

HANSER

Stefan Priebisch

PHP migrieren

Konzepte und Lösungen zur Migration von PHP-Anwendungen und
-Umgebungen

ISBN-10: 3-446-41394-4

ISBN-13: 978-3-446-41394-8

Inhaltsverzeichnis

Weitere Informationen oder Bestellungen unter
<http://www.hanser.de/978-3-446-41394-8>
sowie im Buchhandel.



Inhalt

1	Einführung.....	1
1.1	Eine kurze Geschichte der Internet-Zeit.....	1
1.2	PHP erblickt das Licht der Welt.....	3
1.3	PHP 5 und die große Migration.....	5
1.4	Blick in die Zukunft: PHP 6.....	7
2	Strategien.....	9
2.1	Never touch a running system.....	10
2.1.1	Systemumgebung.....	11
2.1.2	Programmcode.....	13
2.2	Immer die neueste Version einsetzen.....	16
2.3	Auf der grünen Wiese beginnen.....	18
2.3.1	Programmcode von Grund auf neu entwickeln.....	19
2.3.2	Die Systemumgebung von Grund auf neu aufbauen.....	20
2.4	Den richtigen Mittelweg finden.....	21
3	Aspekte einer Migration.....	23
3.1	Plattform.....	25
3.1.1	Architektur.....	25
3.1.2	Prozessor.....	26
3.1.3	Instruction Set.....	27
3.1.4	Wortlänge.....	28
3.1.5	Byte-Reihenfolge.....	30
3.1.6	Bauformen und Schnittstellen.....	31
3.2	Betriebssystem.....	32
3.2.1	Wortlänge.....	35
3.2.2	Zeilenendezeichen.....	35
3.2.3	Zugriffsrechte.....	38
3.2.4	Pfade und Dateinamen.....	40
3.2.5	Temporäre Dateien.....	43

3.2.6	Der Dateisuchpfad	44
3.2.7	Zeichensätze	44
3.3	Datenbank	47
3.3.1	SQL ist nicht gleich SQL.....	48
3.3.2	Programmcode in der Datenbank	49
3.3.3	Datentypen.....	51
3.3.4	Zeichensätze	51
3.3.5	Backup und Restore.....	53
3.4	Webserver	56
3.4.1	Apache und Apache2.....	57
3.4.2	Sicherheit.....	58
3.4.3	Apache übersetzen	60
3.4.4	Mehrere Webserver auf einem Rechner	61
3.5	PHP	62
3.5.1	Integration mit dem Webserver	62
3.5.2	PHP übersetzen	67
3.5.3	Thread-Modell.....	68
3.5.4	Die PHP-Konfiguration	69
3.5.5	PHP-Erweiterungen	87
3.5.6	Mehrere PHP-Versionen installieren	94
3.6	PHP-Code.....	94
3.6.1	Code von Drittanbietern	95
3.6.2	Eigener PHP-Code.....	96
3.7	Externe Programme.....	97
3.8	Schnittstellen zu Drittsystemen	100
3.9	Zeichenkodierungen	101
3.10	Browser	109
3.11	Weitere Aspekte	111
3.11.1	Harte Limits.....	111
3.11.2	Sicherheit.....	113
3.11.3	Systemkonzept.....	114
4	Die Migration vorbereiten	117
4.1	Der Istzustand.....	118
4.2	Das Zielsystem.....	121
4.3	Die Migration planen	124
5	Die Migration durchführen.....	129
5.1	Vorbereitungen.....	131
5.1.1	Das erste Testsystem	131
5.1.2	Das zweite Testsystem.....	132
5.2	Testen	133
5.2.1	Relevante Testfälle finden	134
5.2.2	Testdaten finden	135
5.2.3	Tests programmieren	136

5.3	Refactoring	138
5.3.1	Redundanten Code eliminieren	139
5.3.2	Codeblöcke kürzen	140
5.3.3	Unterschiedliche Belange trennen	140
5.4	Migrieren	142
5.4.1	Altlasten beseitigen	143
5.4.2	Module ersetzen	144
5.4.3	Syntaxfehler beseitigen	145
5.4.4	Alle PHP-Fehlermeldungen beseitigen	145
5.4.5	Logische Fehler beseitigen	146
5.4.6	Die PHP-Konfiguration normieren	146
5.5	Das Live-System migrieren	147
5.6	Die Migration abschließen	150
6	Nach der Migration	153
6.1	Gesammelte Erfahrungen	153
6.1.1	Modular programmieren	154
6.1.2	Coding Guidelines etablieren	154
6.1.3	Defensiv programmieren	154
6.1.4	Nicht der Erste sein	155
6.1.5	Laufendes Refactoring	156
6.1.6	Agile Migration	157
7	Werkzeuge	159
7.1	Versionsverwaltung	160
7.2	Kommandozeilenwerkzeuge	161
7.2.1	HTTP-Anfragen senden und Dateien herunterladen: wget	161
7.2.2	Dateien und Verzeichnisse durchsuchen: grep	163
7.2.3	Ersetzungen in Dateien vornehmen: sed	164
7.2.4	Dateien und Verzeichnissen vergleichen: diff-Werkzeuge	164
7.3	(X)HTML-Dateien validieren	166
7.3.1	Der W3C-Validator	168
7.3.2	HTML Tidy	170
7.3.3	Die PHP-Erweiterung Tidy	172
7.4	CSS-Dateien validieren	174
7.5	XML-Dateien validieren	176
7.5.1	xmllint	176
7.5.2	PHP	178
7.6	Statische Analyse von JavaScript-Dateien	178
7.6.1	jsl	179
7.6.2	JSLint	180
7.7	Firefox-Erweiterungen	182
7.7.1	Webdeveloper	182
7.7.2	XPather	184
7.7.3	Firebug	185

