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

Sebastian Bergmann, Stefan Priebsch

REAL-WORLD SOLUTIONS FOR DEVELOPING HIGH-QUALITY PHP FRAMEWORKS AND APPLICATIONS

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

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

Sebastian Bergmann
Stefan Priebsch

Wiley Publishing, Inc.

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

Published by
Wiley Publishing, Inc.
10475 Crosspoint Boulevard
Indianapolis, IN 46256
www.wiley.com

Published by Wiley Publishing, Inc., Indianapolis, Indiana

Published simultaneously in Canada

ISBN: 978-0-470-87249-9
ISBN: 978-1-118-09822-6
ISBN: 978-1-118-09824-0
ISBN: 978-1-118-09823-3

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

Library of Congress Control Number: 2010939958

ABOUT THE AUTHORS

SEBASTIAN BERGMANN (thePHP.cc) holds a degree in computer science and is a pioneer in the field of quality assurance in PHP projects. His test framework, PHPUnit, is a de facto standard. He is actively involved in the development of PHP and is the creator of various development tools. Sebastian Bergmann is an internationally sought-after expert. As an author, he shares his long-standing experience in books and articles. He is a frequent speaker at conferences around the world.

STEFAN PRIEBSCH (thePHP.cc) is a co-founder and Principal Consultant with thePHP.cc. He holds a degree in computer science and is the author of various books and technical articles. As a consultant, he helps customers to improve development processes and make better use of PHP, with a focus on software architecture, OOP, design patterns, and tools and methods. Stefan is a frequent speaker at IT conferences around the world.

CREDITS

EXECUTIVE EDITOR
Carol Long

PROJECT EDITOR
Tom Dinse

CONSULTING AND TECHNICAL EDITOR
Elizabeth Naramore

PRODUCTION EDITOR
Daniel Scribner

COPY EDITOR
Gwenette Gaddis

EDITORIAL DIRECTOR
Robyn B. Siesky

EDITORIAL MANAGER
Mary Beth Wakefield

FREELANCER EDITORIAL MANAGER
Rosemarie Graham

ASSOCIATE DIRECTOR OF MARKETING
Ashley Zurcher

PRODUCTION MANAGER
Tim Tate

VICE PRESIDENT AND EXECUTIVE GROUP PUBLISHER
Richard Swadley

VICE PRESIDENT AND EXECUTIVE PUBLISHER
Barry Pruett

ASSOCIATE PUBLISHER
Jim Minatel

PROJECT COORDINATOR, COVER
Katherine Crocker

PROOFREADER
Louise Watson, Paul Sagan,
Word One New York

INDEXER
Ron Strauss

COVER DESIGN
Michael E. Trent

COVER IMAGE
© istockphoto.com/Dmitry Mordvintsev

CONTENTS

FOREWORD

Building and assuring quality software is not a new concept, and few will argue it is not important. I have had the privilege of building truly mission-critical operational software for many years—the kind where people's lives can be at stake. During that time, I learned lots about how to implement and drive a quality process from project inception to mission-critical use. Creating a high-quality process is not trivial, requires the support and commitment of the organization's leadership, and can impact choice of people, systems, processes, communications, and even organizational structures.

In my opinion, the challenges of the Internet's broad reach and pace dwarf the challenges of the mission-critical systems I was building. While many of these new systems are "only" business-critical, the truth is that they are no less critical and are dealing with additional layers of complexity such as more distributed development teams, well-known and evolving security attacks on web standards and software, internationalization challenges, shorter release cycles in SaaS settings, and more. In addition, in e-commerce applications, where downtime directly equates to money, the requirement for a strong quality-assurance program is even more critical and requires a special emphasis on compliance, quick time to fix (and time to verify and deploy that fix), and ability to run real-time end-to-end transactions to ensure not only that the application is up, but also that transactions can actually occur. In addition, the increasing emphasis on user experience also means that perceived quality has become increasingly critical and a functioning system that is not delivering the desired user experience has to be enhanced within a very short time frame without compromising on quality. The quality process and systems have to support such rapid turnaround of changes at high quality.

Suffice to say that these challenges have led to significant innovation and changes in approach when it comes to quality assurance compared with the well-established past practices of building mission-critical software. Software development has made huge strides over the past years in establishing strong best practices and awareness around quality assurance. Some key advances include a recognition that the developers must be a huge part of the quality process and cannot defer sole responsibility to the quality team. Continuous integration methodology mitigates one of the biggest challenges and bottlenecks in pushing out quality software—the integration stage. A more strategic focus on automated testing enables pushing out fixes faster, and not only meeting agreed-upon SLAs but exceeding SLAs to deliver end-user satisfaction.

