
Suivi de qualité - PIC
Plateforme d’Intégration Continue

Gabriele Santini

Architecte/Consultant chez SQLI

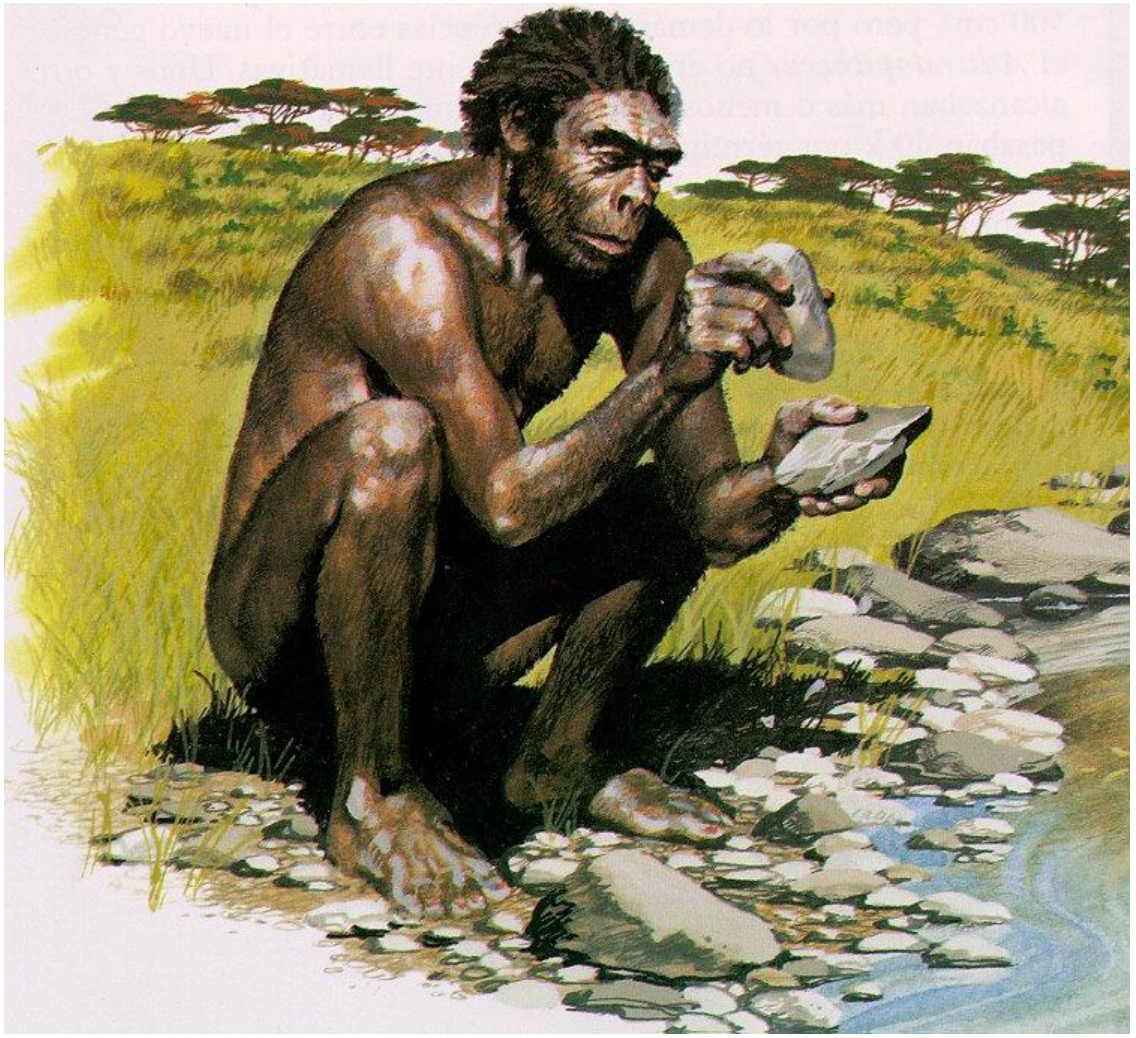
Responsable PIC-PHP-SQLI

Plugin Sonar PHP

afup

FORUM PHP 2010

Avant le suivi qualité

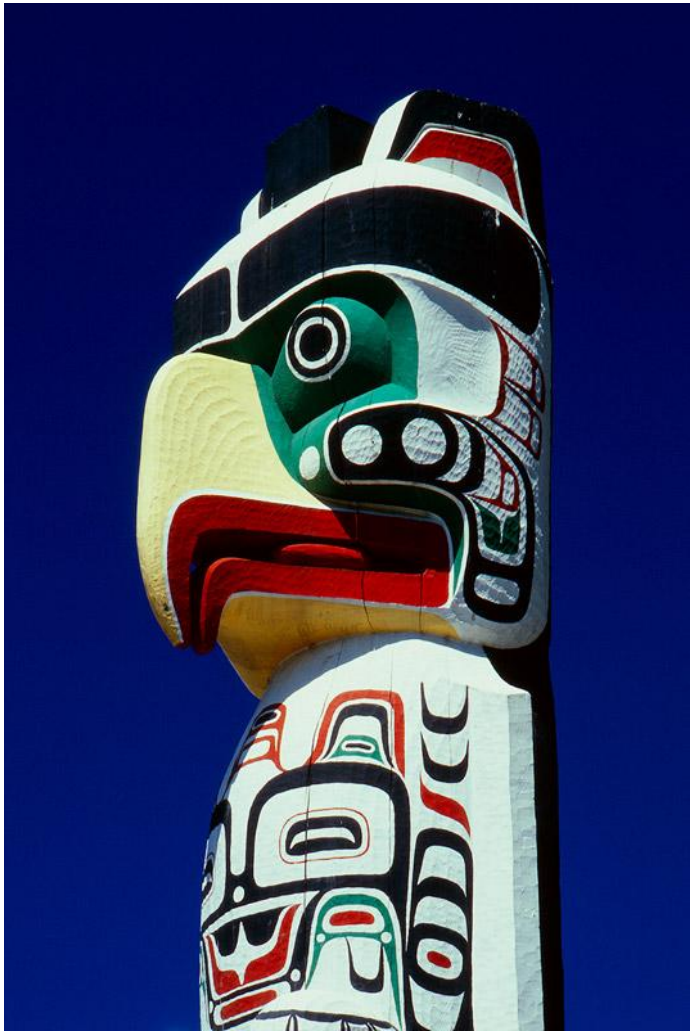


Effet « tunnel »

