

HANSER

Johann-Peter Hartmann, Björn Schotte

Enterprise PHP 5

Serviceorientierte und webbasierte Anwendungen für den
Unternehmenseinsatz

ISBN-10: 3-446-22563-3

ISBN-13: 978-3-446-22563-3

Vorwort

Weitere Informationen oder Bestellungen unter
<http://www.hanser.de/978-3-446-22563-3>
sowie im Buchhandel.



Vorwort

Eigentlich hatte Rasmus Lerdorf nur sehen wollen, wer seinen Lebenslauf anschaut. Er war auf der Suche nach einem neuen Job, und so hatte er seine Fähigkeiten nebst bisherigem Werdegang online gestellt. Weil er wissen wollte, welcher potenzielle Arbeitgeber seine Seiten besuchte, hatte er mit Hilfe eines kleinen Perl-Scripts die Seitenzugriffe mitloggen lassen. Aus diesem kleinen Skript entwickelte sich später eine ganze Bibliothek, die verschiedene Funktionen vom Versenden von Formularen bis hin zum Zählen der Seitenzugriffe enthielt – die erste Ur-Version von PHP also. Durch die vielen Funktionen entschloss sich Rasmus, diese Skripte mit einem kleinen Parser auszustatten, so dass er die Funktionen direkt in den HTML-Seiten aufrufen konnte. Und um das ganze schneller zu bekommen, programmierte er alles noch einmal in C nach – PHP/FI, die „Personal Homepage Tools/Form Interpreter“ waren geboren. Für nostalgische Momente empfehlen wir Ihnen einen Blick auf <http://museum.php.net/>.

Mit einer Plattform wie Xing oder LinkedIn hätte es also kein PHP gegeben. Aber ohne eine Plattform wie PHP würde es vermutlich auch keine Social Communities geben. Nachdem Rasmus die Version 2.0 seiner „persönlichen Heimatseitenwerkzeuge“ mit für damaliger großer Resonanz zum freien Download online gestellt hatte, fanden sich zwei Studenten in Israel, die für seine Sprache einen neuen Interpreter schrieben – PHP 3.0 entstand, entwickelt von Andi Gutmans und Zeev Zuraski. Die beiden gründeten etwas später die Firma Zend und entwickelten mit PHP 4 einen modernen Kern, einen Bytecode-Compiler und einige Tools, die die professionelle Entwicklung und den professionellen Betrieb von PHP stützten.

Zu etwa der gleichen Zeit fanden sich in Deutschland einige Freelancer zusammen, um ebenfalls den kommerziellen Einsatz von PHP zu stärken. Das sollte über Schulungen, über professionellen und verfügbaren Support und über gut ausgebildete Entwickler geschehen. Dem Thema entsprechend nannte sich dieser Zusammenschluss „ThinkPHP“.

Diesen Sprung in das kommerzielle Umfeld hat PHP geschafft, und die Sprache ist heute weiter verbreitet als die meisten anderen. Wer sich die Projektstatistiken auf Freshmeat oder SourceForge anschaut, hat den Eindruck, es mit der wichtigsten Programmiersprache überhaupt zu tun zu haben. Diese These wird auch durch den TIOBE Programming Com-

munity Index gestützt, der PHP im Januar 2008 als die viertwichtigste Programmiersprache überhaupt identifizierte – nach Java, C und (Visual) Basic.

Aber auch auf der anderen Seite sind die „Personal Homepage Tools“ ihren Wurzeln treu geblieben. PHP ist, um den Vater der Sprache zu zitieren, das „Basic der Neuzeit“. Wurde früher mit dem mageren „MICROSOFT BASIC V2.0“ auf dem Commodore 64 begonnen, so ist heute das Opfer der ersten Programmierversuche die Seite des eigenen Clans oder die Familienseite, die – natürlich – in PHP entwickelt wird, denn die Sprache ist bei nahezu jedem Provider verfügbar.

