



Integrate Your PHP Project with Jenkins

Sebastian Bergmann

OSCON – July 27th 2011





HELLO
my name is
Sebastian



Crucial Practices of Agile Programmers

- Continuous Integration
- Test-Driven Development
- Constant Design Improvement
- Coding Standard
- Collective Code Ownership
- Simple Design
- System Metaphor
- Pair Programming



What this means in practice ...

- Original authorship is immaterial
- Anyone can modify any code at any time
- The team adheres to a coding standard
- Abundant automated tests create confidence
- Version control provides insurance



Continuous Integration

For each change set

- run the tests
- analyse the source code
- generate documentation
- build a package
- ...





sharing experience



The Value of Continuous Integration

- Reduce risks
- Reduce repetitive processes
- Prevent low internal software quality
- Generate deployable software
- Enable better project visibility
- Establish greater project confidence



Prerequisites for Continuous Integration

- Build Automation
- Software Configuration Management



Directory Structure of our Example Project

```
.
|-- build
|   |-- phpcs.xml
|   |-- phpmd.xml
|   |-- src_autoload.php.in
|   `-- tests_autoload.php.in
|-- build.xml
|-- phpunit.xml.dist
|-- README.markdown
|-- src
|   |-- autoload.php
|   `-- ...
`-- tests
    |-- autoload.php
    `-- ...
```



Apache Ant build script that invokes PHPAB

```
<?xml version="1.0" encoding="UTF-8"?>

<project name="BankAccount" default="build">
  <target name="build" depends="phpab" />

  <target name="phpab" description="Generate autoloader scripts">
    <exec executable="phpab">
      <arg value="--output" />
      <arg path="${basedir}/src/autoload.php" />
      <arg value="--template" />
      <arg path="${basedir}/build/src/autoload.php.in" />
      <arg path="${basedir}/src" />
    </exec>

    <exec executable="phpab">
      <arg value="--output" />
      <arg path="${basedir}/tests/autoload.php" />
      <arg value="--template" />
      <arg path="${basedir}/build/tests/autoload.php.in" />
      <arg path="${basedir}/tests" />
    </exec>
  </target>
</project>
```



Apache Ant build script that invokes PHPAB

```
> ant
```

```
Buildfile: /home/sb/bankaccount/build.xml
```

```
phpab:
```

```
[exec] Autoload file 'src/autoload.php' generated.
```

```
[exec]
```

```
[exec] Autoload file 'tests/autoload.php' generated.
```

```
[exec]
```

```
build:
```

```
BUILD SUCCESSFUL
```

```
Total time: 0 seconds
```



Apache Ant build script that invokes PHPUnit

```
<?xml version="1.0" encoding="UTF-8"?>

<project name="BankAccount" default="build">
  <target name="build" depends="prepare,phpunit" />

  <target name="clean" description="Cleanup build artifacts">
    <delete dir="${basedir}/build/coverage" />
    <delete dir="${basedir}/build/logs" />
  </target>

  <target name="prepare" depends="clean,phpab"
    description="Prepare for build">
    <mkdir dir="${basedir}/build/coverage" />
    <mkdir dir="${basedir}/build/logs" />
  </target>

  <target name="phpab" description="Generate autoloader scripts">
    <!-- ... -->
  </target>

  <target name="phpunit" description="Run unit tests with PHPUnit">
    <exec executable="phpunit" failonerror="true" />
  </target>
</project>
```



PHPUnit Configuration

```
<?xml version="1.0" encoding="UTF-8"?>

<phpunit bootstrap="tests/autoload.php" backupGlobals="false"
    backupStaticAttributes="false" strict="true" verbose="true">
    <testsuites>
        <testsuite name="BankAccount">
            <directory suffix="Test.php">tests/unit</directory>
        </testsuite>
    </testsuites>
    <logging>
        <log type="coverage-clover" target="build/logs/clover.xml" />
        <log type="coverage-html" target="build/coverage"
            title="BankAccount" />
        <log type="junit" target="build/logs/junit.xml" />
    </logging>
    <filter>
        <whitelist addUncoveredFilesFromWhitelist="true">
            <directory suffix=".php">src</directory>
            <exclude>
                <file>src/autoload.php</file>
            </exclude>
        </whitelist>
    </filter>
</phpunit>
```



Apache Ant build script that invokes PHPUnit

```
> ant
Buildfile: /home/sb/bankaccount/build.xml

clean:
  [delete] Deleting directory /home/sb/bankaccount/build/coverage
  [delete] Deleting directory /home/sb/bankaccount/build/logs

phpab:
  [exec] Autoload file '/home/sb/bankaccount/src/autoload.php' generated.
  [exec] Autoload file '/home/sb/bankaccount/tests/autoload.php' generated.

prepare:
  [mkdir] Created dir: /home/sb/bankaccount/build/coverage
  [mkdir] Created dir: /home/sb/bankaccount/build/logs

phpunit:
  [exec] PHPUnit 3.5.14 by Sebastian Bergmann.
  [exec]
  [exec] .....
  [exec]
  [exec] Time: 0 seconds, Memory: 9.50Mb
  [exec]
  [exec] OK (39 tests, 69 assertions)
  [exec]
  [exec] Writing code coverage data to XML file, this may take a moment.
  [exec] Generating code coverage report, this may take a moment.

build:

BUILD SUCCESSFUL
Total time: 2 seconds
```



Continuous Integration of PHP Projects

- Jenkins

<http://jenkins-ci.org/>

- Template for Jenkins Jobs for PHP Projects

– <http://jenkins-php.org/>

- PHP Project Wizard (PPW)

– On GitHub, linked from <http://jenkins-php.org/>



Installing Jenkins

```
> mkdir /usr/local/jenkins  
> cd /usr/local/jenkins  
> wget http://mirrors.jenkins-ci.org/war-stable/latest/jenkins.war  
> export JENKINS_HOME=/usr/local/jenkins  
> java -jar jenkins.war
```



Installing Plugin for Jenkins (using Jenkins CLI)

- > `wget http://localhost:8080/jnlpJars/jenkins-cli.jar`
- > `java -jar jenkins-cli.jar -s http://localhost:8080 install-plugin checkstyle`
- > `java -jar jenkins-cli.jar -s http://localhost:8080 install-plugin cloverphp`
- > `java -jar jenkins-cli.jar -s http://localhost:8080 install-plugin dry`
- > `java -jar jenkins-cli.jar -s http://localhost:8080 install-plugin htmlpublisher`
- > `java -jar jenkins-cli.jar -s http://localhost:8080 install-plugin jdepend`
- > `java -jar jenkins-cli.jar -s http://localhost:8080 install-plugin plot`
- > `java -jar jenkins-cli.jar -s http://localhost:8080 install-plugin pmd`
- > `java -jar jenkins-cli.jar -s http://localhost:8080 install-plugin violations`
- > `java -jar jenkins-cli.jar -s http://localhost:8080 install-plugin xunit`

- > `java -jar jenkins-cli.jar -s http://localhost:8080 install-plugin git`
- > `java -jar jenkins-cli.jar -s http://localhost:8080 install-plugin greenballs`

- > `java -jar jenkins-cli.jar -s http://localhost:8080 safe-restart`



Continuous Integration of PHP Projects



Demo Time!

The End

- Web: <http://thePHP.cc/>
<http://sebastian-bergmann.de/>
- Mail: sebastian@thePHP.cc
- Twitter: [@s_bergmann](https://twitter.com/s_bergmann)
- Slides: <http://talks.thePHP.cc/>
- Book: <http://oreilly.com/catalog/0636920021353/>