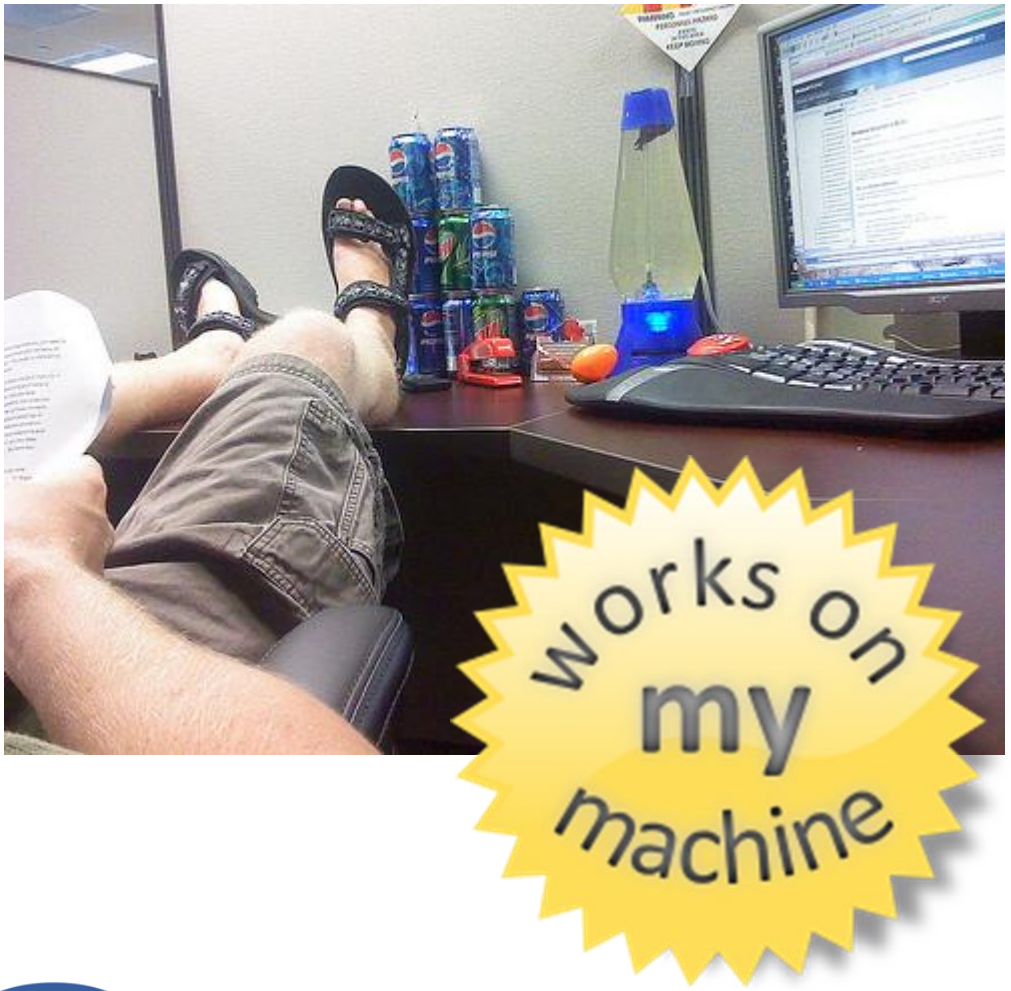
… bon voyage !

Code « totémisé »

« Code qui marche tu ne toucheras point ! »



Problèmes de déploiement



Problèmes de maintenance

Enter : Intégration Continue!

Automatise la procédure d'intégration

- l'étape de construction est appelée ***build***

A chaque build :

- Exécute les tests automatisés
- Analyse la qualité du produit
- Remonte des indicateurs (n°1 : ***état du build***)

Continue = build à chaque commit

Objectifs

- Projet stable et déployable
- Informations immédiates pour l'équipe de développement
- Indicateurs continus pour la gestion du projet

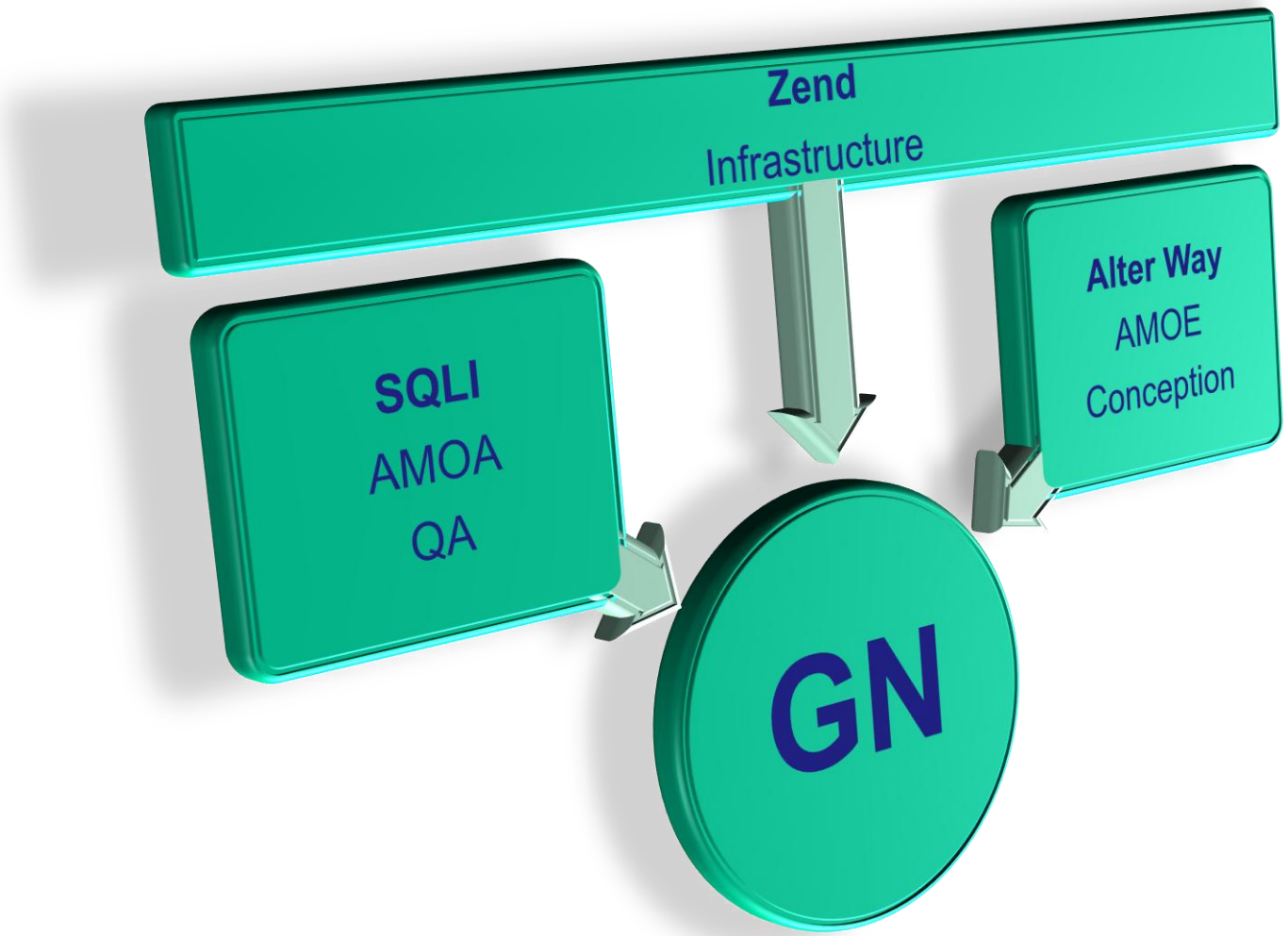
Le projet GN Puls@r

Projet d'envergure Gendarmerie Nationale

- Le système de gestion des brigades
- Gestion des « Formules », services, statistiques, …

- V1 en Java (2005-2007)
  - Problèmes de montée en charge
  - Fonctionnel figé
  - Non maintenable en interne