Gleichzeitig wird PHP in kritischen Environments, in Banken, in den größten Portalen (Yahoo!, studiVZ, Facebook usw.), als Basis von Shops, von Videoportalen, von Dokumenten- und Content Management-Systemen genutzt. Aber wie kann eine Sprache gleichzeitig für Einsteiger und für Enterprise-Systeme geeignet sein? Wie können Vierzehnjährige Software auf dieser Basis schreiben und gleichzeitig riesige Geldsummen in kurzer Zeit auf Systemen mit der gleichen Basis umgesetzt werden?

Die Antwort ist ganz einfach: Indem ein Methoden- und Technologieset genutzt wird, das genau dies erlaubt. Schließlich läuft noch heute jede dritte geschäftskritische Anwendung auf einer Cobol-Software, deren Coding-Stil bequem jede Objektorientierung, jeden Coding-Style und jedes Design-Pattern predated. Auch wenn es heute unmöglich scheint, ohne deren Einsatz auch nur einen Zehnzeiler verlässlich zu entwickeln, so laufen diese Lösungen doch auch heute noch verlässlich und ungestört.

Von Entwicklern anderer Sprachen wird heute gerne etwas hochnäsiger auf PHP herabgeschaut, weil ihnen eben eines dieser unverzichtbaren Features fehlt. Das erinnert uns an einen Ausspruch von Kristian Koehntopp: „PHP ist die BORG.“¹ Jede noch so fremde Technologie, die attraktiv und nützlich erscheint, wird letztlich doch adaptiert.

Und so finden sich im modernen PHP Konstrukte wie private Properties, Interfaces und Namespaces, standardisierte Dokumentation, und bei der Entwicklung wird das ganze Set etablierter Design Patterns ausgeschöpft. Zur Qualitätssicherung werden die gleichen Methoden wie in der Java-Enterprise-Welt eingesetzt – Unit-Tests, Coding-Style-Checks und Code-Metriken. Jede programmierte Zeile Code führt, sobald sie in das Versionsmanagement zurückgeflossen ist, zu einem umfangreichen Test auf Funktionalität, auf vollständige Dokumentation, auf einen korrekten Coding-Style; Copy & Paste-Code und Anti-Pattern wie God-Objekte werden automatisch erkannt. Wenn diese Tests durchlaufen sind, wird ein neues Paket auf den Test-Server gespielt, und mit einem von Selenium ferngesteuerten Browser wird dann sogleich die ganze Applikation durchlaufen, um deren Funktionsfähigkeit festzustellen. Ist auch dies erfolgreich, wird auf dem identisch zum Produktionsserver konfigurierten Beta-Stage ein umfassender Integrationstest durchgeführt, und mit seinem positiven Resultat wird schliesslich das neue Paket für den Produktionsbetrieb freigegeben.

¹ <http://blog.koehntopp.de/archives/848-Zehn-Jahre-PHP.html>

„Enterprise“ bedeutet dabei aber nicht nur Professionalisierung in den Entwicklungsmethoden, sondern auch eine deutliche Verbesserung in der Kommunikation mit anderen Systemen und mit dem Nutzer. Web Services, MashUps und RSS stehen allorts zur Verfügung, und mit AJAX beginnen auch Webapplikationen sich nach den Bedürfnissen des Nutzers und umgekehrt Nutzer sich nach den Bedürfnissen der Webapplikation zu richten.

Mit SOA, der service-orientierten Architektur, wurde aus dem „ein Prozess, eine CPU und ein Ablauf“-Paradigma eine Interaktion zahlreicher Services, die jeweils ihren Beitrag zu einem Gesamtergebn liefern, das dann schliesslich zum Klienten geliefert wird. Mit XMLHttpRequest wurde das Seitenparadigma durch eine Komponentensicht ähnlich wie bei Desktop-Anwendungen abgelöst, in der viel häufiger mit dem Server kommuniziert wird, aber auch jeweils viel weniger Information übertragen wird.

Im Vergleich zu anderen Programmierwelten ergibt die Änderungsfrequenz in der Webentwicklung bereits einen leisen Summtton, so dass der Entwickler niemals zu Ende gelernt hat, sondern sich ständig fortbilden muss. Für die schnellen Änderungen haben sich so Blogs, Planets und Foren etabliert.

Für die großen Änderungen haben Sie dieses Buch.

München und Würzburg, im Februar 2008

Johann-Peter Hartmann, Björn Schotte