Pribsch经常在国际IT会议上发言。

序 言

开发高质量的软件并不是一个新概念，几乎所有人都承认它的重要性。多年来，我有幸开发了与人们生活密切相关的关键任务操作软件。期间，我学到了如何从项目初期到最后关键任务使用中实现和驱动质量过程。开发高质量软件的过程并非微不足道，它需要团队领导层的支持和参与，可以影响人才、系统、过程、成员的沟通甚至组织结构的甄选。

就我所知，Internet的广泛应用及快速发展所带来的挑战使我所开发的那些关键任务的操作系统相形见绌。现在许多新系统只是关键业务型，事实上，其他复杂层面同样是重要的，如更强分布式的开发团队、众所周知且不断变化的针对Web标准及软件的安全攻击、国际化挑战、SaaS(软件即服务环境)下要求更短的发布周期等。此外，因为在电子商务应用方面因故障而停机即意味着损失财产，所以高质量的程序在该方面尤为重要，且对兼容性、安装时间(检验和调整安装时间)及端对端交易的实时运行能力方面有更高要求，以确保应用程序运行的同时交易也在进行。另外，对用户体验的日益重视也意味着关注质量将变得更加重要，而不能提供理想的用户体验的运行系统则必须在短期内不折不扣地改进自身质量。质量过程和系统都必须极好地支持外部的急剧变化。

毋庸置疑，与过去开发任务关键软件的已有经验相比，这些挑战促进了引起质量保证方法的重要创新和变化的发生。在过去的几年里，软件开发在营造质量保证的意识氛围及最佳实践中取得了巨大的成就。开发人员必须是质量过程的重要组成部分且不能将责任仅仅推给质量团队，这些共识是质量保证的关键进展。持续集成方法缓和了推出高质量软件的集成阶段的最大挑战。对自动测试更具战略性眼光的关注使得安装变得更快，不仅兑现SLA(Service Level Agreement，服务水平协议)甚至比其规定的做得更到位，以满足终端用户的需求。

本书重点强调许多质量保证的规定，且从实用角度出发关注 PHP，全书涉及人员分工、系统、过程和开发工具。本书作者集该方向的实践与理论知识于一身，我认为作者是写作该类书籍的最佳人选。他们不仅有丰富的实践经验，而且在PHP生态系统内开发出了质量工具或对其有所贡献。此外，作为向他人学习如何最好地解决所遇到的问题的一种途径，本书对案例研究的关注也是非常宝贵的。

——安迪·古特曼斯

前言

实践是最残忍的老师，但实践出真知。

——C.S.LEWIS

关于此书

根据TIOBE编程语言排行榜，PHP是继C/C++和Java之后最受欢迎的编程语言。Gartner预言动态编程语言对许多下一代应用程序开发计划的成功至关重要，他认为PHP是该类编程语言最有力的代表。PHP的设计初衷就是服务于Web应用程序开发，并且可能是2000年初推动互联网蓬勃发展的力量之一。从那时起，PHP逐渐发展成熟成为一种过程化和面向对象程序的、具有多种用途的编程语言。在PHP社区中，关于性能(Performance)、可扩展性(Scalability)及安全性(Security)的课题曾经盛极一时。但是近几年来，体系结构和质量引起了更多的关注。据我们的咨询经验，我们看到越来越多的企业将开发过程建立在敏捷的基础上，且希望使他们的基于PHP的软件更加现代化。基本代码的现代化通常由PHP 4转向PHP 5驱动或者通过引入标准开发框架实现。

在此背景下，PHP框架存在过剩现象就不足为奇了。所有这些框架都是为了帮助解决循环使用案例及应用程序开发的标准化。动态测试和静态测试技巧及自动构造和持续集成对于PHP开发人员不再是一个陌生的概念。尤其在企业关键应用程序中，简单的PHP编程已经发展成采用PHP的软件工程。

这是一本关于PHP的书吗

本书以PHP案例为基础讲述如何计划、执行不同软件层的自动测试、用软件度量和持续集成测量软件质量。我们假定读者或者是有经验的PHP开发人员并且对PHP项目的质量保证感兴趣，或者是可以熟练运用另一种编程语言实现案例的开发人员。