Puls@r V2 : PHP

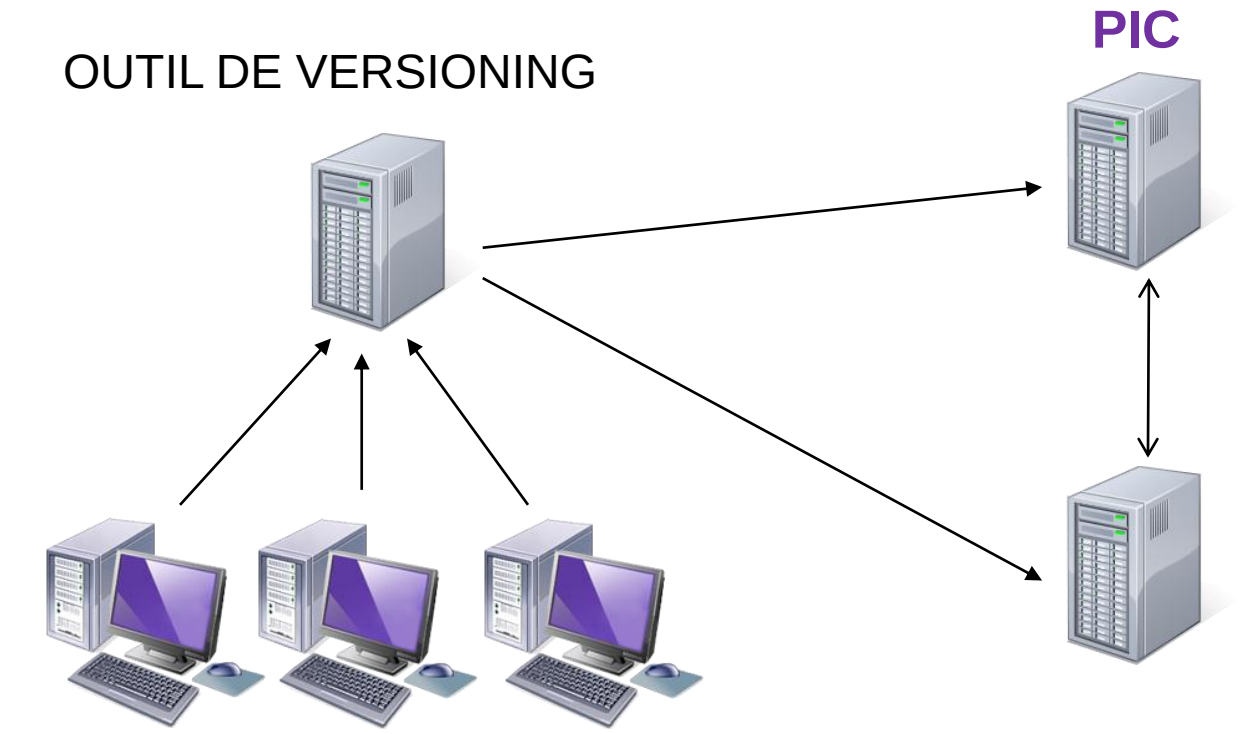

Zend
Infrastructure
SQLI
AMOA
QA
Alter Way
AMOE
Conception
GN

SQLI : suivi qualité

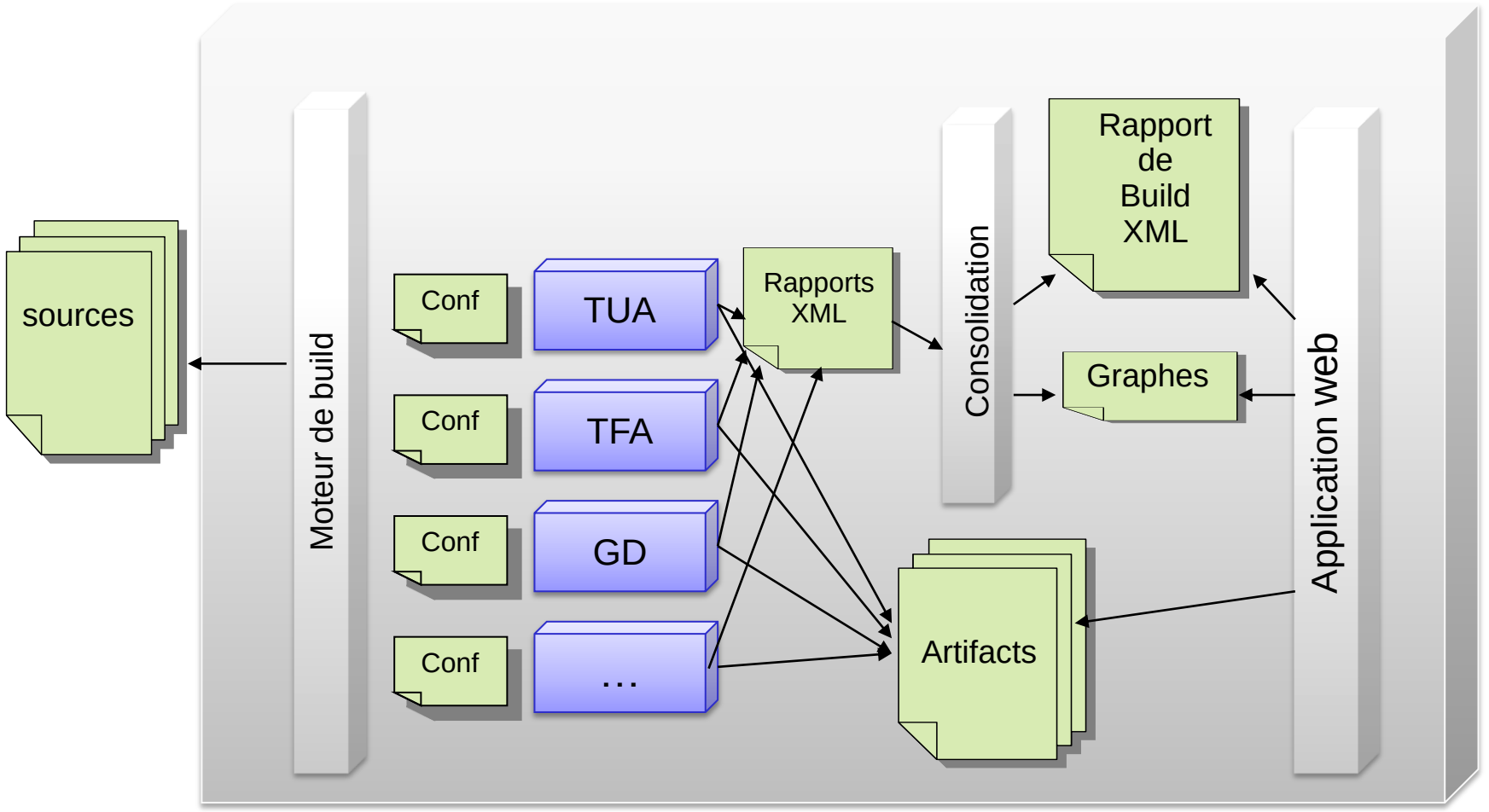
Mise en place d'une PIC pour le projet

Garantir la qualité des développements

PIC dans l'usine logicielle



PIC - Schéma


sources
Moteur de build
Conf
TUA
Conf
TFA
Conf
GD
Conf
…
Rapports XML
Consolidation
Rapport de Build XML
Graphes
Artifacts
Application web

