

# **Bonnes Pratiques**

En un clin d'oeil

Gérald Croës - gerald@copix.org

http://gcroes.com/conf/php/quebec/2009/bonnes\_pratiques.pdf

# Objectifs de la présentation



💮 ... des outils

🖐 ... des méthodes

💮 ... des pièges

🥮 ... d'idées

.... pour en tirer un ensemble de bonnes pratiques

## **Avants propos**

Cette présentation et les conseils qu'elle contient...

- Ne sont pas des vérités absolues,
- Sont à appliquer avec modération
- Sont a sélectionner avec précaution
- Ne sont pas toujours applicables / opportuns

## **Etude / Conception**

- Démarrage du projet
- Réunions de suivi
- Normes de développement
- **%** KISS
- YAGNI
- Patterns
- Templates ou pas ?

# [Conception] Démarrage projet

- Présenter le CDC
- Discuter des fonctionnalités
- Proposer des solutions / alternatives
- Marifier les risques / problèmes potentiels
- Définir les API
- Développer l'architecture
- Etablir un plan de test
- Etablir un plan de livraison
- Etablir des réunions de suivi à date fixe
- Toute l'équipe doit être impliquée dans la démarche

# [Conception] Réunions de suivi

- Avec le client
- Avec l'équipe projet

- Eviter l'effet tunel
- Identifier les problèmes au plus tôt

## [Conception] Normes de développement

- Des normes stables et classiques (PEAR, Zend)
- Indentation
  Et acolades, retraits, opérateurs ternaires, ...
- Règles de nommage Fonctions, classes, variables, tables, champs, ...
- Documentation (phpDoc)
- Une norme n'a de chance d'être appliquée que Si elle est simple, claire et concise
  - Si elle est « automatisable » / Assistée par des outils (phpCodeSniffer, Hook SVN, IDE, ...)

## [Conception] Normes de développement 2

```
* Fonction avec deux paramètres
* @param $p1 le paramètre 1
* @param $p2 le paramètre 2
function do ($p1, $p2) { for ($i=0; $i<count ($p2); $i+
+) //parcours de $p2
        //on définit a $x la valeur trouvée dans p2 a la
position $i qu'on additionne avec $p1
     x = p2[i] + p1;
   global $1; isset ($1) ? $1('fin do'): ";
```

## [Conception] Normes de développement 3

```
* Calcul un tableau dont les valeurs sont celles de $arSource auxquelles on
ajoute $toAdd
* @param $toAdd la valeur fixe a ajouter a chaque élément de tableau
* @param $arSource le tableau de valeurs
* @TODO: ne fonctionne qu'avec des tableaux séquentiels
function addToArray ($toAdd, $arSource) {
 $return = array ();
 for ($i=0; $i<count ($arSource); $i++) {
   //création du tableau résultat
   $return[$i] = $toAdd + $arSource[$i];
 If ($log = Log::getInstance ()){
   $log->add ('ajout terminé');
 return $return;
```

# [Conception] KISS

- Keep It Smart & Simple (oui, il existe d'autres acronymes ;-))
- Simple = lisible = compréhensible
- Compliqué & Sophistiqué ne sont pas synonymes de qualité
- « La simplicité est la sophistication suprême »
- « On devrait tout rendre aussi simple que possible, mais pas plus »
- Ne « voir » que ce qui est nécessaire

## [Conception] YAGNI



Le code superflus Complexifie l'implémentation

Est source de bug

Prend du temps qui aurait pu être utilisé sur des besoins immédiats (documentation, tests, ...)

Ne servira peut être jamais

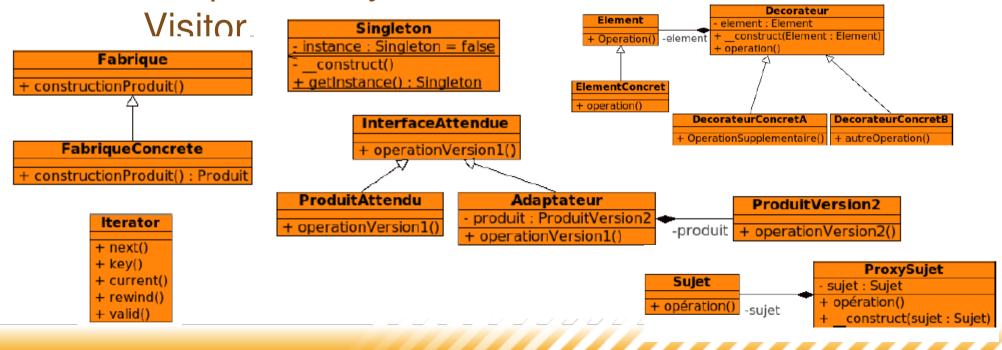
Ne sera peut être pas ce dont on aura très exactement besoin dans le futur



## [Conception] Patterns



Les incontournables Factory, Singleton, Adapter, Decorator, Proxy, Façade, Chain of Responsability, Iterator, Observer. Strategy.