7.8	PHP-Bordmittel.....	187
7.8.1	PHP-Konfiguration.....	187
7.8.2	Syntaxprüfung.....	190
7.8.3	Prepend- und Append-Dateien.....	191
7.9	PEAR-Komponenten.....	192
7.9.1	PHP_Compat.....	194
7.9.2	PHP_Beautifier.....	195
7.9.3	PHP_CodeSniffer.....	199
7.9.4	PHP_CompatInfo.....	205
7.10	Virtuelle Maschinen.....	207
7.10.1	VMWare.....	208
7.10.2	Eine virtuelle Maschine installieren.....	209
7.10.3	Mit Snapshots arbeiten.....	211
7.11	Testwerkzeuge.....	212
7.11.1	Komponententests mit PHPUnit.....	213
7.11.2	Systemtests mit Selenium.....	216
7.12	Programmanalyse und Debugging.....	225
7.12.1	Installation.....	225
7.12.2	Nützliche Features.....	226
7.12.3	Tracing.....	226
7.12.4	Debugging.....	228
7.12.5	Codeabdeckung.....	231
7.13	Build-Automation.....	232
7.13.1	Installation.....	232
7.13.2	Codequalitätssicherung.....	233
7.13.3	Testautomation.....	235
7.13.4	Codeabdeckungsstatistiken.....	237
8	PHP-Code migrieren.....	241
8.1	Groß- und Kleinschreibung.....	242
8.1.1	Variablen.....	244
8.1.2	Konstanten.....	245
8.1.3	Magische Konstanten.....	248
8.1.4	Funktionen und Methoden.....	249
8.1.5	Klassen.....	250
8.1.6	Dateien.....	251
8.2	Namenskonflikte.....	252
8.2.1	Reservierte Schlüsselwörter.....	253
8.2.2	Funktionen.....	256
8.2.3	Klassen, Interfaces und Exceptions.....	257
8.2.4	Konstanten.....	260
8.2.5	Magische Konstanten, Funktionen und Methoden.....	263
8.2.6	Variablen.....	263
8.2.7	Komponenten und Bibliotheken.....	265

8.3	Verarbeitung von Eingabedaten	268
8.3.1	Globale Variablen registrieren.....	268
8.3.2	Lange Eingabe-Arrays	271
8.3.3	Superglobale Variablen.....	271
8.3.4	Magic Quotes.....	272
8.3.5	Auf POST-Daten zugreifen.....	276
8.3.6	Skriptname: \$PHP_SELF	277
8.4	Fehlerbehandlung	278
8.4.1	Fehler unterdrücken mit @	279
8.4.2	Die letzte Fehlermeldung speichern.....	280
8.4.3	Die Fehleranzeige konfigurieren.....	281
8.4.4	Neue und geänderte Fehlermeldungen.....	282
8.4.5	Eigene Fehler-Handler	285
8.4.6	Exceptions	287
8.5	Referenzen.....	287
8.5.1	Der PHP 4-Kompatibilitätsmodus	288
8.5.2	Eine Referenz erzeugen	289
8.5.3	Referenzen übergeben.....	290
8.5.4	Referenzen zurückgeben.....	292
8.5.5	Zur Laufzeit Referenzen übergeben.....	293
8.5.6	Objekte kopieren.....	294
8.6	Magische Konstanten	295
8.7	Geändertes Verhalten von PHP-Funktionen.....	297
8.7.1	array_merge()	297
8.7.2	ip2long()	299
8.7.3	strrpos().....	299
8.7.4	stripos()	300
8.7.5	strtotime()	300
8.8	Klassen	301
8.8.1	Statische Methoden und dynamische Aufrufe	301
8.8.2	Abstrakte private Methoden.....	302
8.8.3	Abstrakte statische Methoden	302
8.8.4	Geänderte Methodensignatur in abgeleiteten Klassen	303
8.9	Objekte.....	304
8.9.1	Konstruktor.....	304
8.9.2	Destruktoren	306
8.9.3	Klassenkonstanten neu definieren.....	307
8.9.4	instanceOf anstelle von is_a().....	307
8.9.5	Namenskonflikte mit \$this	308
8.9.6	\$this neu zuweisen	309
8.9.7	Objekte vergleichen	310
8.10	Dynamische Aufrufe	310
8.10.1	Systemaufrufe	311
8.10.2	Klassennamen	312
8.10.3	Die call_user_func()-Familie	313

8.10.4	Nachgeladener Code.....	315
8.10.5	eval	316
8.11	Kleine Einheiten.....	317
8.11.1	unset() und Zeichenketten.....	317
8.11.2	Fehler beim Senden der HTTP-Header.....	318
8.11.3	Datums- und Zeitfunktionen.....	320
8.11.4	Modulo-Division	320
8.11.5	Falsche Parameteranzahl in Funktionsaufrufen	321
8.11.6	Automatisches Umwandeln von Integer-Werten.....	323
8.11.7	Leere Objekte	324
8.11.8	\$this, Delegation und statische Aufrufe.....	324
8.11.9	Objekte ausgeben und die magische Methode __toString().....	326
8.12	PHP-Erweiterungen.....	326
8.12.1	mysql und mysqli	327
8.12.2	SPL.....	328
8.12.3	Tidy	328
8.12.4	Tokenizer.....	328
8.12.5	XML.....	328
	Literatur	331
	Register	335