PIC-PHP-SQLI

Fondée sur phpUnderControl

Qui se « greffe » sur CruiseControl,

moteur d'IC Java très stable

phpUnderControl « patche » pour php

le *build* et la *restitution*

Consolide les résultats (stats, graphes)

Outils dans PUC

phpUnderControl intègre :

- PHPUnit pour les **tests unitaires et fonctionnels automatisés**
- PHPUnit + Xdebug pour la **couverture de code**
- PHP_CodeSniffer pour la **vérification des règles de codage** (checkstyle)
- PHPDocumentor pour la **gestion de la documentation**
- PHPUnit (PHPMD) pour les **métriques projet** (PMD)

PHP_Documentor

Génération de la documentation

- Travail collaboratif de l'équipe de développement
- Les IDEs l'utilisent pour l'autocomplétion

phpDocumentor :

- Utilise et définit le standard de documentation **phpDoc**
- Artifact : site documentaire HTML
- Reporting de build avec les *violations de documentation*

PHP_CodeSniffer

Garantir le respect des règles choisies

- Diminution du nombre d'anomalies
- Indicateur fin sur la qualité globale de l'application

PHP_CodeSniffer :

- Solution OS de référence pour PHP
- Traite aussi Javascript et CSS
- Rapport XML en build
- Indicateur intégré dans les graphes phpUnderControl

PHP_Unit : TUA

Exécution des Tests Automatisés

- Pivot de l'intégration continue : non régression

PhpUnit :

- Projet mûr et parfaitement intégré dans l'univers des outils php
- Vaste panel de fonctionnalités (fixtures, mock objects, génération de squelettes, tests DB, ...)
- Exécute les TUA et fournit un rapport détaillé ainsi qu'un rapport d'état

PHP_Unit + Xdebug : CC

- Couverture de code
  - Quelles parties du code sont couvertes par les tests
  - Indicateur de l'efficacité des tests
- PhpUnit avec Xdebug :
  - Collecte les informations de couverture de code pendant l'exécution
  - Artifact : site HTML avec statistiques par dossier et détails par fichier
  - Fournit un rapport détaillé consolidé dans des indicateurs et des violations PMD

PHP_Unit 3.2-3.X : PMD

Calcul des métriques projet

- Remonté des violations liées aux valeurs anormales

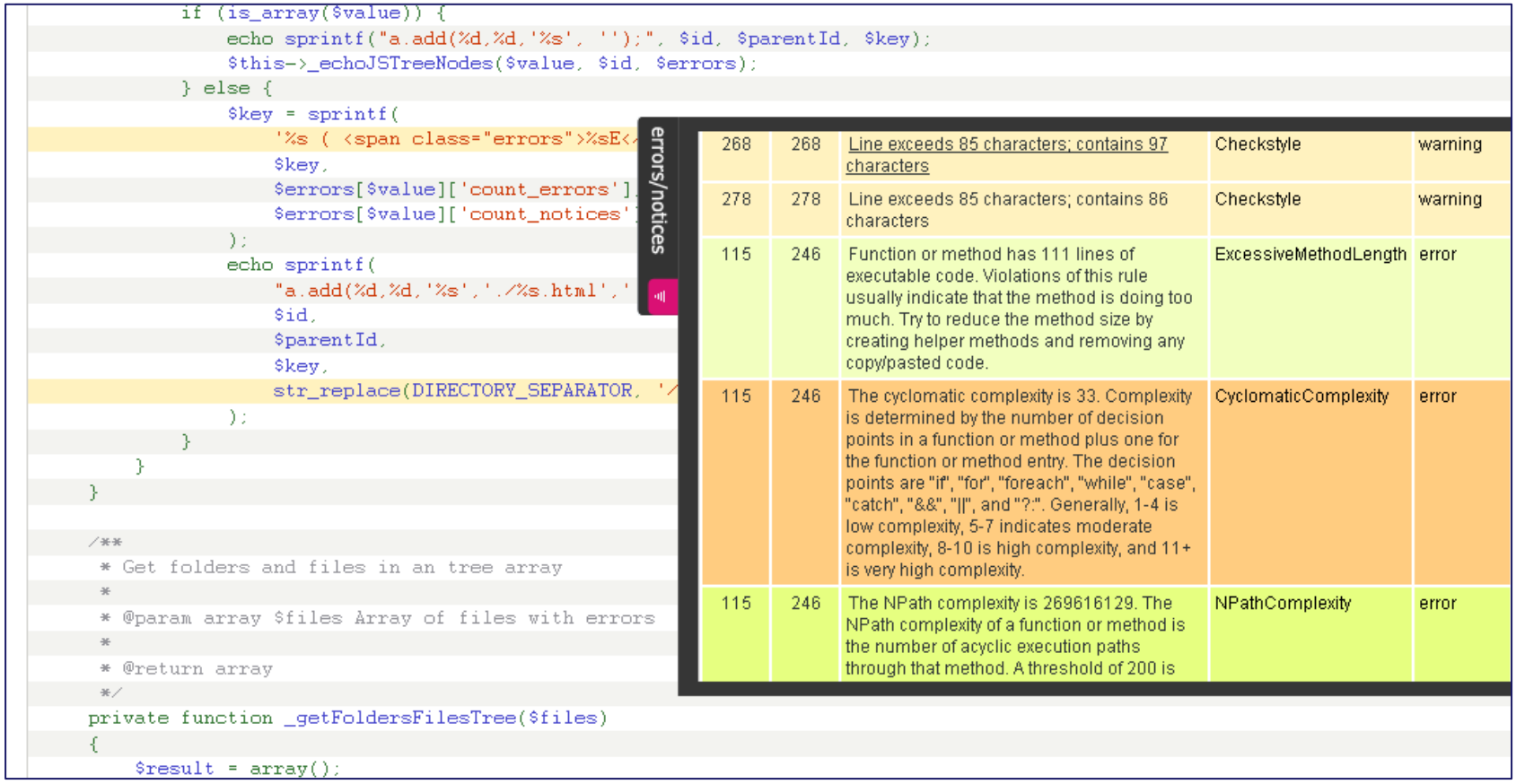
Fonctionnalité temporairement dans PHP_Unit

- Déportée par la suite dans d'autres outils

CRAP	PHPMD
Complexité Cyclomatique	PHPDepend, PHPMD
Complexité Npath	PHPDepend, PHPMD
ELOC	PHPDepend, phploc
Détection des Copier/Coller	phpcpd