## [Conception] Templates ou pas?

```
Moteur de template != Template
{$value}
<?php echo isset ($value) ? html_entities ($value) :
"; ?>
```

Un template
 Encapsule la logique d'affichage
 N'encapsule pas les règles de gestion

## Développement

- Configuration des postes
- Gestionnaire de sources
- Gestionnaire de tâches
- Frameworks
- **IDE**
- SPL

- Extensions PHP
- Javascript non intrusif
- 🥮 Ajax
- Procrastination
- Transactions
- Exceptions
- Opérateurs

# [Développement] Configuration des postes

- « Eco Attitude » recyclez les vieux PC!
- Short open tags
- Magic quotes
- @ error\_reporting E\_ALL | E\_STRICT
- Aucun cache d'opCode
- Memory limit 8
- output\_buffering désactivé

Astuce : ne pas mettre les ?> en fin de fichier PHP

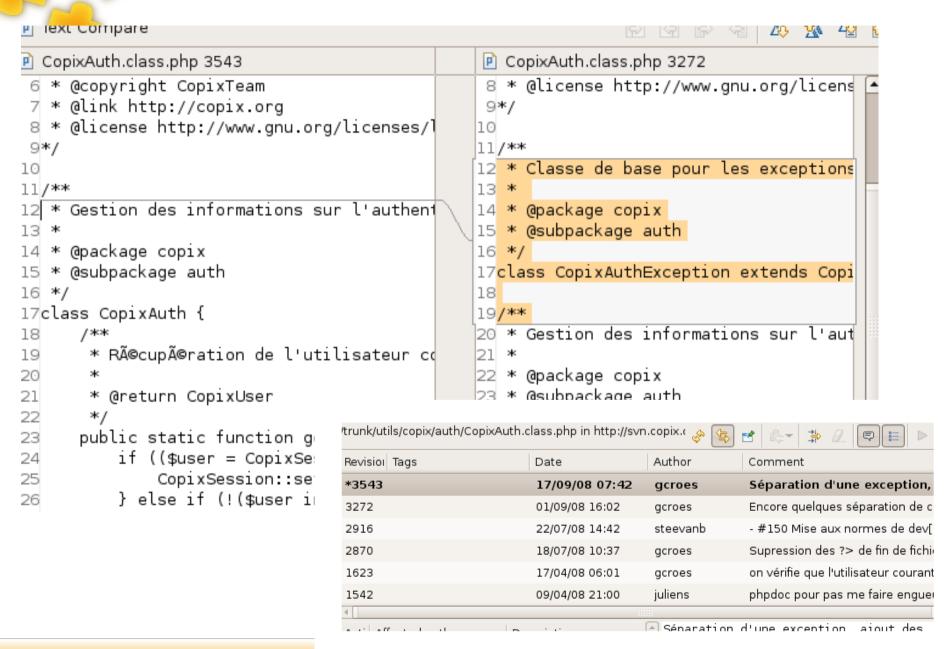
## [Développement] Gestionnaire de sources

- SVN, CVS, Mercurial ou autres...
- Une fonctionnalité = Un commit
- Des commentaires explicites
- Règles d'utilisation des branches / troncs / tags Branche = Projet

Tronc = Production

Tag = Livraison

# [Développement] Gestionnaire de sources



## [Développement] Gestionnaire de tâches



Timeline = Daily News du développeur

Utilisez les tickets à outrance (résultat mesurable)

Faites référence aux tickets dans les commentaires de commit

Ticket	Summary	Component	Version	Milestone	Туре
#366	MySQL #1064 - "`" manquants	DB	3.0.0		Defect
#342	Surcharge DAO avec appel _dao('module table')	DAO	3.0.3	3.0.5	Defect
#375	BUG à l'installation sur Free	Auth	3.0.4	3.0.5	Defect
#384	arModulePath et chemin sans / de fin	Core	3.0.0	3.0.5	Defect
#288	Arborescence modules Copix 3.1	Auth	3.0.0	3.1.0	Defect
#296	Documenter les options de CopixConfig	Documentation	3.0.3	3.1.0	Defect
#310	Bug avec la méthode initialize de CopixConfig	Core	trunk	3.1.0	Defect
#311	Erreurs avec CopixFile dans class CopixModuleInstaller	Module	3.0.0	3.1.0	Defect
#314	Cacher le bloc "Plus d'informations" dans les exceptions par javascript	Core	3.0.0	3.1.0	Defect
#332	Vérifier les requêtes pour le charset du driver mysql	DB	trunk	3.1.0	Defect
#341	Bug sur CopixPPO avec un tableau	Core	trunk	3.1.0	Defect