This book puts a strong focus on many of the required quality-assurance disciplines with a very practical focus on PHP and covers people, systems, processes, and tools. The authors of this book have a great mix of both practical and theoretical knowledge of the subject, and I cannot think of better authors to write such a book. Not only do they have practical experience from a diverse set of real-life projects, but they also have created and contributed to quality tools within the PHP eco-system. In addition, the focus on case studies is invaluable as a way to learn from others and understand how everyone tailors his best practices to his situation.

I am sure this book will serve you well in taking the quality of your projects to the next level and help you instill an even stronger sense of pride in the software you are creating within your teams and your management.

—ANDI GUTMANS
Menlo Park, CA
February 2010

INTRODUCTION

Experience: that most brutal of teachers. But you learn, my God do you learn.

— C.S. Lewis

ABOUT THIS BOOK

According to the TIOBE Programming Community Index, PHP is the most popular programming language after C/C++ and Java.[1] Gartner predicts that dynamic programming languages will be critical to the success of many next-generation application development efforts and sees PHP as one of the strongest representatives of this type of programming language.[2] Since the beginning, PHP was designed for web application development and was likely one of the driving forces behind the dot-com boom at the turn of the millennium. Since then, PHP has matured to a general-purpose programming language that supports both procedural and object-oriented programming. In the past, subjects such as performance, scalability, and security were hot in the PHP community. In recent years, however, architecture and quality are getting more attention. In our consulting practice, we see more enterprises that want to modernize their PHP-based software and to base their development processes on agile values. The modernization of a code base is usually driven by a migration from PHP4 to PHP5 or by the introduction of a framework to standardize development.

Against this backdrop, it is hardly surprising that a plethora of PHP frameworks exists. All these frameworks want to help with solving recurring use cases and the standardization of application development. Dynamic and static testing techniques as well as automated builds and continuous integration are no longer alien concepts to PHP developers. Especially in enterprise-critical applications, simple PHP programming has evolved into software engineering with PHP.

Is This a PHP Book?

Based on examples from the PHP world, this book teaches the planning, execution, and automation of tests for the different software layers, the measuring of software quality using software metrics, and the application of appropriate practices such as continuous integration. We assume the reader is either an experienced PHP developer and interested in quality assurance for PHP projects or a developer who is proficient enough with another programming language to follow the examples.

---

[1]TIOBE Software BV, "TIOBE Programming Community Index for December 2010," accessed December, 2010, `http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html`.

[2]Gartner Inc., "Dynamic Programming Languages Will Be Critical to the Success of Many Next-Generation AD Efforts," 2008, accessed April 10, 2010, `http://www.gartner.com/DisplayDocument?ref=g_search&id=832417`.

This book cannot and does not want to be an introduction to (object-oriented) programming with PHP5. And although many companies think about quality-assurance measures for the first time while migrating from PHP4 to PHP5, topics related to migration of PHP applications and environments are not covered in this book. For those topics, refer to *Professionelle Softwareentwicklung mit PHP 5: Objektorientierung Entwurfsmuster, Modellierung und fortgeschrittene Datenbankprogrammierung* by Sebastian Bergmann (dpunkt.verlag, 2005, ISBN 978-3-89864-229-3) and *PHP migrieren: Konzepte und Lösungen zur Migration von PHPAnwendungen und -Umgebungen* by Stefan Priebsch (Carl Hanser Verlag, 2008, ISBN 978-3-446-41394-8).

In addition to developers, project leaders and quality-assurance engineers should be concerned with the topic of quality assurance. We hope this book fosters mutual trust among all stakeholders of a software project and motivates all readers to improve the internal quality of their software (see the "Internal Quality" section in Chapter 1).

Structure of the Book

Following the idea that we learn best through experience—and especially from the experience of others—this book brings together case studies that allow a look behind the scenes of well-known enterprises and projects and imparts valuable practical experience.

The first part, "Foundations," explains how we define and understand software quality and how you can test the different layers of software. Part II, "Best Practices," shows tried and true approaches and strategies (for instance, with regard to the writing of unit tests) and how the developers of Digg Inc. and the TYPO3 project implement them. The "Unit Testing Bad Practices" chapter captures the same topic from a different angle and shows the pitfalls you should avoid while writing unit tests.

Part III, "Servers and Services," discusses testing service-oriented APIs and server components. Part IV, "Architecture," uses Symfony as an example to show how both a framework itself and applications built using it can be tested. Using the Graph component from eZComponents, we discuss how a good architecture of loosely coupled objects can enable even the testing of binary image data. Testing database interaction is a topic that affects multiple layers of an application's architecture and thus is worth its own chapter.