本书并不想成为如何用PHP 5编程的初级读物。虽然许多公司在由PHP 4转向PHP 5的过程中第一次考虑到质量保证测量，但本书并不包括关于PHP应用程序及环境迁移的相关主题。

除了开发人员，项目领导者及质量保证工程师都应该参与到质量保证中来。我们希望本书能在软件项目的所有参与人员之间建立相互信任的关系，并鼓励所有读者改进软件的内部质量(参见第1.2节)。

本书结构

实践出真知，我们尤其可以从他人的实践中学到许多——本书集各种案例于一体，通过这些案例我们可以看到知名企业和项目的幕后情况并了解到宝贵的实践经验。

第 I 部分介绍了如何定义和理解软件质量、如何测试各种软件层。第 II 部分给出了那些经过实践证明的好的方法和对策(如关于单元测试的编写)及Digg Inc.和TYPO3项目的开发人员是如何实现的。第4章从另一角度叙述了同一主题，并给出编写单元测试时可以避免的、易犯的错误。

第III部分讨论了对面向服务API(Application Program Interface, 应用程序接口)及服务器组件的测试。第IV部分以Symfony为例，展示了如何测试框架自身和由该框架创建的应用程序。利用eZComponets中的Graph组件，我们讨论了一个优秀的松散耦合项目的体系结构如何实现二进制映像数据的测试。测试数据库的交互作用会影响应用程序体系结构的多个层次，这非常重要，所以独立成章。

在第V部分中，studiVZ开发人员及swoodoo报告了他们在大型项目和团队的质量保证经验。第12章将动态测试技巧与静态测试技巧相结合，展示了不同的质量保证工具是如何被高效地联合使用的。

最后一部分作为结尾，讲述了软件的易用性、性能和安全性。

关于案例研究的作者

Robert Lemke和Karsten Dambekalns(TYPO3协会)

Robert Lemke是TYPO3协会的联合创始人，引领TYPO3 v5(代码名Phoenix)和FLOW3的开发。他热衷于敏捷开发方法与整洁代码。

Karsten Dambekalns，通过研究其他网站的HTML源代码，学习了来之不易的网络技术基础。在使用了OS/2、Windows和Linux之后，他现在使用Mac。他在自己的Commodore C128上学习了BASIC和汇编语言。他从1999年开始使用PHP，于2002年了解了TYPO3并被其巨大潜力所吸引。现在他是TYPO3 Phoenix和FLOW3核心开发小组的一员，并且是TYPO3协会指导委员会的委员。

在他们所写的第3章中，Robert Lemke和Karsten Dambekalns以可持续的方式给出TYPO3项目改进其软件质量的基础和采用的技术。

Benjamin Eberlei(direct:effect GmbH)

Benjamin Eberlei是直接:effect GmbH的软件开发人员。闲暇时间，作为Doctrine团队的一员，他维护Zend框架的部分组件，并促成了其他几个小开源项目。

在他写的第4章中，为了使软件的自动测试投资回报最大化，他介绍了在编写单元测试时可以避免的常犯错误。

Matthew Weier O' Phinney(ZendTechnologies Ltd.)

Matthew Weier O' Phinney是Zend框架的项目领导，在最初的预览版发布之前，他一直致力于该项目。他是开源软件的支持者，就PHP最佳实践主题作出了阐释。

在Matthew Weier O' Phinney撰写的第6章中，他强调网络服务测试的挑战并介绍了在Zend框架项目中被证明非常成功的解决方案。

Tobias Schlitt

Tobias Schlitt拥有计算机科学学位，利用PHP从事专业网络项目长达十余年。作为开源的忠实爱好者，他经常对各种社区计划作贡献。Tobias是Qafoo GmbH的联合创始人。Qafoo GmbH为开发高质量的PHP提供服务，服务项目包括为质量保证和更好的编程提供咨询与培训，并且为几个PHP QA(Quality Assurance, 质量保证)工具提供技术支持。