# [Développement] Gestionnaire de tâches 2

#### 10/20/08:

18:28 Changeset [3697] by juliens

Gestion des caches files avec des CopixSerializableObject

3 17:21 Changeset [3696] by gcroes

Ajout du test de la version de PHP pour éviter que ceux qui sont en PHP4 ...

17:20 Changeset [3695] by juliens

Gestion des caches files avec des CopixSerializableObject

3 16:59 Changeset [3694] by gcroes

Start appèle maintenant stop pour être sûr que deux processus ne tournent ...

16:57 Changeset [3693] by gcroes

Ajout du test sur 404

16:57 Changeset [3692] by gcroes

Bricoles dans les noms de variable

@ 16:56 Changeset [3691] by gcroes

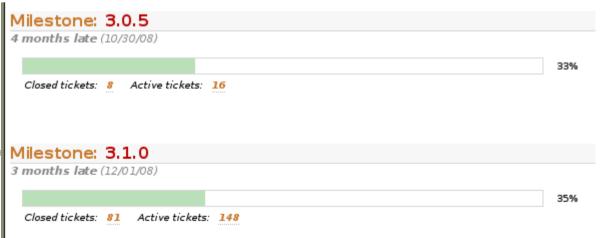
Inutile de tester longDescription qui est toujours renseigné

(diff) 12:53 CopixFormObserver edited by gcroes

12:15 Changeset [3690] by gcroes

Utilisation de la méthode start plutot que de dupliquer le code de

12:12 CopixFormObserver edited by gcroes (diff)



## [Développement] Framework

- Ne réinventez pas la roue
- Choisissez votre Framework en fonction De sa communauté
  - De sa documentation
  - De ses fonctionnalités
  - .... et surtout en fonction de vos affinités
- … ou mieux, n'en utilisez pas :-)

## [Développement] IDE

- PHPEclipse, PDT, Netbeens, Visual Studio For PHP, PHPEdit, Delphi for PHP, vi, ...
- Les fonctionnalités dont vous avez besoin Intégration SVN / CVS

