



## Vorwort

Sebastian Bergmann, Stefan Priebsch

Softwarequalität in PHP-Projekten

ISBN (Buch): 978-3-446-43539-1

ISBN (E-Book): 978-3-446-43582-7

Weitere Informationen oder Bestellungen unter

<http://www.hanser-fachbuch.de/978-3-446-43539-1>

sowie im Buchhandel.

# Geleitwort

Die Entwicklung qualitativ hochwertiger Software sowie das Sicherstellen der Softwarequalität sind keine neuartigen Konzepte, und kaum jemand wird widersprechen, dass diese Konzepte für die Software-Entwicklung von enormer Bedeutung sind. Ich hatte das Privileg, über viele Jahre wirklich missionskritische Software zu entwickeln. Ich meine die Art von Software, von der Menschenleben abhängen.

Während dieser Zeit habe ich eine Menge darüber gelernt, wie man einen Prozess für die Qualitätssicherung zu Beginn eines Projektes einführt und bis zum unternehmenskritischen Einsatz in der Produktion vorantreibt. Das Gestalten eines Entwicklungsprozesses, der zu qualitativ hochwertiger Software führt, ist nicht trivial und erfordert nicht nur Unterstützung durch das Management, sondern auch dessen eigenes Engagement. Dies hat Einfluss auf die Aufbauorganisation des Unternehmens sowie seine Mitarbeiter, Systeme und Prozesse.

Meiner Meinung nach stellen die Probleme, die sich aus der großen Reichweite sowie aus der hohen Entwicklungsgeschwindigkeit des Internets ergeben, die Probleme in den Schatten, denen ich mich bei der Entwicklung der oben erwähnten missionskritischen Software stellen musste. Während viele dieser neuen Software-Systeme „nur“ unternehmenskritisch sind, sind sie in Wahrheit nicht weniger wichtig und müssen zusätzliche Schwierigkeiten wie beispielsweise Internationalisierung, Schutz vor bekannten und neuen Angriffen im Web sowie die Arbeit in verteilten Teams und mit immer kürzeren Release-Zyklen bewältigen.

Im E-Commerce-Bereich schlägt sich ein Ausfall der Anwendung direkt in einem Ausfall von Einnahmen nieder; daher ist dort der Bedarf der Softwarequalität noch größer. Besonderes Augenmerk liegt hierbei auf Compliance sowie der Möglichkeit, eventuelle Fehler schnell beheben und diese Fehlerbehebung sofort testen und umgehend bereitstellen zu können. Die Anwendung muss nicht nur „online“, sondern tatsächlich in der Lage sein, Transaktionen in Echtzeit zu verarbeiten.

Die wachsende Bedeutung der *User Experience* führt außerdem dazu, dass die durch die Benutzer der Anwendung wahrgenommene Qualität unternehmenskritisch wird: Die Software muss nicht nur korrekt funktionieren, sondern auch den Erwartungen der Benutzer genügen. Ist dies nicht der Fall, müssen die entsprechenden Änderungen in kürzester Zeit so umsetzbar sein, dass die Softwarequalität nicht beeinträchtigt wird. Sowohl die Prozesse für die Entwicklung und Qualitätssicherung der Software als auch die eingesetzten Systeme müssen diese schnellen Entwicklungen unterstützen.

Diese Herausforderungen haben zu signifikanten Änderungen im Bereich der Qualitätssicherung geführt, besonders im Vergleich dazu, wie früher auftragsentscheidende Software entwickelt wurde. Die Software-Entwicklung hat in den letzten Jahren große Fortschritte gemacht. Best Practices wurden etabliert und das Bewusstsein für Softwarequalität wurde gestärkt. Zu den wichtigsten Fortschritten gehört hierbei die Erkenntnis, dass die Entwickler in die Qualitätssicherung einbezogen werden müssen und die entsprechende Verantwortung nicht alleine bei der QA-Abteilung liegen darf. Die Methodik der kontinuierlichen Integration entschärft eines der größten Probleme – und einen der größten Flaschenhälse – bei der Entwicklung von qualitativ hochwertiger Software: die Integrationsphase. Der strategische Fokus auf automatisiertes Testen ermöglicht es, Fehlerbehebungen schneller ausliefern zu können. Hierdurch können Dienstgütevereinbarungen (englisch: *Service Level Agreements*) nicht nur eingehalten, sondern meist auch übererfüllt werden, was zu einer höheren Kundenzufriedenheit führt.