在他撰写的第7章中，指出在编写自动测试时需要借助非常规方法来实现目标。

Fabien Potencier(Sensio Labs)

Fabien Potencier于1994年接触到了Web，当时连接Internet需要借助制造有害、刺耳的噪音的调制解调器。作为一名酷爱该行业的开发人员，他立即使用Perl语言创建网站。但是随着PHP 5的发布，他决定将注意力转移到PHP，并于2004年创建Symfony框架项目以帮助其公司改变PHP对客户的影响。Fabien是一名多产的企业家，他于1998年创建了Sensio公司，该公司是专门从事网络技术和Internet市场营销的服务与咨询公司。Fabien也是其他几个开源项目的创建者，他同时还在写作、写博客，并经常在国际会议上发言。

在他撰写的第8章中，介绍了他从Symfony项目中得到的经验，并展示了Symfony测试如何改进框架的编程接口。

Kore Nordmann

Kore Nordmann拥有计算机科学学位。他是eZ Systems(企业开源CMS的龙头生产商)的开发人员和软件体系结构师。除此之外，他正在开发、维护不同的开源项目，如eZComponents、Arbit、wcv、Image3D、PHPUnit等。Kore还曾是美国国内、国际会议的正式发言人，他还出版了几本书并发表过不少文章。作为Qafoo GmbH的一员，他也提供咨询服务。

在他撰写的第9章中，描述了一个精心设计的体系结构如何通过使用伪对象测试产生二进制输出的组件。

Michael Lively Jr.(Selling Source LLC)

Michael Lively Jr.从2001年开始使用PHP，从2005年开始加入到PHP测试社区。他是PHPUnit数据库扩展测试的创建者，并对PHPUnit做出过其他一些贡献。 he现在是Selling Source LLC基于Las Vegas的程序体系结构师。在Selling Source LLC的工作期间，他还从事过几个项目，其中包括企业级贷款管理、利用PHP语言开发出为成千上万客户和几百个代理商提供服务的处理平台。

在他撰写的第10章中，描述了DbUnit的功能、PHPUnit关于数据库测试的扩展，并展示了这个强大的工具如何提高系统效率。

Christiane Philipps(Rebate Networks GmbH)和Max Horváth(Vodafone GmbH)

Christiane Philipps是Rebate Networks的CTO(Chief Technology Officer, 首席技术官), 并且热衷于Rebate Networks敏捷开发。她经常撰写敏捷测试方面的文章。

Max Horváth是Vodafone Internet Services的软件总工程师, 有着十余年的网络开发经验。在撰写本书时, 他是VZnet Netzwerke的移动开发小组的成员。

在他们撰写的第11章中, Christiane Philipps和Max Horváth讲解了他们如何成功地为欧洲最大的社交网络平台之一引入PHPUnit和Selenium RC。

Manuel Pichler(OnVista Media GmbH)与Sebastian Nohn(Ligatus GmbH)

Manuel Pichler创建了PHP质量保证工具, 如PHP Depend、PHPMD和phpUnderControl。他是Qafoo GmbH的联合创始人, 该公司为开发高质量的PHP 提供服务。

Sebastian Nohn从1996年开始开发动态网站, 并于2002年起负责商业和开源项目的质量保证。他是首次在PHP项目的持续集成中使用CruiseControl的人之一。

在他们撰写的第12章中, 介绍了如何持续集成、如何编写单元测试、软件度量及其他静态测试技巧, 解释持续集成如何帮助改进旧有应用程序的软件质量。

Lars Jankowsky(swoodoo AG)

Lars Jankowsky是swoodoo AG的CTO, 负责提供基于PHP的航班和饭店价格比较服务。他已经有超过15年的开发网络应用程序的经验, 从PHP的早期版本开始他一直使用PHP。他的热情还投注在领导eXtreme编程小组上。