Complétion

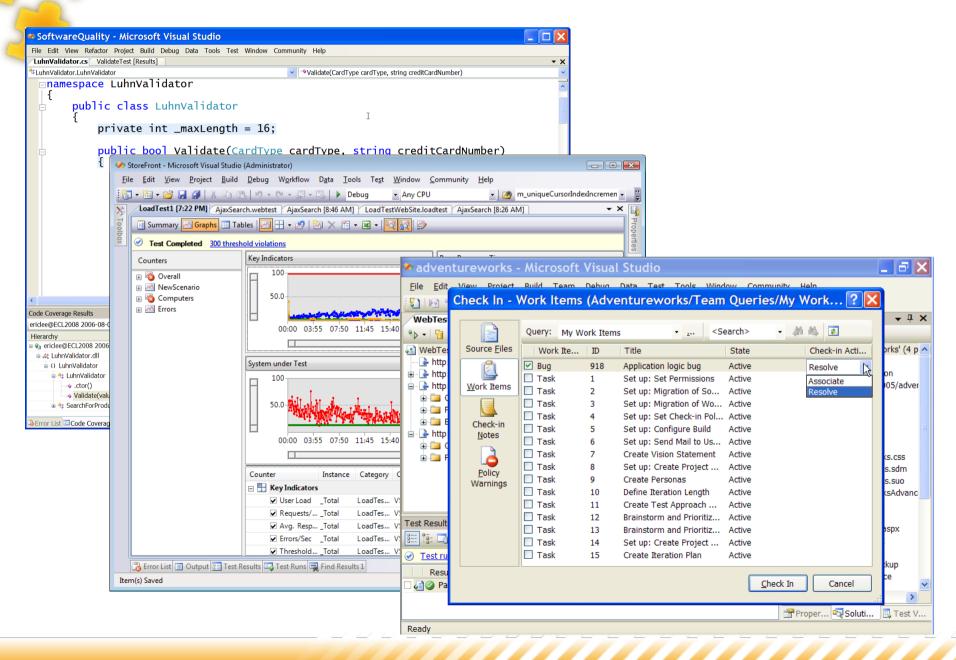
Recherche dans les sources

**Documentation** 

Todo / FIXME

Explorateurs de classes

# [Développement] IDE 2



## [Développement] SPL (Interfaces)

- Iterator
  [current, key, next, valid, rewind]
- SeekableIterator [(iterator), seek]
- RecursiveIterator [(iterator), getChildren, hasChildren]
- ArrayAccess
   [offsetGet, offsetSet, offsetExists, offsetUnset]
- Countable [count]

## [Développement] SPL (Classes)

DirectoryIterator, RecursiveDirectoryIterator
foreach (\$directories = new DirectoryIterator ('./') as
\$file){ //...

SplObjectStorage
Attach / Dettach

Gère l'unicité des instances dans la collection

## [Développement] Autoload

- Evite l'inclusion de classes non nécessaires
- Une problématique de moins à gérer
- \_\_autoload (\$className)
- class\_exists (\$nom, [boolean])
  Le deuxième paramètre indique s'il faut utiliser l'autoloader ou non
- spl\_autoload\_register (\$callback);
  La solution la plus souple

! Doit être un processus « rapide »

# [Développement] Fonctions magiques

- get
- set
- \_\_isset
- unset
- 🥮 call
- —callStatic (5.3+)
- sleep
- 🥮 \_\_\_wakeup
- 🥟 clone
- \_\_toString

## [Développement] Fonctions magiques 2

#### **\$ppo['test']->foo->test['somekey'][] = 1;**

```
class PPO implements ArrayAccess {
  public function & get ($pName) {
      $this->$pName = new PPO ();
     return $this->$pName;
  public function offsetGet ($pOffset) {
      return $this->$p0ffset;
  public function offsetSet ($pOffset, $pValue) {
      if ($p0ffset === null) {
         $vars = get object vars ($this);
         if (count (\$vars) === 0) {
            poffset = 0;
         } else {
            $pOffset = max (array keys (get object vars ($this))) + 1;
      $this->$p0ffset = $pValue;
  public function offsetExists ($p0ffset) {
      return isset ($this->$p0ffset);
  public function offsetUnset ($p0ffset) {
      $this->$p0ffset = null;
}
```

## [Développement] Extensions PHP









memcache



tidy

🥮 mbstring

Le code C sera toujours plus rapide

# [Développement] Javascript



« non intrusif »



Encore des frameworks

Mootools

Jquery

Dojo

Yahoo UI

. . .

# [Développement] Ajax

- A utiliser a bon escient
- Limiter le trafic au strict nécessaire (complétion)
- Aider à la saisie
- Sauvegarder en cours de frappe
- Notifier / Effectuer des tâches transparentes

```
var req = new Request.HTML({url:'ajax.php',
    onSuccess: function(html) {
        $('result').set('text', ");
        $('result').adopt(html);
    },
    onFailure: function() {
        $('result').set('text', 'La requête a échouée.');
    }
});
```

## [Développement] Procrastination

- Synonyme d'abandon dans 80% des cas
- Synonyme de perte de temps future

## [Développement] Transactions

Pour les opérations multi-table qui impliquent du code PHP

```
$pdo->beginTransaction ();
try {
    $cuisseDePoulet = Frigo::get ()->ingredient ('cuisse de
poulet');
    Cuistot::get ()->recette ('cuisse de poulet grillé')->cuisiner
($cuisseDePoulet);
    $pdo->commit ();
}catch (StockInsuffisant $e){
    $pdo->rollback ();
}
```

## [Développement] Exceptions

- A réserver pour les cas exceptionnels
- Préférable dans tous les cas au test de valeur de retour
- Hierarchiser

## [Développement] Opérateurs

```
" === et !== plutôt que == et !=
some = '1';
$one_string = ' 1 gaffe qui pourrait poser problèmes';
$one_string_2 = 'la 1 gaffe pose moins de problèmes';
$bool = '0 ';
if (1 == \$one){
 echo "1 est bien '1' <br />";
if (1 == \text{sone\_string})
 echo "1 est bien $one string <br/> />";
if (1 == $one_string_2){
  echo "1 est bien $one_string_2 <br />";
if (true == $bool){
  echo "$bool est vrai";
```