PHP_CodeBrowser

Montre les violations dans les sources



```
            if (is_array($value)) {
                echo sprintf("a.add(%d,%d,'%s', '');", $id, $parentId, $key);
                $this->_echoJSTreeNodes($value, $id, $errors);
            } else {
                $key = sprintf(
                    '%s ( <span class="errors">%sE</
                    $key,
                    $errors[$value]['count_errors'],
                    $errors[$value]['count_notices']
                );
                echo sprintf(
                    "a.add(%d,%d,'%s','./%s.html','
                    $id,
                    $parentId,
                    $key,
                    str_replace(DIRECTORY_SEPARATOR, '/
                );
            }
        }
    }

    /**
     * Get folders and files in an tree array
     *
     * @param array $files Array of files with errors
     *
     * @return array
     */
    private function _getFoldersFilesTree($files)
    {
        $result = array();
```

errors/notices

268	268	Line exceeds 85 characters; contains 97 characters	Checkstyle	warning
278	278	Line exceeds 85 characters; contains 86 characters	Checkstyle	warning
115	246	Function or method has 111 lines of executable code. Violations of this rule usually indicate that the method is doing too much. Try to reduce the method size by creating helper methods and removing any copy/pasted code.	ExcessiveMethodLength	error
115	246	The cyclomatic complexity is 33. Complexity is determined by the number of decision points in a function or method plus one for the function or method entry. The decision points are "if", "for", "foreach", "while", "case", "catch", "&&", "||", and "?:". Generally, 1-4 is low complexity, 5-7 indicates moderate complexity, 8-10 is high complexity, and 11+ is very high complexity.	CyclomaticComplexity	error
115	246	The NPath complexity is 269616129. The NPath complexity of a function or method is the number of acyclic execution paths through that method. A threshold of 200 is	NPathComplexity	error

afup

FORUM PHP 2010

TOP mission Puls@r

Les défis :

Equipe jeune

Conception externe

Suivi d’externes !

Adaptation PHPCS

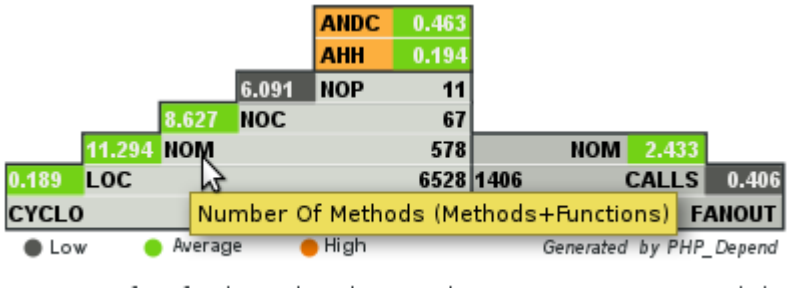
- RefQual
  - Guide de Programmation PHP
    - Règles, Recommandations, Bonnes Pratiques
    - Identifiés par des codes
  - Souvent standard, parfois spécifiques

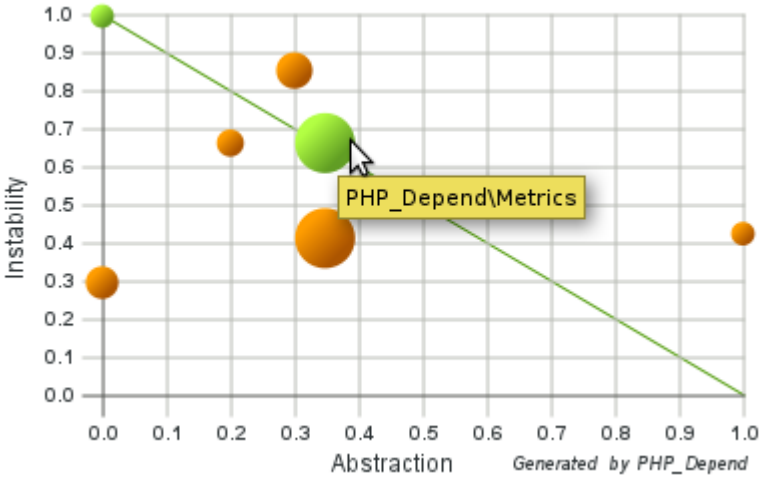
Création d'un standard PHPCS GN

Intégration PHP_Depend

Métriques++

Analyse Orienté Objet





afup

FORUM PHP 2010

Documentation & Formation

Outil riche mais composite

- Prise en main complexe
- Doc : installation, paramétrage, utilisation

Formation aux tests automatisés

- PHPUnit, TDD, intégration Zend Studio, Zend_Test

Premiers retours « terrain »

Installation plutôt laborieuse

TU difficiles à mettre en place

PHPCS difficilement exploitable

Aucune visibilité sur les performances

Suivi périodique

Hebdomadaire, puis bimensuelle

Mise en place des environnements

Assistance technique

Suivi PIC

Audits de code

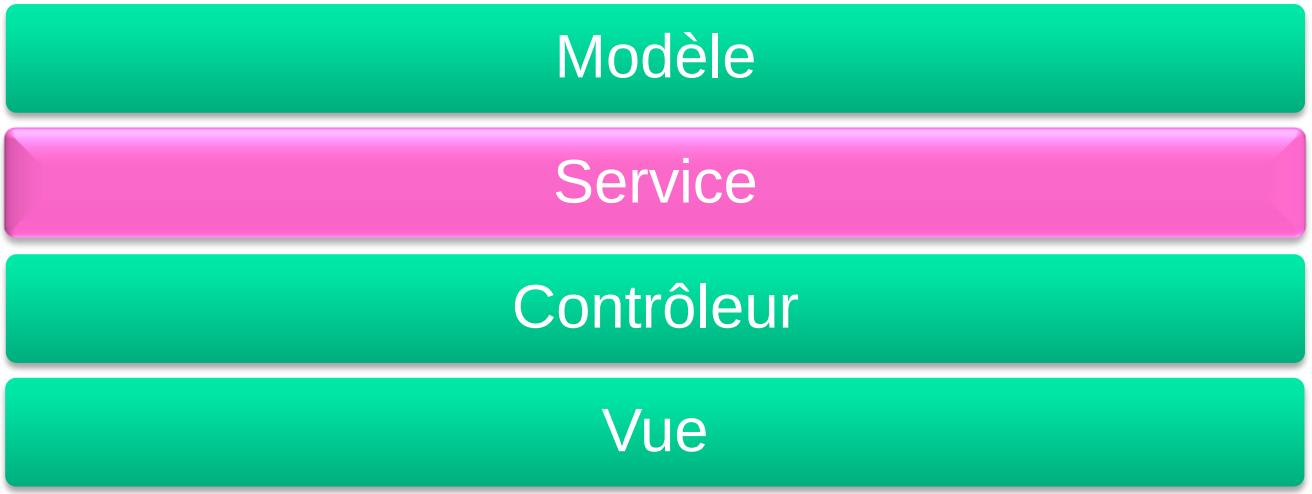
…

SQLI_CodeSniffer

- Réponse aux problèmes d'usage GN
- Identifier chaque violation
- Attribuer sévérités et messages par configuration
  - Personnaliser les messages (RefQual)
  - Modifier dynamiquement les sévérités
  - Collecter des statistiques par type de violation