Dieses Buch behandelt die verschiedenen Disziplinen der Qualitätssicherung und ihre Auswirkungen auf Menschen, Systeme, Prozesse sowie Werkzeuge. Hierbei liegt der Fokus auf der praktischen Anwendung in PHP-basierten Projekten. Die in diesem Buch zusammengetragenen Fallstudien sind von unschätzbarem Wert. Sie erlauben das Lernen von anderen Entwicklerteams, beispielsweise wie diese Best Practices und Werkzeuge „in der richtigen Welt“ einsetzen. Die Autoren verfügen über eine beispiellose Mischung aus theoretischem Hintergrundwissen und praktischer Erfahrung aus dem Alltag großer Projekte, die mit PHP realisiert wurden. Darüber hinaus leisten sie durch die Entwicklung von Werkzeugen für die Qualitätssicherung einen entscheidenden Beitrag für das PHP-Ökosystem. Ich kann mir keine besseren Autoren für ein solches Buch vorstellen.

Ich bin mir sicher, dass Ihnen dieses Buch dabei helfen wird, die Qualität Ihrer Projekte zu steigern, sodass sowohl Ihr Entwicklerteam als auch das Management auf die entwickelte Software stolz sein können.

*Andi Gutmans, CEO Zend Technologies Ltd.*

# Vorwort

*„Experience: that most brutal of teachers.  
But you learn, my God do you learn.“* — C.S. Lewis

## ■ Über dieses Buch

78% aller Webseiten, von denen bekannt ist, in welcher Programmiersprache sie realisiert sind, setzen PHP ein [W3Techs 2012]. Gartner geht davon aus, dass viele neue, geschäftskritische Projekte von Unternehmen in einer dynamischen Sprache realisiert werden, und sieht PHP als derzeit stärksten Vertreter dieser Art von Programmiersprache [Gartner 2008]. PHP war von Anfang an für die Webprogrammierung konzipiert und dürfte um die Jahrtausendwende einer der wesentlichen Motoren des Dotcom-Booms gewesen sein.

Mittlerweile ist PHP zu einer Mehrzweckprogrammiersprache gereift und unterstützt sowohl die prozedurale als auch die objektorientierte Programmierung. Waren in der Vergangenheit Themen wie Performanz, Skalierbarkeit und Sicherheit Dauerbrenner in der PHP-Community, so sind in den letzten Jahren neben Buzzwords wie Cloud und NoSQL auch Themen wie Architektur und Qualität in den Fokus gerückt.

Es gibt heute eine Fülle an PHP-Frameworks. Diese wollen beim Lösen wiederkehrender Probleme und bei der Standardisierung der Anwendungsentwicklung helfen. In letzter Zeit sind auch viele leichtgewichtige Micro-Frameworks entstanden. Bei mehr und mehr Entwicklern setzt sich allerdings die Erkenntnis durch, dass ein Framework allein keine Architektur ausmacht. Im Sinne der Entkopplung und Wiederverwendbarkeit sollte man ohnehin die zentrale Geschäftslogik außerhalb eines Frameworks implementieren. Vor diesem Hintergrund ist das Thema Qualitätssicherung wichtiger denn je.

Dynamische und statische Testverfahren sowie automatisierte Builds und kontinuierliche Integration sind für viele PHP-Entwickler längst keine Fremdwörter mehr. Aus PHP-Programmierung ist, gerade in unternehmenskritischen Projekten, Software-Engineering mit PHP geworden.

