

### 图书基本信息

书名：<<开发高质量PHP框架与应用的实际案例解析>>

13位ISBN编号：9787302285267

10位ISBN编号：7302285268

出版时间：2012-5

出版时间：清华大学出版社

作者：Sebastian Bergmann

页数：314

字数：531000

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>

## 内容概要

PHP已经成为最受欢迎的编程语言之一，这使得用PHP创建高质量、易维护的应用程序和框架比以往受到更多的青睐。

通过使用来自于知名公司的真实案例研究，《开发高质量PHP框架与应用的实际案例解析》为Web软件体系结构的不同层次介绍了规划、执行以及测试自动化方面的内容，并解释了这些公司如何测量和测试软件质量。

《开发高质量PHP框架与应用的实际案例解析》作者提供了解决问题的不同观点，因此您可以在开发自己的PHP应用程序和框架的过程中从不同的方法中获益良多。

## 作者简介

伯格曼, Sebastian Bergmann, 是PHP Consulting Company的创始人之一。他是软件质量保证领域的倡导者和领导者, 并且是PHP Unit和很多其他工具的创建者。他同时也在很多全球性会议上发言。

普瑞斯克, Stefan Priebsch, 是PHP Consulting Company的创始人之一, 他帮助客户改进开发过程并更好地使用PHP。他出版过多部书籍并经常在IT会议上发言。

## 书籍目录

### 第1部分 基础

#### 第1章 软件质量

##### 1.1 外部质量

##### 1.2 内部质量

##### 1.3 技术债务

##### 1.4 结构性质量保证

##### 1.5 整洁代码

##### 1.5.1 显式、最小的依赖

##### 1.5.2 责任明确

##### 1.5.3 不可复制

##### 1.5.4 尽可能少的执行分支、尽可能简短的方法

##### 1.6 软件度量

##### 1.6.1 圈复杂度与npath复杂度

##### 1.6.2 CRAP

##### 1.6.3 non-mockable total recursive cyclomatic complexity

##### 1.6.4 全局可变状态

##### 1.6.5 聚合与耦合

##### 1.7 工具

##### 1.7.1 PHP Unit

##### 1.7.2 phpLOC

##### 1.7.3 PHP Copy-Paste-Detector

##### 1.7.4 PHP Dead Code Detector

##### 1.7.5 PHP Depend

##### 1.7.6 PHP Mess Detector

##### 1.7.7 PHP Code Sniffer

##### 1.7.8 bytekite-cli

##### 1.7.9 PHP-Code Browser

##### 1.7.10 Cruise Control与PHP Under Control

##### 1.7.11 HUd SOn

##### 1.7.12 Arbic

##### 1.8 本章小结

#### 第2章 软件测试

##### 2.1 黑盒测试与白盒测试

##### 2.2 需要多少次测试

##### 2.3 系统测试

##### 2.3.1 浏览器测试

##### 2.3.2 自动测试

##### 2.3.3 孤立测试

##### 2.3.4 验收测试

##### 2.3.5 系统测试的局限

##### 2.4 单元测试

##### 2.4.1 返回值

##### 2.4.2 依赖性

##### 2.4.3 副作用

##### 2.5 案例探究

- 2.5.1 分析待测试代码
- 2.5.2 创建测试环境
- 2.5.3 避免全局依赖性
- 2.5.4 独立于数据源的测试
- 2.5.5 异步事件测试
- 2.5.6 数据库变化的存储
- 2.5.7 意外的结果
- 2.5.8 封装输入数据
- 2.5.9 问题的进一步研究
- 2.6 本章小结
- 第II部分 最佳实践
- 第3章 TYP03：呆板项目之敏捷未来
- 3.1 简介
- 3.1.1 TYP03的历史：13个春秋
- 3.1.2 敢于重新开始
- 3.1.3 我们的测试经验
- 3.2 方法和技术
- 3.2.1 苦甜参半的“大块头”
- 3.2.2 测试驱动开发
- 3.2.3 文档式测试
- 3.2.4 持续集成
- 3.2.5 整洁代码
- 3.2.6 代码重构
- 3.2.7 编程向导
- 3.2.8 领域驱动设计
- 3.3 开发过程中的行动方向
- 3.3.1 开发新代码
- 3.3.2 扩展和调试代码
- 3.3.3 优化代码
- 3.3.4 寻找并修复漏洞
- 3.3.5 处理旧代码
- 3.4 测试处方
- 3.4.1 无意的功能性单元测试
- 3.4.2 文件系统入口
- 3.4.3 接口处的构造函数
- 3.4.4 测试抽象类
- 3.4.5 测试受保护方法
- 3.4.6 回调函数的使用
- 3.5 迈向未来
- 第4章 单元测试的糟糕实践
- 4.1 为什么测试质量很重要
- 4.2 糟糕实践和测试味道
- 4.2.1 测试代码重复
- 4.2.2 断言复杂和测试试图
- 4.2.3 脆弱测试
- 4.2.4 模糊测试
- 4.2.5 谎言测试

