



Martin Schütte

6. Brandenburger Linux-Infotag

21. November 2009

# Was ist PEAR?

## *PHP Extension and Application Repository*

- Bibliothek offenen PHP-Quellcodes
- PHP Coding Standards
- Infrastruktur

*PEAR's mission is to provide reusable components, lead innovation in PHP, provide best practices for PHP development and educate developers.*

# Was ist PEAR?

PEAR ist Meta-Projekt für viele unabhängige *packages*.

Gemeinsamkeiten:

- Anforderungen an Pakete
- Website, SVN, Bugtracker, Mailinglisten
- Installer, Download-Mirrors
- Verfahrensregeln, Qualitätssicherung

# Entwicklergemeinde

- Keine zuverlässige Zählung, nur Indizien aus Statistik
- ca. 750 Entwickler
- 314 User haben  $\geq 1$  Bug geschlossen
- 109 User haben  $\geq 10$  Bugs geschlossen
- 16 User haben  $\geq 100$  Bugs geschlossen
- ohloh.net zählt 87 "Active Contributors" im SVN

# Governance

Institutionen nach Verfassung von März 2007:

- PEAR Group
  - President
  - Organisation
  - Konfliktlösung
- QA Team
  - Beseitigung ernster Bugs
  - Verwaltung verwaister Pakete
- Website Administrators
- Developer/Maintainer

# Governance

Institutionen nach Verfassung von März 2007:

- PEAR Group
  - President
  - Organisation
  - Konfliktlösung
- QA Team
  - Beseitigung ernster Bugs
  - Verwaltung verwaister Pakete
- Website Administrators
- Developer/Maintainer

# QA Team

- Anlaufstelle bei Qualitätsproblemen
- Verwaltung der Pakete ohne Maintainer
- Dürfen
  - Bugs beseitigen
  - Releases veröffentlichen
  - verwaiste Pakete finden
  - Releases löschen

# PEPr

## PEAR Proposal system

- Entscheidungsprozess für neue Pakete oder Regeln
- Diskussion per E-Mail, formale Schritte auf Website
- “Draft” → “Proposal” → “Call for Votes” → “Finished”
- Abstimmung mit +1/0/-1
- Kommentaren und Bedingungen möglich
- $\Sigma \geq 5$  nötig zur Annahme des Vorschlags

Beispiele: Net\_FastCGI, “Prefix all classes with PEAR\_”



# Anforderungen an ein neues Paket

- unterliegt akzeptierter freier Lizenz (PHP, Apache, LGPL, BSD, MIT)
- erfüllt Coding Standards
- liegt in öffentlicher Versionsverwaltung
- ist nicht-redundant, erweiterbar, objektorientiert
- umfasst Dokumentation
- umfasst Tests
- wird vom Entwickler gepflegt (support & bugfixes)

# Hilfsprogramme

- PHPdoc: automatische API Dokumentation
- PHPUnit: Unit- und Coverage-Tests
- PHP\_CodeSniffer: Quelltext-Formatierung
- PEAR\_PackageFileManager2: Hilfe für `package.xml`
- PEAR package manager: Installation und Paketverwaltung

# PHPdoc

„Javadoc für PHP“ –  
generiert API Dokumentation aus Kommentaren

Beispiel:

```
/**  
 * Checks if the given string contains only valid chars.  
 *  
 * @param string $string String that is checked.  
 *  
 * @access public  
 * @return boolean  
 */  
function validString($string)
```

# PHPUnit

## Unit- und Coverage-Tests

Beispiel:

```
public function testValidStringFalse1()
{
    $result = $this->fixture->validString("ä");
    $this->assertFalse($result);
}
```

# PHP\_CodeSniffer

Prüft ob Quelltext dem Coding Standards entspricht.

