

# **statische Codeanalyse**

**WTF?!**

# WTF?!

```
hunk ./lib/Foo/Bar.php 13
-$response = Response::singleton();
+
+ // decide what decorator to use
+ if (isset($_GET['__env'])) {
+ $response = Response::singleton();
+ switch ($_GET['__env']) {
+ case 'popup':
+ case 'slide':
+ $response->setBaseDecorator('XHTML_Base_Slide');
+ break;
+ case 'profilepopup':
+ $response->setBaseDecorator('XHTML_Base_ProfilePopup');
+ break;
+ case 'frame':
+ $response->setBaseDecorator('XHTML_Base_Frame');
+ break;
+ case 'dialogue':
+ $response->setBaseDecorator('XHTML_Base_Dialogue');
+ break;
+ case 'json':
+ //verify if json Request is valid
+ $auth = Config::get('auth')->json;
+ $sid = $_COOKIE[Config::get('cfg')->session_name];
+ if ($this->route['json_access'] !== true && ($sid.$auth[$_GET['__app']] !== $foo))
+ die(118n('valid API-Key needed'));
+
+ $response->setBaseDecorator('JSON');
+ break;
+ }
+ }
```

# WTF?!

```
hunk ./lib/Foo/Bar.php 13
-$response = Response::singleton();
+
+ // decide what decorator to use
+ if (isset($_GET['__env'])) {
+ $response = Response::singleton();
+ switch ($_GET['__env']) {
+ case 'popup':
+ case 'slide':
+ $response->setBaseDecorator('XHTML_Base_Slide');
+ break;
+ case 'profilepopup':
+ $response->setBaseDecorator('XHTML_Base_ProfilePopup');
+ break;
+ case 'frame':
+ $response->setBaseDecorator('XHTML_Base_Frame');
+ break;
+ case 'dialogue':
+ $response->setBaseDecorator('XHTML_Base_Dialogue');
+ break;
+ case 'json':
+ //verify if json Request is valid
+ $auth = Config::get('auth')->json;
+ $sid = $_COOKIE[Config::get('cfg')->session_name];
+ if ($this->route['json_access'] !== true && ($sid.$auth[$_GET['__app']] !== $foo)) {
+ die(118n('valid API-Key needed'));
+ }
+ $response->setBaseDecorator('JSON');
+ break;
+ }
+ }
```

# Statische Codeanalyse

- man glotzt auf den Code...
  - stupide und repetitiv
  - auf Dauer ermüdend
  - sehr Fehleranfällig
  - Ergebnis fragwürdig!



# Statische Codeanalyse

- Codeanalyse automatisieren...
  - Computer ermüden nicht
  - Ergebnis nicht von Tagesform abhängig
  - nichts wird übersehen



# PHP\_CodeSniffer



- PEAR-Packet
- auf intra1 installiert (`$> phpcs --help`)
- unterstützt verschiedene Coding-Standarts
  - Zend, Squiz, PEAR ...
  - `$> phpcs -i`
- erweitert den PHP-Tokenizer
- ASB kann mehr als Reg-Ex/strpos
- kann Checkstyle

# Beispiele

- For loop with test function sniff:

```
class Foo {  
    public function bar($x) {  
        $a = array(1, 2, 3, 4);  
  
        for ($i = 0; $i < count($a); $i++) {  
            $a[$i] *= $i;  
        }  
    }  
}
```



# Beispiele

- Jumbled incrementor sniff:

```
class Foo {  
    public function bar($x) {  
        for ($i = 0; $i < 10; $i++) {  
            for ($k = 0; $k < 20; $i++) {  
                echo 'Hello';  
            }  
        }  
    }  
}
```

# Beispiele

- duplicate class name sniff
- empty statement sniff
- unnecessary final modifier sniff
- unused function parameter sniff
- useless overriding method sniff
- line endings sniff
- forbidden functions sniff
- ...

# Erweiterbar

- man kann eigene Standards definieren
  - Klassenhierarchie innerhalb PEAR :-(
  - ev. eigenes PEAR-Paket bauen
  - soll refactored werden laut Maintainer
- Kwick-Standard -> WIP
  - Coding-Guidelines (Wiki)
  - Performance-Guidelines (Wiki)
  - Benchmarks (/repos/tools)

# Continuous Integration

- frühzeitig und oft Änderungen in die Versionsverwaltung einchecken
- große Änderung inkrementell durch funktionsfähige kleine Änderungen
- jeder mindestens täglich seine Änderungen einchecken

# Continuous Integration

- nach jedem einchecken Gesamtsystem neu bauen und testen
- Entwickler frühstmöglich auf Integrationsprobleme hinweisen

Mit der Hand am Arm?