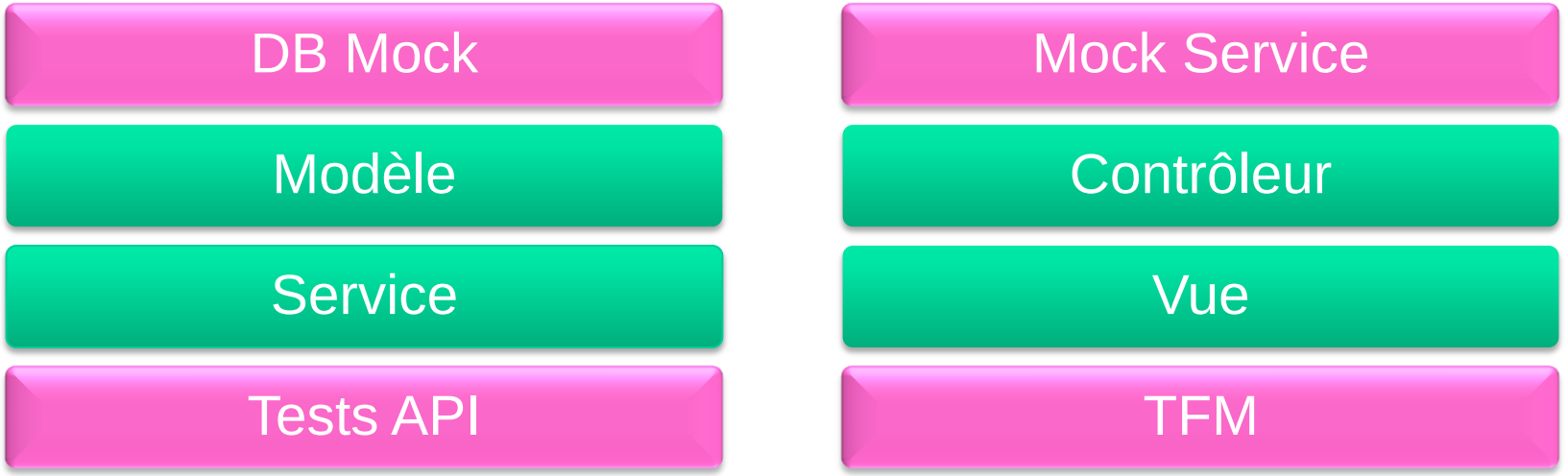
Repris dans la dernière version 1.3 de PHPCS

Couche Services



API Métier

Couche Services



parallelisation des devs

tests et mocks obligatoires

conception d'un système d'instanciation des objets doubles

Sélénium / Zend Platform

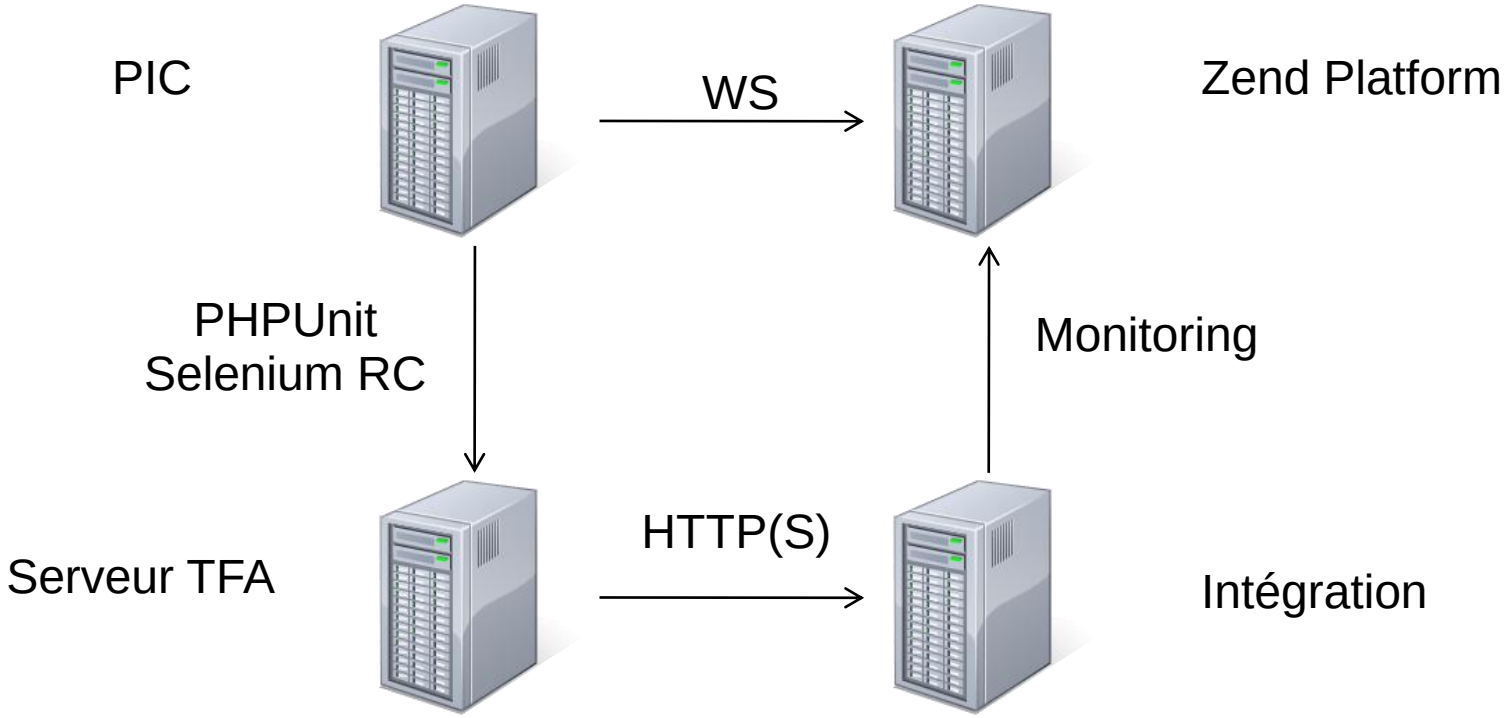
Analyse des performances dès les premières développements