## Anmerkungen zur zweiten Auflage

Wir waren sehr erfreut, als uns der Verlag mitteilte, dass die erste Auflage dieses Buches bald vergriffen sei, und uns nach einer zweiten Auflage fragte. Am liebsten hätten wir das halbe Buch neu geschrieben – nicht etwa, weil der Inhalt schlecht ist, sondern vielmehr weil man heute viele Dinge noch besser erklären könnte.

Leider ist der Zeitplan für eine zweite Auflage deutlich enger gesteckt, als dies bei einer Erstauflage der Fall ist. Allein deshalb ist es vielen Fallstudienautoren nicht möglich, ihre Fallstudien zu überarbeiten oder gar neu zu verfassen, um beispielsweise geänderten Rahmenbedingungen oder neuen Software-Versionen Rechnung zu tragen. Hinzu kommt, dass viele unserer Kontributoren heute in neuen Firmen oder Projekten arbeiten.

Die Versionen einiger Software-Produkte, auf die in Fallstudien Bezug genommen wird, sind nicht mehr aktuell, wenn Sie dieses Buch lesen. Symfony ist ebenso wie Zend Framework in Version 2 erschienen. Symfony 2 nutzt, wie bereits in der Fallstudie angedeutet, im Gegensatz zu Symfony 1 PHPUnit anstelle von Lime.

Die eZ Components waren zeitweise unter dem Namen Zeta Components ein Projekt der Apache Software Foundation. Es ist schade, dass die Zukunft dieses Projekts derzeit unklar ist, denn die Qualität dieser Software ist sehr gut.

Die in Kapitel 5 beschriebene Software phpUnderControl wird nicht mehr weiterentwickelt. Nichtsdestotrotz sind die in diesem Kapitel vorgestellten Konzepte richtig und wichtig. Das heute am häufigsten eingesetzte Werkzeug für die kontinuierliche Integration ist Jenkins. Eine gute Einführung in den Einsatz von Jenkins für PHP-Projekte bietet [\[Bergmann 2011a\]](#).

Die in Kapitel 17 in Aussicht gestellte Veröffentlichung einer domänenspezifischen Sprache zur Realisierung von Systemtests ist niemals erfolgt, da die beiden Autoren der Fallstudie das Unternehmen verlassen haben, bevor sie die Implementierung fertig stellen konnten.

Wir sind davon überzeugt, dass die beschriebenen Probleme und Lösungsansätze auch heute noch mindestens so relevant sind wie zum Zeitpunkt der Erstveröffentlichung. Nichtsdestotrotz haben wir uns – und hat sich die Welt – in den letzten Jahren weiterentwickelt. In der ersten Ausgabe schrieben wir:

*„Wir hoffen, dass dieses Buch das gegenseitige Verständnis zwischen den verschiedenen an Software-Projekten beteiligten Gruppen fördert und allen Lesern eine Motivation bietet, die interne Qualität [...] ihres Codes zu verbessern.“*

Damals waren wir uns noch nicht bewusst, dass wir mit diesem Satz ein Grundproblem der Software-Entwicklung angesprochen haben. Zu wenig „gegenseitiges Verständnis“ ist eine Folge von Kommunikationsproblemen, die nach unserer Erfahrung durch den Einsatz von noch mehr Software-Werkzeugen nicht gelöst werden können.

Gelingt es den Entwicklern nicht, dem eigenen Management oder dem Kunden den Wert von Qualitätssicherung zu vermitteln, so stimmt das Rollenbild, das die beteiligten Parteien vom Entwicklungsteam haben, vermutlich nicht überein.

Wir haben deshalb dieses Buch um ein Kapitel ergänzt, das Ansätze dafür liefert, wie Entwickler einem internen oder externen Kunden den Wert von Qualitätssicherung vermitteln können.