In Part V, "Q&A in the Large," the developers of studiVZ and swoodoo report on their experience with quality assurance in large projects and teams. Chapter 12, "Continuous Integration," brings together dynamic and static testing techniques and shows how the various quality-assurance tools can be effectively used together.

The last part, "Non-Functional Aspects," tops off the book with chapters covering usability, performance, and security.

ABOUT THE CASE STUDY AUTHORS

Robert Lemke, TYPO3 Association, and Karsten Dambekalns, TYPO3 Association

Robert Lemke is co-founder of the TYPO3 Association and leads the development of TYPO3 v5 (code name Phoenix) and FLOW3. He's passionate about agile development methods and clean code. Robert lives in Lübeck, with his wife Heike, his daughter Smilla, and Vibiemme, their espresso machine.

Karsten Dambekalns, learned the basics of web technology the hard way—by looking at other websites' HTML source code. After using OS/2, Windows, and Linux, he now uses a Mac. All this happened after he learned BASIC and Assembler on a good old Commodore C128. Using PHP since 1999, Karsten discovered TYPO3 in 2002 and got caught by its immense possibilities. Now he is part of the TYPO3 Phoenix and FLOW3 core development teams and is a Steering Committee member of the TYPO3 Association.

In their chapter, "TYPO3: The Agile Future of a Ponderous Project," Robert Lemke and Karsten Dambekalns show foundations and techniques that the TYPO3 project uses to improve the quality of their software in a sustainable way.

Benjamin Eberlei, direkt:effekt GmbH

Benjamin Eberlei is software developer at direkt:effekt GmbH. In his spare time, he is part of the Doctrine team, maintains some components for the Zend Framework, and contributes to a handful of other small open-source projects.

In his "Unit Testing Bad Practices" chapter, Benjamin Eberlei shows common mistakes you should avoid while writing unit tests in order to maximize the return on investment from automated testing of your software.

Matthew Weier O'Phinney, Zend Technologies Ltd.

Matthew Weier O'Phinney is Project Lead for Zend Framework and has been contributing to the project since before the initial preview release. He is an advocate for open-source software and regularly blogs and speaks on PHP best practice topics. You'll find him most often in Vermont, where he resides with his wife, daughter, son, and aging basset hound.

In his "Testing Service-Oriented APIs" chapter, Matthew Weier O'Phinney highlights the challenges of testing web services and presents solutions that proved successful in the Zend Framework project.

Tobias Schlitt

Tobias Schlitt has a degree in computer science and has worked for more than 10 years on professional web projects using PHP. As an open-source enthusiast, he contributes to various community projects. Tobias is a co-founder of Qafoo GmbH, which provides services for high-quality PHP development. This includes consulting and training for quality assurance and better programming, as well as technical support for several PHP QA tools.

In his "Testing a WebDAV Server" chapter, Tobias Schlitt shows that you sometimes have to resort to unusual methods to reach your goals when writing automated tests.

Fabien Potencier, Sensio Labs

Fabien Potencier[3] discovered the Web in 1994, at a time when connecting to the Internet was still associated with the harmful, strident sounds of a modem. Being a developer by passion, he immediately started to build websites with Perl. But with the release of PHP5, he decided to switch focus to PHP and created the Symfony framework[4] project in 2004 to help his company leverage the power of PHP for its customers. Fabien is a serial entrepreneur, and among other companies, he created Sensio, a services and consulting company specializing in web technologies and Internet marketing, in 1998. Fabien is also the creator of several other open-source projects, a writer, blogger, speaker at international conferences, and happy father of two wonderful kids.

In his "Testing Symfony and Symfony Projects" chapter, Fabien Potencier reports his experience from the Symfony project and shows how the testing of Symfony improved the framework's programming interfaces.

Kore Nordmann

Kore Nordmann has a degree in computer science. During his studies, he worked as a developer and software architect at eZ Systems, the manufacturer of the leading enterprise open-source CMS. In addition, he is developing and/or maintaining various open-source projects, such as eZComponents, Arbit, WCV, Image 3D, PHPUnit, and others. For several years, Kore has been a regular speaker at national and international conferences and has published several books and articles. He offers consulting as a partner in Qafoo GmbH.

In his "Testing the ezcGraph Component" chapter, Kore Nordmann describes how a well-designed architecture, together with the use of mock objects, enables even the testing of a component that produces binary output.