Aide à détecter rapidement des erreurs de conception/implémentation

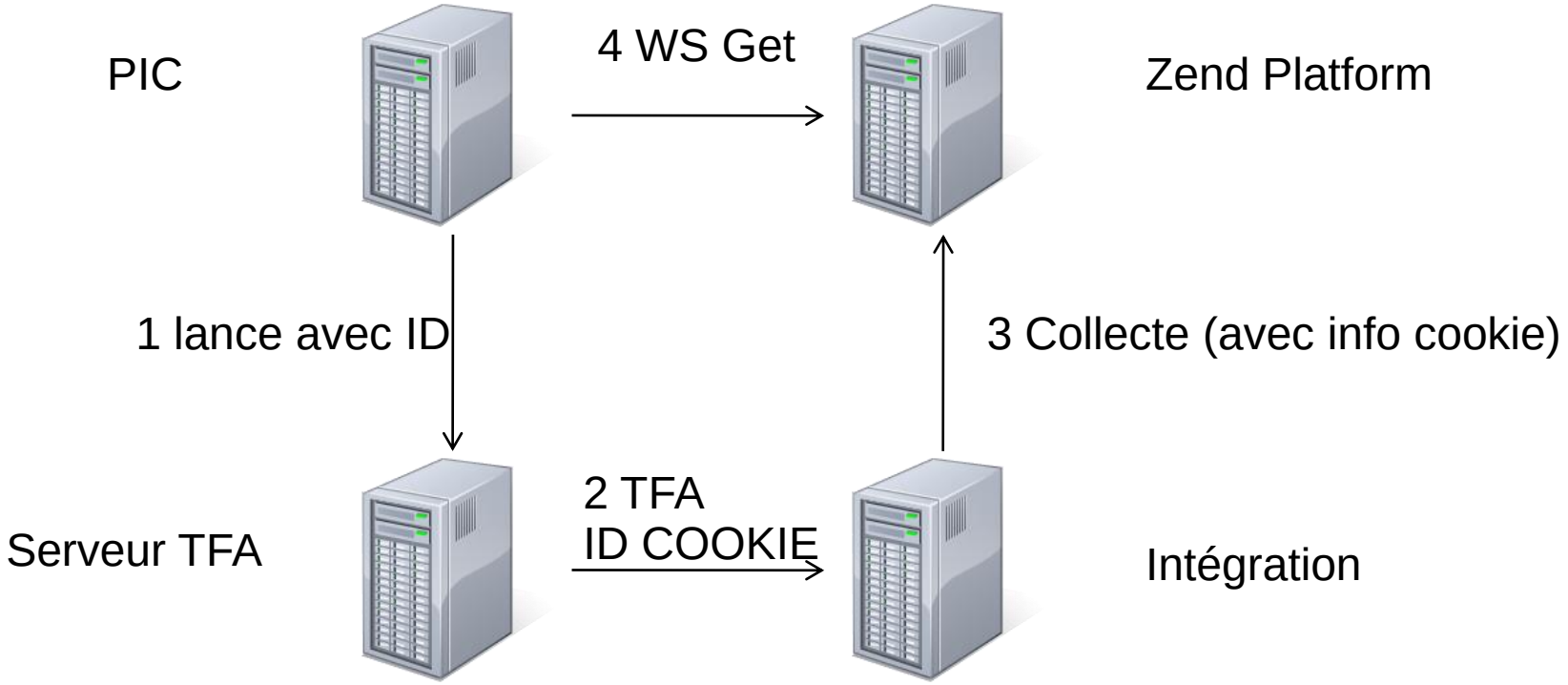
Utilise le pilotage de Sélénium par PHPUnit

Utilise la fonctionnalité de monitoring de la Zend Platform

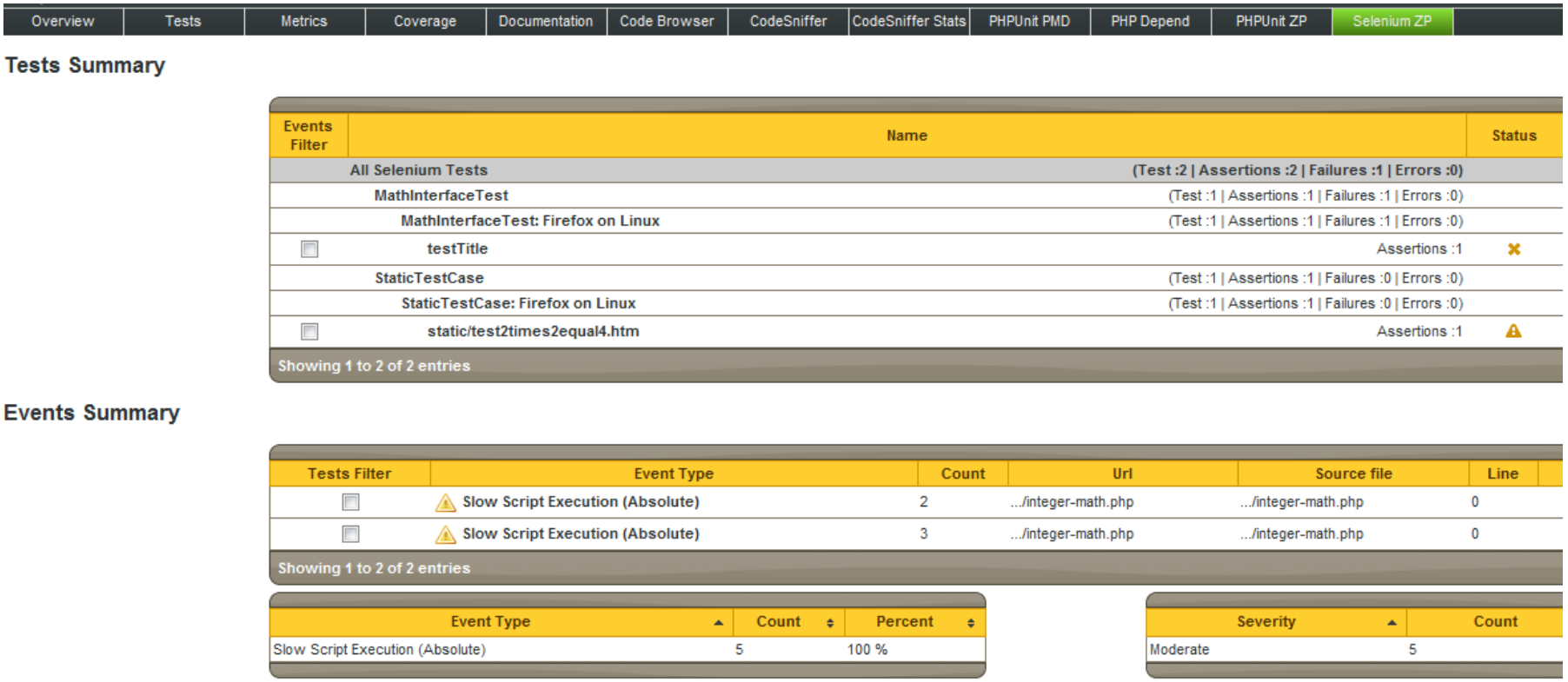
Sélénium / Zend Platform



Sélénium / Zend Platform



Sélénium / Zend Platform



Overview | Tests | Metrics | Coverage | Documentation | Code Browser | CodeSniffer | CodeSniffer Stats | PHPUnit PMD | PHP Depend | PHPUnit ZP | Selenium ZP

Tests Summary

Events Filter	Name		Status
	All Selenium Tests	(Test :2 | Assertions :2 | Failures :1 | Errors :0)	
	MathInterfaceTest	(Test :1 | Assertions :1 | Failures :1 | Errors :0)	
	MathInterfaceTest: Firefox on Linux	(Test :1 | Assertions :1 | Failures :1 | Errors :0)	
☐	testTitle	Assertions :1	✖
	StaticTestCase	(Test :1 | Assertions :1 | Failures :0 | Errors :0)	
	StaticTestCase: Firefox on Linux	(Test :1 | Assertions :1 | Failures :0 | Errors :0)	
☐	static/test2times2equal4.htm	Assertions :1	⚠

Showing 1 to 2 of 2 entries

Events Summary

Tests Filter	Event Type	Count	Url	Source file	Line
☐	⚠ Slow Script Execution (Absolute)	2	.../integer-math.php	.../integer-math.php	0
☐	⚠ Slow Script Execution (Absolute)	3	.../integer-math.php	.../integer-math.php	0

Showing 1 to 2 of 2 entries

Event Type	Count	Percent
Slow Script Execution (Absolute)	5	100 %

Severity	Count
Moderate	5

afup

FORUM PHP 2010

Problèmes d’installation…

Plus de deux jours!

Erreur humaine inévitable

Documentation d’install de plus de 100 pages

Débianisation !