## Ist dies ein PHP-Buch?

Anhand von Beispielen aus der PHP-Welt vermittelt dieses Buch die Planung, Durchführung und Automation von Tests für die unterschiedlichen Software-Schichten, die Messung von Softwarequalität mithilfe von Software-Metriken sowie den Einsatz geeigneter Methoden wie beispielsweise kontinuierlicher Integration. Wir gehen davon aus, dass unsere Leser entweder erfahrene PHP-Entwickler sind, die sich für die Qualitätssicherung in PHP-Projekten interessieren, oder Entwickler, die mit einer anderen Programmiersprache so weit vertraut sind, dass sie den Beispielen folgen können.

Dieses Buch kann und will keine Einführung in die (objektorientierte) Programmierung mit PHP 5 sein. Und obwohl sich viele Unternehmen im Rahmen der Migration von PHP 4 nach PHP 5 zum ersten Mal intensiver mit Qualitätssicherung in einem PHP-Projekt auseinandersetzen, so kann auch die Migration von PHP-Umgebungen und -Anwendungen in diesem Buch nicht behandelt werden. Für diese beiden Themen sei auf [\[Bergmann 2005\]](#) und [\[Pribsch 2008\]](#) verwiesen.

Neben den Entwicklern müssen sich auch Projektleiter und Qualitätsbeauftragte mit dem Thema Softwarequalität befassen. Wir hoffen, dass dieses Buch das gegenseitige Verständnis zwischen den verschiedenen an Software-Projekten beteiligten Gruppen fördert und allen Lesern eine Motivation bietet, die interne Qualität (siehe Abschnitt [1.3](#)) ihres Codes zu verbessern.

## ■ Vorstellung der Autoren

### Sebastian Bergmann

Wenn der Launch neuer PHP-Projekte zu unerfreulichen Überraschungen führen könnte, ist das Know-how von Sebastian Bergmann gefragt. Der Diplom-Informatiker ist einer der Pioniere für Qualitätssicherung in PHP-Projekten. Die von ihm entwickelten Werkzeuge, darunter PHPUnit, sorgen weltweit dafür, dass PHP-Entwickler besser arbeiten können und Unternehmen Software mit weniger Fehlern generieren.

Seine Leistungen haben ihm einen exzellenten Ruf in der Fachwelt und bei seinen Kunden eingetragen. Zu Recht, hat er doch entscheidend daran mitgewirkt, dass PHP sich zu einer ernst zu nehmenden und zuverlässigen Plattform für große kritische Projekte entwickelt hat. Er besitzt die seltene Gabe, seine außergewöhnlichen technischen Fähigkeiten für jedermann nachvollziehbar zu vermitteln. Das macht ihn als Autor viel gelesener Bücher und Fachartikel ebenso beliebt wie als Redner, Coach oder Trainer.

Sebastian Bergmann entwickelt PHP aktiv weiter und ist Mitbegründer von thePHP.cc, die namhafte Firmen auf der ganzen Welt zu ihren Kunden zählt. Dass er nicht nur Kopf-, sondern auch Genussmensch ist, zeigt seine Vorliebe für feine Schokoladen und guten Tee. Sie kommt besonders dann zum Tragen, wenn ihn unberechtigte Bugreports für PHPUnit erreichen.

## Stefan Priebisch

Spätestens dann, wenn PHP-Architekturen ein unerwünschtes Eigenleben entwickeln, suchen viele Unternehmen den Kontakt zu Stefan Priebisch. Der Diplom-Informatiker ist seit vielen Jahren auf die Entwicklung PHP-basierter Software für Unternehmen spezialisiert. In der Branche gilt er als die Kapazität für Fragen der Software-Architektur und der Objektorientierten Programmierung.

Komplexes Fachwissen und breite Praxiserfahrung gehen bei ihm mit der Fähigkeit einher, genau zu erfassen, was sein Gegenüber benötigt. Ob als Berater von Unternehmen, als interessanter Redner für Fachauditorien oder als Trainer für IT-Entwickler – Stefan Priebisch ist ein gefragter Spezialist. Seine Fähigkeit, technische und unternehmerische Aspekte thematisch perfekt auszubalancieren, macht ihn zu einem viel gelesenen Autor von Publikationen, die in der Fachwelt ein nachhaltiges Echo gefunden haben.