Michael Lively Jr., Selling Source LLC

Michael Lively has been working with PHP since 2001 and has been involved to some degree with the PHP testing community since 2005. He is the creator of the database testing extension for PHPUnit and makes other small contributions to PHPUnit. He is now an Application Architect for

[3] http://fabien.potencier.org

[4] http://www.symfony-project.org

Las Vegas-based Selling Source LLC. While working with Selling Source, he has worked on several projects, including enterprise-level loan management and processing platforms written in PHP that service millions of customers and hundreds of agents.

In his "Testing Database Interaction" chapter, Michael Lively Jr. documents the functionality of DbUnit, PHPUnit's extension for database testing, and shows how this powerful tool can be leveraged effectively.

Christiane Philipps, Rebate Networks GmbH, and Max Horváth, Vodafone GmbH

Christiane Philipps is CTO and Agile Enthusiast with Rebate Networks. She regularly writes about agile testing and agile leadership, topics that are dear to her heart, in her blog.[5]

Max Horváth is Lead Software Engineer with Vodafone Internet Services and has more than 10 years of experience with web development. At the time of this writing, he was the Team Lead Mobile Development at VZnet Netzwerke.

In their "Quality Assurance at studiVZ" chapter, Christiane Philipps and Max Horváth report how they successfully introduced PHPUnit and Selenium RC for one of Europe's largest social networking platforms.

Manuel Pichler, OnVista Media GmbH, and Sebastian Nohn, Ligatus GmbH

Manuel Pichler is the creator of PHP quality assurance tools such as PHP Depend,[6] PHPMD,[7] and phpUnderControl.[8] He is a co-founder of Qafoo GmbH, which provides services for high-quality PHP development.

Sebastian Nohn started developing dynamic websites in 1996 and has handled quality assurance in commercial and open-source projects since 2002. He was one of the first to use CruiseControl for the continuous integration of PHP projects.

In their chapter, "Continuous Integration with phpUnderControl," Manuel Pichler and Sebastian Nohn report how continuous integration, retroactively written unit tests, software metrics, and other static testing techniques helped improve the software quality of a legacy application.

Lars Jankowfsky, swoodoo AG

Lars Jankowfsky is the CTO of swoodoo AG and is responsible for the PHP-based flight and hotel price comparison service. He has been developing web applications for over 15 years and has used PHP since its early versions. Another passion of his is leading eXtreme Programming teams.

[5] http://agile-qa.de

[6] http://pdepend.org/

[7] http://phpmd.org/

[8] http://phpUnderControl.org/

In his chapter, "swoodoo: A True Agile Story," Lars Jankowfsky tells how swoodoo introduced agile methods and a service-oriented architecture to allow for the smooth and continuous evolution of their product.

Jens Grochtdreis

Jens Grochtdreis[9] is a self-employed web developer and consultant who specializes in front-end development and accessibility.

In his "Usability" chapter, Jens Grochtdreis shows how to develop websites that are comprehensible and easy to use, as well as how to test for usability.

Brian Shire

Brian Shire started programming around age eight on an Apple IIe. When he wasn't playing games, he was learning the Basic programming language. At the time of this writing, Brian was working for Facebook, Inc., where he focused on scaling their PHP infrastructure. In his four years with Facebook, the site grew from 5 million to over 175 million users. During this time, Brian became a primary contributor to APC, an opcode and user variable cache for PHP. He also made contributions to PHP itself and other PECL extensions. Brian has shared some of this knowledge and experience as a speaker at various conferences around the world. He currently resides in San Francisco, California, and maintains a personal and technical blog.[10]

In his "Performance Testing" chapter, Brian Shire provides motivation for performance testing of web applications and introduces the reader to the appropriate tools and processes for performance testing.

Arne Blankerts, thePHP.cc

Arne Blankerts has long-standing experience as Head of IT. His software fCMS makes innovative use of XML technologies and is vital to business-critical applications in international corporations. He is actively involved with the documentation of PHP. Arne Blankerts is an expert on IT security and writes about this in a magazine column. He is a sought-after speaker at international conferences and a book author, and he publishes articles in IT magazines.

In his "Security" chapter, Arne Blankerts shows how easy the development of fundamentally secure applications is if you know the common attack vectors and follow some important rules.

ERRATA

We make every effort to ensure that there are no errors in the text or in the code. However, no one is perfect, and mistakes do occur. If you find an error in one of our books, like a spelling mistake or faulty piece of code, we would be very grateful for your feedback. By sending in errata, you might save another reader hours of frustration, and at the same time, you will be helping us provide even higher-quality information.

[9] http://grochtdreis.de

[10] http://tekrat.com