在他撰写的第13章中, 介绍了swoodoo如何引进敏捷方法和服务型体系结构使得他们的产品稳定、持续发展。

Jens Grochtdreis

Jens Grochtdreis是自由网络开发人员, 而且是一名专于前端开发和可访问性的顾问。

在他撰写的第14章中, 展示了如何开发易于理解和使用的网站以及如何测试易用性。

Brian Shire

Brian Shire 大约在8岁时开始在一台Apple IIe上编程。当不玩游戏时, 他就学习Basic编程语言。撰写此书时, Brian在Facebook,Inc.工作, 致力于PHP基础设施的缩放。他在Facebook工作的4年里, 网站的用户已经从500万增加到15.7亿。在此期间, Brian成为APC(一种操作码, 是PHP的用户变量的高速缓冲存储器)的主要贡献者。他对PHP本身及其他PECL扩展都有贡献。Brian曾在世界各地不同的会议上分享他的经验和阅历。

在他撰写的第15章中, Brian Shire为网络应用程序的性能测试提供了动力, 并向读者介绍了合适的性能测试的工具和过程。

Arne Blankerts(thePHP.cc)

Arne Blankerts作为IT的领导有着长期的开发经验。他的软件fCMS创新地使用了XML技术

且对国际公司的业务关键应用程序至关重要。他积极参与PHP的程序说明书的编写工作。Arne Blankerts是IT安全方面的专家，并且在一家杂志上撰写相关内容的专栏。他是国际会议上广受欢迎的发言者，撰写过一本书并在IT杂志上发表过多篇文章。

在他撰写的第16章中，介绍了如果知道常见的攻击途径并遵守一些重要原则，那么开发安全的应用程序将是一件轻而易举的事。

源代码

读者在学习本书中的示例时，既可以手动输入所有的代码，也可以使用本书附带的源代码文件。本书使用的所有源代码都可以从本书合作站点<http://www.wrox.com/>或www.tupwk.com.cn/downpage上下载。只要登录站点<http://www.wrox.com/>，使用Search工具或使用书名列表就可以找到本书。接着单击本书细目页面上的Download Code链接，就可以获得所有源代码。



由于许多图书的书名都很类似，所以按ISBN进行搜索是最简单的，本书英文版的ISBN是978-0-470-87249-9。

勘误表

尽管我们已经尽了最大的努力来保证文章或代码中不出现错误，但是错误总是难免的，如果您在本书中找到了错误，例如拼写错误或代码错误，请告诉我们，我们将非常感激。通过勘误表，可以让其他读者避免走入误区，当然，这还有助于提供更高质量的信息。

要在网站上找到本书英文版的勘误表，可以登录<http://www.wrox.com>，通过Search工具或书名列表查找本书，然后在本书的细目页面上，单击Book Errata链接。在这个页面上可以查看到Wrox编辑已提交和粘贴的所有勘误项。完整的图书列表还包括每本书的勘误表，网址是www.wrox.com/misc-pages/booklist.shtml。

如果你在勘误表上没有找到错误，那么可以到www.wrox.com/contact/techsupport.shtml上完成上面的表格，并把找到的错误发送给我们。我们将会核查这些信息，如果无误的话，会把它放置到本书的勘误表中，并在本书的后续版本中更正这些问题。

p2p.wrox.com

要与作者和同行讨论，请加入p2p.wrox.com上的P2P论坛。这个论坛是一个基于Web的系统，便于您张贴与Wrox图书相关的消息和相关技术，与其他读者和技术用户交流心得。该论坛提供了订阅功能，当论坛上有新的消息时，它可以给您传送感兴趣的论题。Wrox作者、编辑和其他业界专家和读者都会到这个论坛上来探讨问题。

在<http://p2p.wrox.com>上，有许多不同的论坛，它们不仅有助于阅读本书，还有助于开发自己的应用程序。要加入论坛，可以遵循下面的步骤：

- (1) 进入p2p.wrox.com，单击Register链接。
- (2) 阅读使用协议，并单击Agree按钮。
- (3) 填写加入该论坛所需要的信息和自己希望提供的其他信息，并单击Submit按钮。
- (4) 你会收到一封电子邮件，其中的信息描述了如何验证账户和完成加入过程。