Geschätzt wird er als geduldiger, ermutigender Ausbilder und Coach, der stets bereit ist, anderen seinen breiten Erfahrungsschatz zugänglich zu machen. Stefan Priebisch ist Mitbegründer von thePHP.cc, die namhafte Firmen auf der ganzen Welt zu ihren Kunden zählt. Als Vater von Zwillingen beschäftigt er sich übrigens auch privat intensiv mit Skalierungsproblemen.

## Benjamin Eberlei

Benjamin Eberlei arbeitet als IT-Berater bei der Qafoo GmbH im Bereich Qualitätssicherung für PHP-Anwendungen und Software-Architektur. Er ist Projektleiter der Open-Source-Bibliothek Doctrine und ist an zahlreichen Projekten im Symfony-Umfeld beteiligt.

In Kapitel 4 zeigt Benjamin Eberlei, welche Fehler man beim Schreiben von Tests vermeiden sollte, um den größtmöglichen Nutzen aus dem Testen von Software ziehen zu können.

## Manuel Pichler und Sebastian Nohn

Manuel Pichler kam erstmals 1999 mit der Programmiersprache PHP in Kontakt und ist ihr bis heute treu geblieben. Während seines Studiums sammelte er erste Erfahrungen im Bereich Qualitätssicherung. Im Anschluss an das Studium arbeitete er als Software-Architekt und entwickelte im Rahmen dieser Tätigkeit verschiedene Erweiterungen für CruiseControl, die dann 2007 die Grundlage für phpUnderControl bildeten. Neben diesem Projekt ist er auch der Schöpfer von PHP\_Depend und Werkzeugen zur statischen Analyse von PHP-Code. Im Sommer 2010 hat er zusammen mit Kore Nordmann und Tobias Schlitt die Qafoo GmbH gegründet und bietet darüber Support und Schulungen rund um das Themengebiet Qualitätssicherung an.

Sebastian Nohn beschäftigt sich seit 1996 mit dynamischen Websites und seit 2002 mit Qualitätssicherung im kommerziellen und Open-Source-Bereich. Er war einer der ersten, der CruiseControl für die Nutzung in PHP-Projekten adaptierte, und gab den Anstoß für die Entwicklung von phpUnderControl. Zurzeit ist er bei der Ligatus GmbH, einem der führenden Performance-Marketing-Anbieter, beschäftigt, wo er den Bereich Qualitätssicherung aufbaute und die Verantwortung als Teamleiter für die Bereiche Qualitätssicherung und

Infrastruktur trägt. Sebastian Nohn ist Wirtschaftsinformatiker und schreibt in unregelmäßigen Abständen in seinem Weblog über IT-Themen.

In Kapitel 5 berichten Manuel Pichler und Sebastian Nohn, wie kontinuierliche Integration, nachträglich eingeführte Unit-Tests, Software-Metriken und weitere statische Testverfahren dazu beigetragen haben, die Qualität einer Legacy-Applikation deutlich zu erhöhen.

## Michael Lively

Michael Lively arbeitet seit 2001 mit PHP und bringt sich seit 2005 in der PHP Testing Community ein. Er ist der Schöpfer der Erweiterung für Datenbanktests in PHPUnit, zu dem er auch weitere Patches beigetragen hat. Michael Lively arbeitet als Lead Developer und Application Architect für die SellingSource LLC mit Sitz in Las Vegas. Zu seinem Arbeitsbereich gehört die Entwicklung einer mit PHP realisierten Enterprise-Plattform für Kreditmanagement, die von Hunderten Maklern für Millionen von Kunden verwendet wird.