## **Tests / Debogages**

- Tests unitaires
- Selenium
- 🥮 Xdebug
- Firebug
- Firephp
- Dbug

## [Tests] Tests unitaires

- PHPUnit, SimpleTest
- Test Driven Development
- Refactoriser / Améliorer sans détériorer

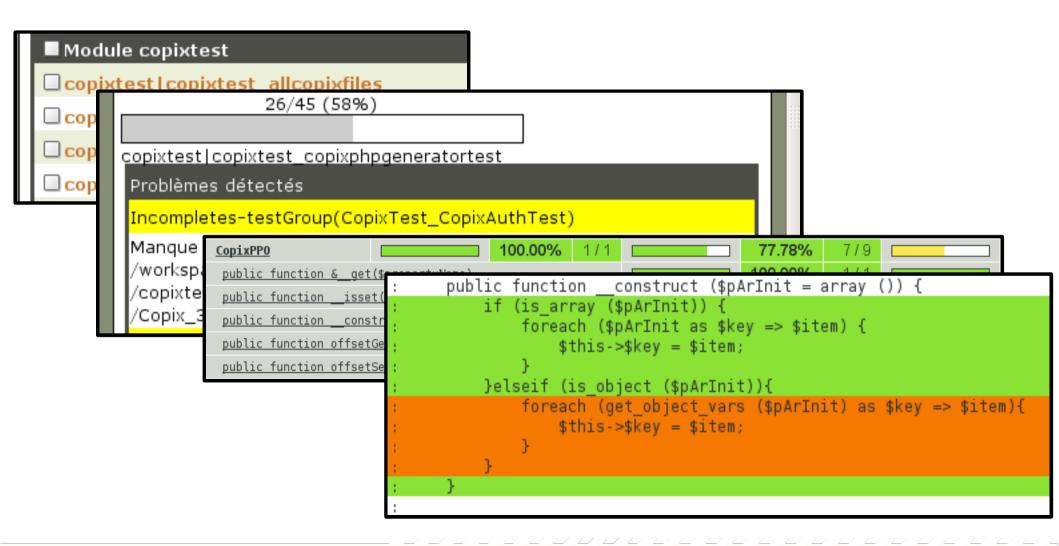


## [Tests] Tests unitaires 2

```
@subpackage copixtest
🗸 🏣 tests
     n copixtest allcopixfiles.class.ph
                                        15class CopixTest CopixDateTimeTest extends CopixTest {
                                               function testToXXXX (){
     n copixtest alldaotest.class.php
                                                   CopixIl8N::setLang ('fr');
     copixtest copixauthtest.class.
                                                   $this->assertEquals ('19761225', CopixDateTime::dateToYYYYMMDD (
                                        19
                                                   $this->assertFalse (CopixDateTime::dateToYYYYMMDD ('215
     n copixtest copixcachetest.class
                                        20
21
22
23
24
25
26
27
28
30
                                                   $this->assertFalse (CopixDateTime::dateToYYYYMMDD ('215/a276'));
     R copixtest copixclassesfactory
                                                   $this->assertEquals ('19761225150000'
                                                                                            ,CopixDateTime::DateTimeToyyyyr
     n copixtest copixconfigtest.class
                                                   $this->assertEquals ('19761225100106', CopixDateTime::timestampToyyyy
CopixI18N::setLang ('en');
     copixtest_copixcontexttest.cla
                                                   $this->assertEquals ('19761225', CopixDateTime::dateToYYYYMMDD ('12
     R copixtest copixcookieobject.cl
                                                   $this->assertFalse (CopixDateTime::dateToYYYYMMDD (':
                                                   $this->assertNull (CopixDateTime::dateToYYYYMMDD (null));
     n copixtest copixcsv.class.php 2
                                                   $this->assertEquals ('19761225150000',CopixDateTime::DateTimeToyyyy
     n copixtest copixdaofactorytest
                                                   $this->assertEquals ('19761225030000'
                                                                                            ,CopixDateTime::DateTimeToyyyy
                                                   $this->assertEquals ('19761225', CopixDateTime::timestampToyyyymmdd(?
     n copixtest copixdataobjecttest
     n copixtest copixdatetimetest.c
```