不加入P2P也可以阅读论坛上的消息，但要张贴自己的消息，就必须加入该论坛。

加入论坛后，就可以张贴新消息，回复其他用户张贴的消息。可以随时在Web上阅读消息。如果要让该网站给自己发送特定论坛中的消息，可以单击论坛列表中该论坛名旁边的Subscribe to this Forum图标。

关于使用Wrox P2P的更多信息，可阅读P2P FAQ，了解论坛软件的工作情况以及P2P和Wrox图书的许多常见问题。要阅读FAQ，可以在任意P2P页面上单击FAQ链接。

目 录

第I部分 基 础

第1章 软件质量	3
1.1 外部质量	4
1.2 内部质量	4
1.3 技术债务	5
1.4 结构性质量保证	6
1.5 整洁代码	7
1.5.1 显式、最小的依赖	7
1.5.2 责任明确	8
1.5.3 不可复制	8
1.5.4 尽可能少的执行分支、尽可能简短的方法	8
1.6 软件度量	8
1.6.1 圈复杂度与npath复杂度	9
1.6.2 CRAP	9
1.6.3 non-mockable total recursive cyclomatic complexity	9
1.6.4 全局可变状态	10
1.6.5 聚合与耦合	10
1.7 工具	10
1.7.1 PHPUnit	10
1.7.2 phploc	10
1.7.3 PHP Copy-Paste-Detector	10
1.7.4 PHP Dead Code Detector	11
1.7.5 PHP_Depend	11
1.7.6 PHP Mess Detector	11
1.7.7 PHP_CodeSniffer	11
1.7.8 bytestkit-cli	11
1.7.9 PHP_CodeBrowser	11
1.7.10 CruiseControl与 PHPUnitControl	11
1.7.11 Hudson	12
1.7.12 Arbit	12
1.8 本章小结	12

第2章 软件测试	13
2.1 黑盒测试与白盒测试	13
2.2 需要多少次测试	13
2.3 系统测试	14
2.3.1 浏览器测试	15
2.3.2 自动测试	16
2.3.3 孤立测试	16
2.3.4 验收测试	17
2.3.5 系统测试的局限	17
2.4 单元测试	18
2.4.1 返回值	19
2.4.2 依赖性	21
2.4.3 副作用	21
2.5 案例探究	21
2.5.1 分析待测试代码	24
2.5.2 创建测试环境	25
2.5.3 避免全局依赖性	27
2.5.4 独立于数据源的测试	28
2.5.5 异步事件测试	33
2.5.6 数据库变化的存储	36
2.5.7 意外的结果	37
2.5.8 封装输入数据	39
2.5.9 问题的进一步研究	40
2.6 本章小结	41

第II部分 最佳实践

第3章 TYPO3: 呆板项目之敏捷未来	45
3.1 简介	45
3.1.1 TYPO3的历史: 13个春秋	45
3.1.2 敢于重新开始	46
3.1.3 我们的测试经验	47
3.2 方法和技术	47
3.2.1 苦甜参半的“大块头”	48

3.2.2	测试驱动开发	48
3.2.3	文档式测试	49
3.2.4	持续集成	50
3.2.5	整洁代码	51
3.2.6	代码重构	52
3.2.7	编程向导	52
3.2.8	领域驱动设计	53
3.3	开发过程中的行动方向	54
3.3.1	开发新代码	54
3.3.2	扩展和调试代码	54
3.3.3	优化代码	55
3.3.4	寻找并修复漏洞	56
3.3.5	处理旧代码	56
3.4	测试处方	56
3.4.1	无意的功能性单元测试	57
3.4.2	文件系统入口	57
3.4.3	接口处的构造函数	58
3.4.4	测试抽象类	58
3.4.5	测试受保护方法	59
3.4.6	回调函数的使用	61
3.5	迈向未来	62
第4章	单元测试的糟糕实践	63
4.1	为什么测试质量很重要	63
4.2	糟糕实践和测试味道	64
4.2.1	测试代码重复	64
4.2.2	断言复杂和测试试图	65
4.2.3	脆弱测试	67
4.2.4	模糊测试	69
4.2.5	谎言测试	73
4.2.6	慢测试	74
4.2.7	测试中的条件逻辑	75
4.2.8	自行验证测试	76
4.2.9	Web冲浪测试	77
4.2.10	模拟过度杀伤	78
4.2.11	跳过蔓延	79
4.3	本章小结	79
第5章	Digg公司的质量保证	81
5.1	面临的问题	81