Beispiel:

```
[mschuett@dagny] ~> phpcs .../DTA.php
```

```
FILE: /media/Shared/PEAR/Payment_DTA-1.2.0/DTA.php
```

```
-----  
FOUND 55 ERROR(S) AND 34 WARNING(S) AFFECTING 73 LINE(S)  
-----
```

```
 2 | ERROR   | You must use "/*" style comments for a file comment  
71 | ERROR   | @author tag comment indented incorrectly. Expected 2 spaces  
   |         | but found 3.  
72 | WARNING | Invalid version "$Revision: 0.81 $" in doc comment; consider  
   |         | "Release: <package_version>" instead  
72 | ERROR   | @version tag comment indented incorrectly. Expected 1 spaces  
   |         | but found 2.  
73 | ERROR   | @since tag comment indented incorrectly. Expected 3 spaces but  
   |         | found 4.  
74 | ERROR   | Missing @category tag in class comment  
74 | ERROR   | Missing @package tag in class comment  
74 | ERROR   | Missing @license tag in class comment  
74 | ERROR   | Missing @link tag in class comment  
75 | ERROR   | Opening brace of a class must be on the line after the  
   |         | definition  
116 | WARNING | Line exceeds 85 characters; contains 109 characters
```

```
...
```

# PEAR\_PackageFileManager2

Aktualisiert die package.xml.

Beispiel:

```
<?php
require_once('PEAR/PackageFileManager2.php');
$packagexml = new PEAR_PackageFileManager2;

$packagexml = $packagexml->importOptions('./package.xml', ...);

$packagexml->setReleaseVersion('1.3.2');
$packagexml->setReleaseStability('stable');
$packagexml->setPhpDep('4.0.6');
$packagexml->setNotes('Hallo Welt!');

$packagexml->generateContents();
$packagexml->writePackageFile();
```

# PEAR package manager

## Installation und Paketverwaltung.

### Beispiel:

```
[mschuett@dagny] ~> sudo pear channel-discover pear.symfony-project.com
Adding Channel "pear.symfony-project.com" succeeded
Discovery of channel "pear.symfony-project.com" succeeded
```

```
[mschuett@dagny] ~> pear list-upgrades
pear.php.net Available Upgrades (stable):
```

```
=====
Channel      Package      Local          Remote          Size
pear.php.net Structures_Graph 1.0.2 (stable) 1.0.3 (stable) 30kB
```

```
pear.phpunit.de Available Upgrades (stable):
```

```
=====
Channel      Package Local          Remote          Size
pear.phpunit.de PHPUnit 3.3.17 (stable) 3.4.2 (stable) 320kB
```

```
Channel pear.symfony-project.com: No upgrades available
```

# Release veröffentlichen

als Maintainer

1. `php tests/AllTests.php`
2. `phpdoc ...`
3. `package.xml` aktualisieren  
(Release Notes, Dateiliste, Abhängigkeiten etc.)
4. `pear package-validate`
5. `pear package`  
(liest `package.xml` und erstellt `.tar.gz`-Archiv)
6. optional: `pear sign` ergänzt PGP-Signatur
7. Hochladen auf `http://pear.php.net`



# Release installieren

als Benutzer

```
1. pear install symfony/symfony
```

*oder:*

```
1. pear upgrade Archive_Tar
```

*oder:*

```
1. pear upgrade-all
```

# PEAR2

jetzt in Arbeit

- neuer Installer: Pyrus
- neue Paketdateien: phar (PHP Archive)
- PHP Version 5.3+
- nur noch ein SVN
- aktualisierte Coding Standards und Policies

# Ressourcen

- Website: <http://pear.php.net>
- Wiki: <http://wiki.pear.php.net/>
- Mailinglisten:
  - PEAR-general
  - PEAR-developers
  - PEAR-cvs
  - PEAR-documentation
  - PEAR-QA
  - PEAR-core
  - PEAR-website
- IRC: #pear im EFnet
- SVN: PHP-SVN auf [SVN.php.net](http://SVN.php.net)