

Software- qualität in PHP-Prozessen

Installation und Betrieb eines
Jenkins-Servers

René Schröder

Basissystem auf den aktuellen Stand zu bringen und alle notwendigen Updates und Patches zu installieren:

```
aptitude update  
aptitude full -upgrade
```

Das Basissystem ist nun auf einem aktuellen Stand, was uns die Möglichkeit eröffnet, die ersten Pakete zu installieren, die für Jenkins nötig sind. Befolgen Sie hierzu die folgenden Befehle, die den Webserver Apache und PHP installieren:

```
aptitude install apache2  
aptitude install php5  
aptitude install php5-xdebug  
aptitude install php5-xsl  
aptitude install php5-intl
```

Apache und PHP sind hiermit korrekt installiert. Um zu überprüfen, ob der Apache-Webserver korrekt läuft, rufen Sie in Ihrem Browser die IP-Adresse des Jenkins-Servers auf. Ist alles korrekt installiert, wird hier die Testseite vom Apache-Server angezeigt. Mit den folgenden Befehlen installieren wir die Grundpakete für einen Standard-Build-Server:

```
aptitude install ant
aptitude install java7 -jdk
aptitude install subversion
aptitude install git
```

Jenkins aufsetzen

Nach erfolgreichem Aufsetzen der Standardkomponenten für Jenkins

kümmern wir uns nun um die entsprechende Umgebung für PHP. Vor einiger Zeit war es gängig, die PHP-Toolchain via PEAR Installer zu installieren und zu pflegen. PEAR war/ist ein Manager und ein Repository für PHP-Pakete und -Erweiterungen. Im Laufe der Zeit kamen neue und bessere Werkzeuge hierfür zum Einsatz. Daher möchte ich hier nicht weiter die Installation der Toolchain mit PEAR sondern mit Composer erläutern.

Composer ist ähnlich oder genauso wie PEAR ein Tool, mit dem sich Abhängigkeiten von PHP-Paketen abbilden lassen. Erweiterungen und

Bibliotheken sind hiermit komfortabel zu installieren. Hierfür installieren wir zuerst Composer, am besten global:

```
curl -sS https://getcomposer.org/installer |  
mv composer.phar /usr/local/bin/composer
```

Nach dem Ausführen von Punkt 2 ist Composer in der Regel global verfügbar. Ist das nicht der Fall, ist die PATH-Variable in `/etc/environments/`, um den obigen Pfad zu erweitern.

Die PHP-Toolchain

Die PHP-Toolchain umfasst Tools, die es uns ermöglichen, die Qualität von PHP-Software zu verbessern und zu überwachen. Angefangen von

statischer Codeanalyse über Codestyleanalyse bis hin zu Unit Testing und einigem mehr, werden viele Bereiche abgedeckt, die für Softwarequalität wichtig sind.

PHPUnit

Mittels PHPUnit werden, wie auch in anderen Sprachen gängig, Unit Tests erstellt, die den Code – die Units – bei jedem Build automatisch testen. Unit Tests sind ein wichtiger Bestandteil zur Sicherstellung der PHP-Softwarequalität:

```
composer global require 'phpunit/phpunit = 4
```

PHP_CodeSniffer