5.1.1	遗留基本代码	82
5.1.2	如何解决这些问题	83
5.1.3	选择测试框架	84
5.1.4	引进专家	84
5.1.5	会议室内的一周	84
5.2	训练团队	84
5.3	编写可测试代码	87
5.3.1	规避静态方法	87
5.3.2	依赖注入	89
5.4	模拟对象	89
5.4.1	综述	89
5.4.2	数据库	89
5.4.3	松散耦合依赖	90
5.4.4	测试类内部构件的试验对象- 观察者	91
5.4.5	分布式缓存	92
5.4.6	模拟服务型架构	92
5.5	Digg的质量保证过程	96
5.5.1	测试	96
5.5.2	益处	97
5.6	本章小结	98

第III部分 服务器与服务

第6章	服务型API测试	101
6.1	问题	102
6.2	解决方案	103
6.2.1	API证书	103
6.2.2	API限制	106
6.2.3	服务协议的脱机测试	107
6.2.4	具体服务的脱机测试	110
6.3	本章小结	115
第7章	测试WebDAV服务器	117
7.1	关于eZ WebDAV组件	117
7.1.1	WebDAV	117
7.1.2	体系结构	119
7.2	开发过程中的挑战	120
7.2.1	需求分析	120
7.2.2	RFC之后的TDD	121

7.2.3 测试一个服务器	121	9.4.5 位图对比	161
7.3 使用PHPUnit进行自动验收测试	123	9.4.6 Flash	162
7.3.1 捕获追踪测试	124	9.5 本章小结	164
7.3.2 测试方法	125	第10章 测试数据库交互	165
7.3.3 集成到PHPUnit	126	10.1 概述	165
7.3.4 定制测试案例	126	10.2 不编写数据库测试的原因	166
7.3.5 验收测试套件	129	10.3 编写数据库测试的原因	166
7.3.6 验收测试示例	130	10.4 测试的内容	167
7.4 本章小结	132	10.5 编写测试: 模拟数据库连接	168
第IV部分 体系结构		10.6 编写测试: PHPUnit数据库扩展	168
第8章 Symfony测试与Symfony项目	135	10.6.1 数据库测试案例类	169
8.1 测试框架	135	10.6.2 建立数据库测试连接	170
8.1.1 Symfony发布管理过程	136	10.6.3 创建数据集	172
8.1.2 测试与真实代码	137	10.6.4 数据操作	185
8.1.3 运行测试套件	137	10.6.5 建立测试	187
8.1.4 主要教训	137	10.6.6 使用数据库测试器	193
8.2 测试Web应用程序	141	10.7 应用测试驱动的设计进行数据库测试	195
8.2.1 减少进入测试的障碍	142	10.8 使用数据库测试进行回归测试	195
8.2.2 单元测试	142	10.8.1 数据的测试问题	196
8.2.3 功能测试	146	10.8.2 数据揭示的测试问题	197
8.3 本章小结	150	10.9 本章小结	197
第9章 测试ezcGraph组件	151	第V部分 整体问题与解决方案	
9.1 开发理念	152	第11章 studiVZ的质量保证	201
9.2 图形组件	152	11.1 简介	201
9.2.1 体系结构	153	11.2 验收测试	202
9.2.2 测试需求	154	11.3 Selenium	203
9.3 驱动程序的模拟化	154	11.4 studiVZ的技术设置	205
9.3.1 模拟驱动程序	154	11.4.1 开发环境	205
9.3.2 多重断言	156	11.4.2 测试环境	206
9.3.3 结构	157	11.5 最佳做法	206
9.3.4 预处理	157	11.5.1 早期的错误	207
9.3.5 小结	158	11.5.2 策略变更	208
9.4 测试二进制数据	158	11.6 我们需要一个DSL	215
9.4.1 驱动程序	158	11.6.1 内部DSL	216
9.4.2 预处理	159	11.6.2 Testing_SeleniumDSL 1.0 版本	216
9.4.3 SVG	159		
9.4.4 创建位图	160		