In Kapitel 6 dokumentiert Michael Lively die Funktionalität der Datenbankerweiterung von PHPUnit und zeigt, wie dieses mächtige Werkzeug effektiv eingesetzt werden kann.

## Jens Grochtdreis

Jens Grochtdreis ist selbstständiger Frontend-Entwickler und bringt die Erfahrung von zehn Jahren Arbeit in Agenturen mit. Er hat 2005 die Webkrauts gegründet und wirbt mit ihnen für ein modernes Web. Neben seinem Blog schreibt er für Zeitschriften, berät Verlage und hält Vorträge. Wenn er nicht gerade bloggt, twittert, surft oder codet, dann entspannt er sich bei Comics, Krimis, hört Blues oder kocht.

In Kapitel 7 zeigt Jens Grochtdreis, wie einfach nutzbare und verständliche Webseiten entwickelt werden können und wie die Gebrauchstauglichkeit getestet werden kann.

## Brian Shire

Brian Shire entdeckte das Programmieren im Alter von acht Jahren auf einem Apple IIe. Wenn er keine Spiele spielte, lernte er die Programmiersprache BASIC. Während der Arbeit an der Fallstudie für dieses Buch arbeitete er bei Facebook Inc., wo er für die Skalierung der PHP-Infrastruktur verantwortlich war. In seinen vier Jahren bei Facebook wuchs die Plattform von 5 Millionen Nutzern auf 175 Millionen Nutzer. In dieser Zeit wurde Brian zu einem Kernentwickler von APC, einem Bytecode- und Daten-Cache für PHP. Er trug außerdem zur Entwicklung des PHP-Interpreters sowie verschiedenen Erweiterungen in PECL bei. Brian teilt seine Erfahrung und sein Wissen als Referent auf internationalen Konferenzen und in seinem Blog. Zurzeit lebt er in San Francisco.

In Kapitel 8 motiviert Brian Shire das Testen der Performanz von Webanwendungen und führt in die wichtigsten Werkzeuge und Methoden dafür ein.



## Arne Blankerts

Arne Blankerts löst IT-Probleme, lange bevor viele Firmen überhaupt merken, dass sie sie haben. Seine Themenschwerpunkte sind IT-Sicherheit, Performanz und Ausfallsicherheit von Infrastrukturen. Mit fast magischer Intuition findet er dafür Lösungen, die unverkennbar seine Handschrift tragen. Weltweit vertrauen Unternehmen auf seine Konzepte und seine Linux-basierten Systemarchitekturen.

In Kapitel 9 zeigt Arne Blankerts, wie einfach das Schreiben grundsätzlich sicherer Anwendungen ist, wenn man die gängigen Angriffsvektoren kennt und einige wichtige Regeln beachtet.

## Judith Andresen

Judith Andresen kombiniert exzellente Projektmanagement-Kompetenz mit dem Know-how der strategischen Firmenleitung. Sie vermittelt als Projektcoach Fertigkeiten wie Kommunikation und Projektmanagement; als Beraterin unterstützt sie Entwicklung guter Projekt- und Unternehmenskulturen. Mit der Organisation der „PHP Unconferences“ (Germany, Europe) unterstützt sie auch ehrenamtlich den Austausch zwischen PHP-Entwicklern.

Der Wechsel in die testgestützte Programmierung erzeugt zunächst höhere Entwicklungskosten. Wie diese vorübergehenden Mehrkosten gegenüber Management und Auftraggeber argumentiert werden können, zeigt Judith Andresen in Kapitel 10. Neben einer Anleitung zur betriebswirtschaftlichen Überprüfung des Vorgehens werden dem Leser Optionen aufgezeigt, wie eine testgestützte Programmierung erfolgreich als Projektmethode vereinbart werden kann.