Mit der Hand am Arm?

**NEIN!**

# Hudson ist unser Buttler



- CI Server ([hudson-ci.org](http://hudson-ci.org))
- Java-Server
  - Winstone Servlet-Container
- voll webbasiert
- führt periodisch Builds aus
  - Ant
  - Maven
  - Phing < our Friend



# Phing - The Build

- Build-Tool Ant-Like
- XML (`/repos/tools/buil.xml`) mit Targets
  - Unittets
  - Code-Coverage
  - Apidoc
  - Checkstyle
  - ...
- [phing.info](http://phing.info)

# Meet the Buttler

## Hudson

search



sxs | Abmelden

Hudson » VClasses

AUTO-AKTUALISIERUNG AUSSCHALTEN

[Zurück zur Übersicht](#)

[Status](#)

[Änderungen](#)

[Arbeitsbereich](#)

[Jetzt bauen](#)

[Löschen](#)

[Konfigurieren](#)

[Checkstyle Warnungen](#)

[Dokumentation](#)

[Clover HTML Coverage Report](#)

[Git Abfrage-Protokoll](#)

## Projekt VClasses

Development-Build aus meine lokalen Repo.

[Beschreibung bearbeiten](#)



[Dokumentation](#)



[Clover HTML Coverage Report](#)



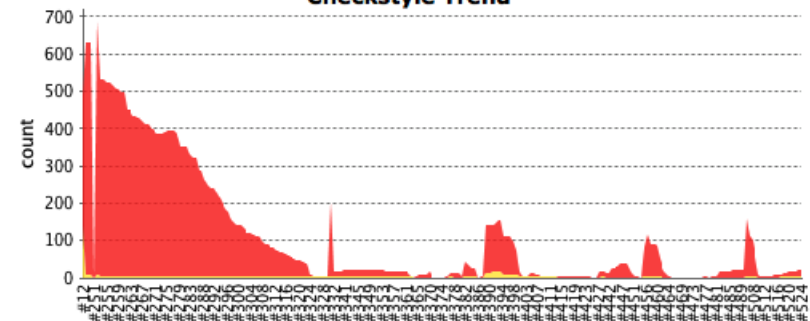
[Arbeitsbereich](#)



[Letzte Änderungen](#)

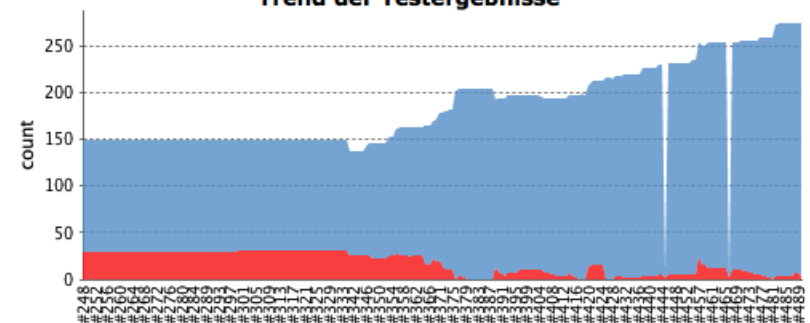
Disk Usage: Workspace 6MB, Builds 882MB

### Checkstyle Trend



[Konfigurieren...](#)

### Trend der Testergebnisse



[\(Nur Fehlschläge anzeigen\)](#) [Vergrößern](#)

**Code Coverage - 91.8% (1925/2098 elements)**

Methods 93.7% Conditionals 100% Statements 91.4%

## Permalinks

- [Letzter Build \(#524\), vor 1 Tag 19 Stunden](#)
- [Letzter stabiler Build \(#481\), vor 16 Tage](#)
- [Letzter erfolgreicher Build \(#490\), vor 14 Tage](#)
- [Letzter fehlgeschlagener Build \(#524\), vor 1 Tag 19 Stunden](#)

Build-Verlauf	(Trend)
#524 09.02.2010 23:04:27	
#523 09.02.2010 23:00:26	-
#522 09.02.2010 22:56:11	5MB
#521 09.02.2010 22:42:21	5MB
#520 09.02.2010 22:36:11	5MB
#519 09.02.2010 21:31:11	5MB
#518 09.02.2010 21:06:11	5MB
#517 09.02.2010 20:26:11	5MB
#516 09.02.2010 20:04:11	5MB
#515 08.02.2010 19:01:11	5MB
#514 08.02.2010 00:39:19	5MB
#513 08.02.2010 00:08:09	5MB
#512 08.02.2010 00:03:01	5MB
#511 08.02.2010 00:00:00	5MB

**<http://intra1.rz.kwick.de:8080/>**