4.2.6 慢测试

4.2.7 测试中的条件逻辑

4.2.8 自行验证测试

4.2.9 Web冲浪测试

4.2.10 模拟过度杀伤

4.2.11 跳过蔓延

4.3 本章小结

第5章 Digg公司的质量保证

5.1 面临的问题

5.1.1 遗留基本代码

5.1.2 如何解决这些问题

5.1.3 选择测试框架

5.1.4 引进专家

5.1.5 会议室内的一周

5.2 训练团队

5.3 编写可测试代码

5.3.1 规避静态方法

5.3.2 依赖注入

5.4 模拟对象

5.4.1 综述

5.4.2 数据库

5.4.3 松散耦合依赖

5.4.4 测试类内部构件的试验对象观察者

5.4.5 分布式缓存

5.4.6 模拟服务型架构

5.5 Digg的质量保证过程

5.5.1 测试

5.5.2 益处

5.6 本章小结

第III部分 服务器与服务

第6章 服务型API测试

6.1 问题

6.2 解决方案

6.2.1 API证书

6.2.2 API限制

6.2.3 服务协议的脱机测试

6.2.4 具体服务的脱机测试

6.3 本章小结

第7章 测试Web DAV服务器

7.1 关于eZ Web DAV组件

7.1.1 Web DAV

7.1.2 体系结构

7.2 开发过程中的挑战

7.2.1 需求分析

7.2.2 RFC之后的TDD

.....



## 章节摘录

1.1外部质量 客户或者应用程序的终端用户最关心他们可见的质量问题。

下面列出了外部质量包含的几个方面： 功能性（Functionality）：指应用程序能够完成预期的任务。

易用性（Usability）：指应用程序能够高效工作并且能够满足用户需求。

可访问性属于易用性。

反应性（Reactivity）：指应用程序能够在短时间内作出反应，这对于满足用户需求至关重要。

安全性（Security）：用户可感觉到的安全性，尤其是构建成功的软件的另一重要因素。

可用性（Availability）与可靠性（Reliability）：对于拥有大量用户的Web应用程序而言，可用性与可靠性是相当重要的。

因为Web应用程序必须能够承载高负荷甚至可以在异常情况下正常工作。

将应用程序作为一个整体进行测试能够检验软件外部质量的各个方面，这称为端到端测试（end-to-endtest）。

我们可以记录客户需求作为验收测试（acceptancetest）的标准。

验收测试不仅是客户与开发人员的一种很好的沟通方式，而且可以自动验证软件产品是否满足所有功能需求。

为了提高应用程序的反应，需要检测其响应时间。

可以使用一些辅助的工具和技术手段以确保软件的低耗费、实现软件的最优化。

将来需要修正应用程序或流量突增时，开发人员和管理者必须能够知道软件潜在的瓶颈在哪里，这样才能确定项目规模。

从长远来看，所有这些信息均有助于软件可用性与可靠性的质量保证。

1.2内部质量 开发人员与管理人员对于应用程序的需求就是应用程序的内部质量。

开发人员关注应用程序的易理解、易调整、易扩充。

如果没有满足这些需求，以后修改应用程序将会很困难，修改的代价也会很高。

这带来的危害是，对应用程序的一个小小的改动都可能会导致意想不到的副作用。

客户与终端用户几乎觉察不到软件的内部质量。

终端用户希望软件满足其所有需求，至少需要满足其预期功能并且软件要易于操作。

当然，如果软件能够通过验收测试且运行速度还够快，那么大多数客户都会满意。

但是，内部质量缺陷通常需要在一段时期后才能够暴露出来。

即使修正小的故障也不是一蹴而成的事情。

对软件的任何修改和扩展都会花费巨大代价。

通常而言，开发人员早晚都会提出申请资金预算，用于整理或者重构代码，但是客户或者管理层看不到重构代码带来的好处，这样的申请多半不会被批准。

第2章 将会讨论独立软件模块（单元测试）的自动测试。

自动测试能够立即反馈修改代码时产生的bug，如不借助自动测试技术，代码重构工作将会非常麻烦。

质量保证（qualityassurance），或者确切地说，质量管理（quality management）的主要目标是：参与项目的各个方面都认可内部质量的开支和受益。

.....



## 编辑推荐

《开发高质量PHP框架与应用的实际案例解析》主要内容阐述优秀内部和外部软件质量的特征，介绍编写新代码、改变和优化既有代码、找到并修复bug的技术，展示差的测试实践，因此您能够知道应该避免什么，论述如何测试面向服务的API、WebDAV服务器以及其他HP框架，重温大规模的基于selenium的测试和测试数据库交互。

#### 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>