## Robert Lemke und Karsten Dambekalns

Robert Lemke ist Gründungsmitglied der TYPO3 Association und leitet die Entwicklung des Rewrites von TYPO3 sowie des Frameworks FLOW3. Er hat eine besondere Vorliebe für agile Entwicklungsmethoden und es sich zum Ziel gesetzt, neue Ansätze wie Domain-Driven Design oder aspektorientiertes Programmieren in der PHP-Welt zu etablieren. Robert lebt in Lübeck, zusammen mit seiner Frau Heike, seiner Tochter Smilla und Vibienne, ihrer Espressomaschine.

Karsten Dambekalns programmiert seit 1999 in PHP und entdeckte 2002 die immensen Möglichkeiten von TYPO3. Er ist heute einer der Kernentwickler von TYPO3 und FLOW3 sowie Mitglied im Steering Committee der TYPO3 Association. Nach der Gründung einer eigenen Firma im Jahre 2000 steht Karsten Dambekalns seit 2008 wieder als Freelancer voll im Dienst der TYPO3-Entwicklung. Außerdem ist er Autor, spricht auf Konferenzen und verbringt den Großteil seiner Freizeit mit seiner Frau und seinen drei Kindern.

In Kapitel 11 stellen Robert Lemke und Karsten Dambekalns Grundsätze und Techniken vor, mit denen das TYPO3-Projekt die Softwarequalität nachhaltig verbessern konnte.

## Fabien Potencier

Fabien Potencier entdeckte das Web 1994, zu einer Zeit, als das Verbinden mit dem Internet noch von kreischenden Tönen eines Modems begleitet wurde. Als passionierter Entwickler begann er sofort mit der Entwicklung von Webseiten mit Perl. Mit der Veröffentlichung von PHP 5 legte er seinen Fokus auf PHP und startete 2004 das Symfony Framework, um in seiner Firma die Mächtigkeit von PHP für Kunden voll nutzen zu können. Fabien ist ein „Serienunternehmer“, der 1998 neben anderen Firmen auch Sensio, einen auf Webtechnologien und Internet-Marketing spezialisierten Dienstleister, gegründet hat. Ferner ist er der Schöpfer mehrerer Open-Source-Projekte, Autor, Blogger und Referent auf internationalen Konferenzen sowie stolzer Vater von zwei wundervollen Kindern.

In Kapitel 12 berichtet Fabien Potencier von seinen Erfahrungen aus dem Symfony-Projekt und zeigt unter anderem, wie das Testen von Symfony zu besseren Programmierschnittstellen geführt hat.

## Kore Nordmann

Kore Nordmann entwickelt, leitet und plant seit mehreren Jahren verschiedene PHP-basierte Open-Source-Projekte. Im Sommer 2010 hat er zusammen mit Manuel Pichler und Tobias Schlitt die Qafoo GmbH gegründet und steht darüber Unternehmen zur Verfügung, um die Architektur und Qualität von Software-Projekten zu verbessern. Seine Expertise teilt er regelmäßig auf verschiedenen Konferenzen sowie in Artikeln und Büchern mit. Neben der Entwicklung der eZ Components leitet er die Entwicklung von Arbit, einer neuen Software für die Verwaltung von Software-Projekten, inklusive der Integration von Werkzeugen für die Qualitätssicherung.

In Kapitel 13 beschreibt Kore Nordmann, wie es mit einer guten Architektur und dem Einsatz von Mock-Objekten möglich ist, selbst eine Komponente, die binäre Grafikausgaben erzeugt, umfassend zu testen.

## Matthew Weier O’Phinney

Matthew Weier O’Phinney arbeitet als Project Lead für das Zend Framework. Zu seinen Aufgaben zählen das Release-Management ebenso wie die Implementierung und Verbreitung von Best Practices sowie die Kommunikation mit der Community. Matthew ist ein aktiver Befürworter von Open-Source-Software und PHP. Seit 2000 programmiert er in PHP und anderen Sprachen. Für Magazine wie PHP Architect und in seinem eigenen Blog sowie in der Zend DevZone schreibt er über aktuelle Themen.