PHP-CI-PKG

Chaque outil est packagé

phpUnderControl est un paquet

CruiseControl aussi (merci Alexis Gruet)!

utilise dh-make-php

www.assembla.com/wiki/show/php-ci-pkg

Débianisation : gore details

- Problèmes de packaging PEAR
- Localisation différente de CruiseControl
- Utilisation de dotdeb
- Respect partiel des règles de débianisation
  - à finaliser et nettoyer par un pro Debian
  - Quelques volontaires dans la salle ? ;-)

The fastest PHPUC install !

La GN a un repository locale (avec dotdeb)

à partir d'une lenny « nue »

apt-get install php-phpuc

apt-get install pic-php-sqli

PHPUC opérationnel en quelques minutes

Puls@r now

Risques maitrisés

Montée en compétence des équipes

Des milliers d’assertions PHPUnit

afup

FORUM PHP 2010

PIC-PHP-SQLI now

Sonar

- Plugin Sonar pour PHP (voir après)
- Etat de l’art Java
- Mieux outillé pour le suivi qualité

Évolution de la plateforme vers Hudson ?

- Les développements/interfaces sont en PHP

Moralité n° 3

L'installation de l'outil ne suffit pas

L'IC est un ensemble de pratiques :

il faut un accompagnement au changement

Moralité n° 2

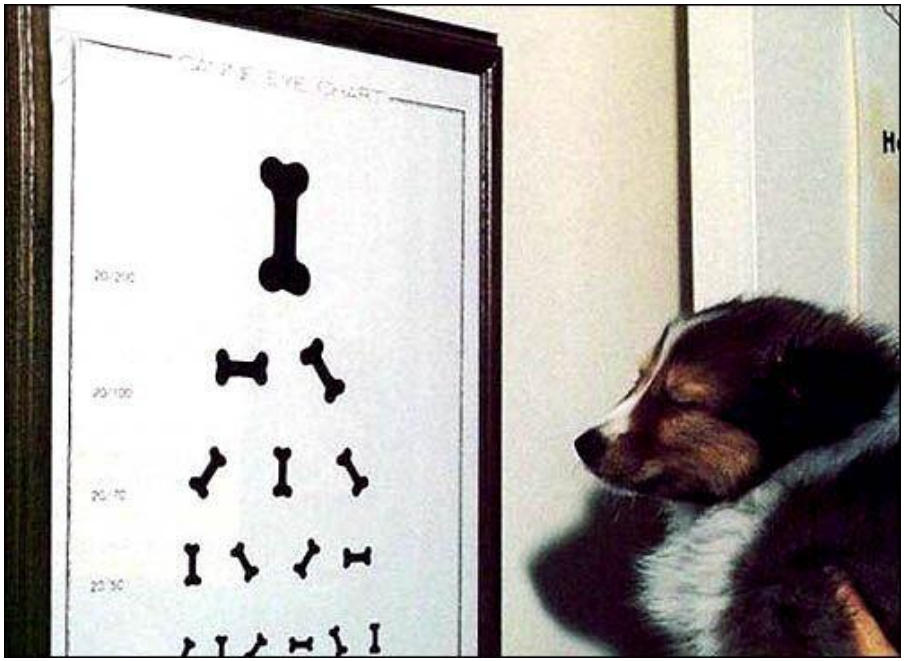
Moins de discrétion !

- Plus de push
- Lier la PIC aux déploiements

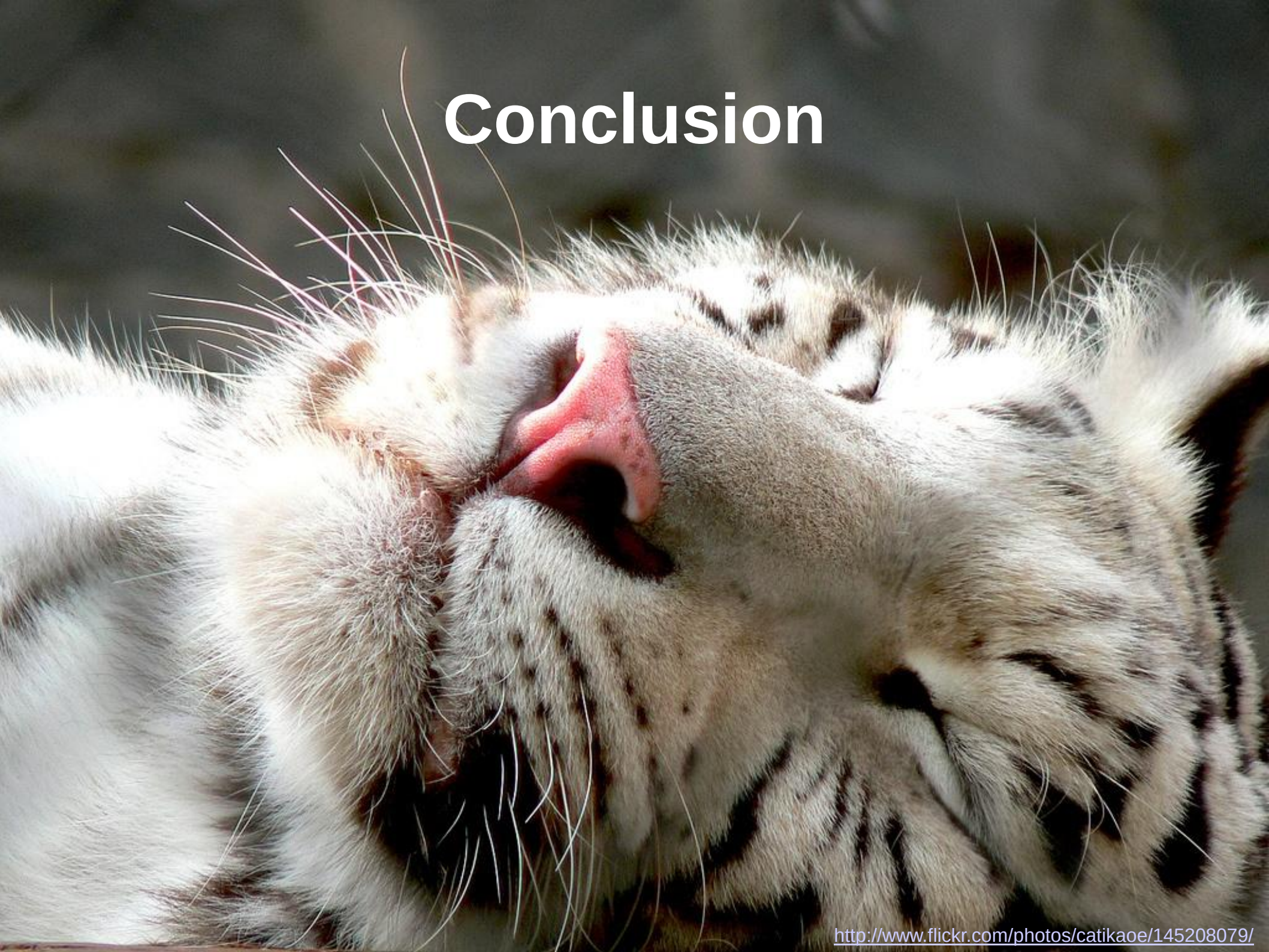
Moralité n° 1

Tests, tests, tests !

Le nerf de la guerre : IC ⇔ tests automatisés



Conclusion



http://www.flickr.com/photos/catikaoe/145208079/

Merci!

Questions/réponses