11.6.3	Testing_SeleniumDSL 2.0 版本——一个草案.....	218
11.6.4	Testing_SeleniumDSL 2.0 版本的现状和展望.....	218
11.7	本章小结	219
第12章	持续集成	221
12.1	简介	221
12.1.1	持续集成.....	222
12.1.2	静态分析.....	223
12.2	安装.....	232
12.3	配置.....	233
12.3.1	静态测试.....	236
12.3.2	动态测试.....	240
12.3.3	报告.....	241
12.3.4	成果.....	242
12.4	操作.....	244
12.5	高级话题.....	244
12.5.1	持续部署.....	244
12.5.2	使用反向代理.....	245
12.5.3	持续集成和敏捷范式.....	246
12.6	本章小结	247
第13章	swoodoo: 真正的敏捷史	249
13.1	简介	249
13.2	强者生存	249
13.3	现在正在做的工作	255
13.4	演化的艺术.....	258
13.4.1	经验不足.....	259
13.4.2	Java开发人员使用PHP编码 的现象.....	260
13.4.3	只有自己理解其代码的 开发人员.....	262
13.5	本章小结	263
第Ⅵ部分 非功能性方面		
第14章	易用性.....	267
14.1	开销和工作的平衡.....	268
14.2	设计.....	269

14.2.1	可访问性	269
14.2.2	可读性	270
14.2.3	表单元素标签	270
14.2.4	键盘导航	270
14.2.5	有效的对比度	271
14.2.6	返回首页的连接标志	271
14.2.7	图像的替代文本	272
14.2.8	背景图像的背景颜色	272
14.2.9	可用的打印版本	272
14.2.10	可视的链接	272
14.2.11	良好的书签	272
14.2.12	不使用帧	272
14.2.13	可缩放的字體	273
14.3	技术	273
14.3.1	性能	273
14.3.2	JavaScript	274
14.4	用户指南	275
14.4.1	“折叠”的神话	275
14.4.2	互动的反馈	276
14.4.3	导航	276
14.4.4	弹出窗口和其他的烦恼	277
14.4.5	习惯和期望	277
14.4.6	容错和反馈	277
14.5	测试易用性	277
14.6	本章小结	278
第15章	性能测试	279
15.1	简介	279
15.1.1	工具	280
15.1.2	环境的注意事项	280
15.2	负载测试	281
15.2.1	Apache Bench	282
15.2.2	Pylot	283
15.2.3	其他负载测试工具	285
15.3	剖析	285
15.3.1	Callgrind	286
15.3.2	KCachegrind	289
15.3.3	APD	290
15.3.4	Xdebug	291
15.3.5	XHPProf	292

15.3.6	OProfile	293	16.2.3	软件开发	303
15.4	系统指标	294	16.3	安全性的开销	305
15.4.1	strace	294	16.4	最常见的问题	306
15.4.2	Sysstat	295	16.4.1	未经验证的重定向和 转发	306
15.4.3	自定义工具	297	16.4.2	传输层保护不足	307
15.5	常见的陷阱	297	16.4.3	限制URL访问失败	307
15.5.1	开发与生产环境	297	16.4.4	不安全的加密存储	308
15.5.2	CPU时间	298	16.4.5	安全性错误配置	308
15.5.3	微优化	298	16.4.6	跨站点请求伪造	309
15.5.4	将PHP作为粘合剂	298	16.4.7	不安全的直接对象引用	309
15.5.5	优化的优先级	299	16.4.8	破坏的认证和会话管理	310
15.6	本章小结	299	16.4.9	跨站点脚本	310
第16章	安全性	301	16.4.10	注入	311
16.1	安全性概述	301	16.5	本章小结	312
16.2	设计安全	302	第17章	结束语	313
16.2.1	操作	302			
16.2.2	物理访问	303			

第I部分 基 础

- ▶ 第1章 软件质量
- ▶ 第2章 软件测试