In Kapitel 14 geht Matthew Weier O’Phinney auf die besonderen Herausforderungen beim Testen von Webdiensten ein und präsentiert Ansätze und Lösungen, die sich im Zend Framework-Projekt bewährt haben.

## **Tobias Schlitt**

Tobias Schlitt ist ausgebildeter Fachinformatiker und Diplom-Informatiker. Er beschäftigt sich seit 1999 mit der Entwicklung von Webanwendungen auf Basis von PHP und war, nach mehrjähriger Aktivität im PEAR-Projekt, maßgeblich an Architektur und Entwicklung der eZ Components beteiligt. Daneben beteiligt er sich dauerhaft an verschiedenen Open-Source-Projekten rund um PHP. Als anerkannter Experte ist er beratend in den Bereichen Software-Architektur und Qualitätssicherung tätig. Tobias Schlitt gründete Mitte 2010 zusammen mit Manuel Pichler und Kore Nordmann, beide ebenfalls als Autoren in diesem Buch vertreten, die Qafoo GmbH, welche Experten-Consulting, Training und Support rund um die Entwicklung von qualitativ hochwertigem PHP-Code und Qualitätssicherung in Software-Projekten anbietet.

In Kapitel 15 zeigt Tobias Schlitt, dass man beim Testen manchmal unkonventionelle Wege gehen muss, um seine Ziele erreichen zu können.

## **Lars Jankowsky**

Lars Jankowsky ist CTO der Yatego GmbH. Davor war er als CTO der Swoodo AG verantwortlich für den PHP-basierten Flug- und Hotel-Preisvergleich. Seit mehr als 15 Jahren entwickelt er Webanwendungen und nutzt PHP seit den frühen Anfängen. Neben der Entwicklung von Software ist die Leitung von eXtreme Programming-Teams eine seiner Leidenschaften.

In Kapitel 16 zeigt Lars Jankowsky, wie agile Methoden und eine serviceorientierte Architektur die sanfte und kontinuierliche Evolution einer Anwendung ermöglicht haben.

## **Christiane Philipps und Max Horváth**

Christiane Philipps ist Fachinformatikerin und hilft Startups als „Interim CTO“. Ihr Herz schlägt besonders für Agile Testing und Agile Leadership, Themen, über die sie auch in ihrem Blog regelmäßig schreibt. Von Frühjahr 2008 bis Herbst 2009 arbeitete sie bei VZnet Netzwerke, davon ein Jahr lang als Leiterin des Bereichs Quality Assurance & Deployment.

Max Horváth ist Mobile Software Architect bei der AGT Group GmbH und beschäftigt sich seit über zehn Jahren mit Webentwicklung. In dieser Zeit hat er mit Unternehmern, Entwicklern und Designern zusammengearbeitet, um Webprojekte für große und kleine Unternehmen umzusetzen. Während der Arbeit an der Fallstudie für dieses Buch war er Team Lead Mobile Development bei VZnet Netzwerke.

In Kapitel 17 berichten Christiane Philipps und Max Horváth, wie sie PHPUnit und Selenium RC erfolgreich in einem der größten sozialen Netzwerke Europas eingeführt haben.

**Robert Balousek, Matt Erkkila, Ian Eure,  
Bill Shupp, Jeremy McCarthy und Brian O’Neill**

Robert Balousek ist CTO von Numote, Matt Erkkila ist Infrastructure Team Lead bei Rdio, Ian Eure ist Entwickler bei BankSimple, Bill Shupp ist Engineering Manager bei OpenDNS, Jeremy McCarthy ist Test Engineering Manager bei Workday und Brian O’Neill ist Senior Automation QA Engineer bei Eventbrite.

In Kapitel 18 berichten Robert Balousek, Matt Erkkila, Ian Eure, Bill Shupp, Jeremy McCarthy und Brian O’Neill über die Einführung von testgetriebener Entwicklung sowie kontinuierlicher Integration bei Digg. Während der Arbeit an der Fallstudie für dieses Buch waren sie alle bei Digg Inc. beschäftigt.