## [Tests] Tests unitaires 3







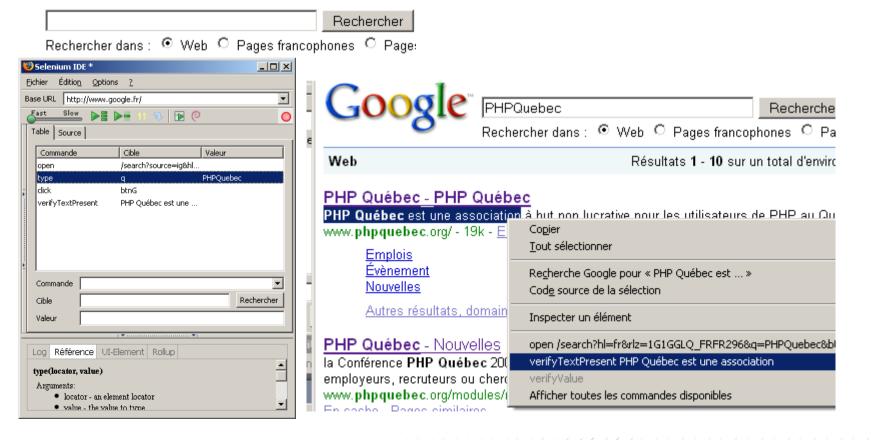
## [Tests] Selenium



#### Tests fonctionnels



... sur plusieurs navigateurs



## [Tests] Selenium 2

```
require once 'PHPUnit/Extensions/SeleniumTestCase.php';
class Example extends PHPUnit Extensions SeleniumTestCase
 function setUp()
   $this->setBrowser("*chrome");
   $this->setBrowserUrl("http://www.google.fr/");
 function testMyTestCase()
   $this->open("/search");
   $this->type("q", "PHPQuebec");
   $this->click("btnG");
   try {
        $this->assertTrue($this->isTextPresent("PHP Québec est une
association")):
    } catch (PHPUnit Framework AssertionFailedError $e) {
        array push($this->verificationErrors, $e->toString());
```



## [Tests] XDebug



Debogueur



Erreurs explicites et traces d'appel



Codecoverage

Notice: Undefined variable: oops in /home/gcroes/workspace/Copix\_3/project/modules/stable/standard/de/default.actiongroup.php on line 25

(!) Notice: Undefined variable: oops in /home/gcroes/workspace/Copix\_3/project/modules/stable/ /actiongroups/default.actiongroup.php on line 25

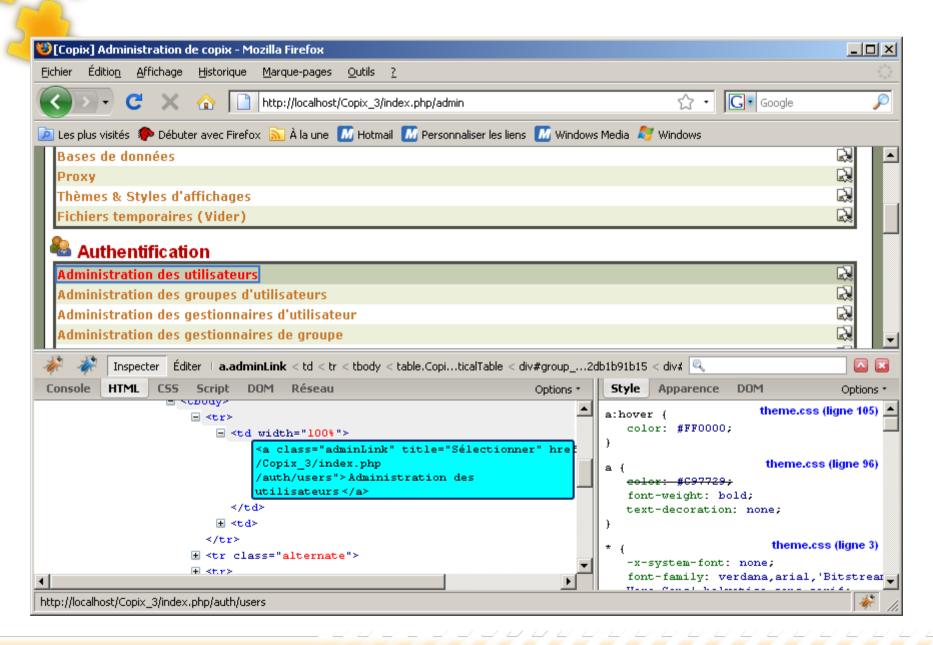
#### Call Stack

#	Time	Memory	Function	Location
1	0.0006	38888	{main}( )	/index.php:0
2	0.0285	448060	CopixController->process()	/index.php:24
3	0.0310	483852	CopixActionGroup::process()	/CopixController.clas
4	0.0325	495980	ActionGroupDefault->processDefault( )	/CopixActionGroup.cl

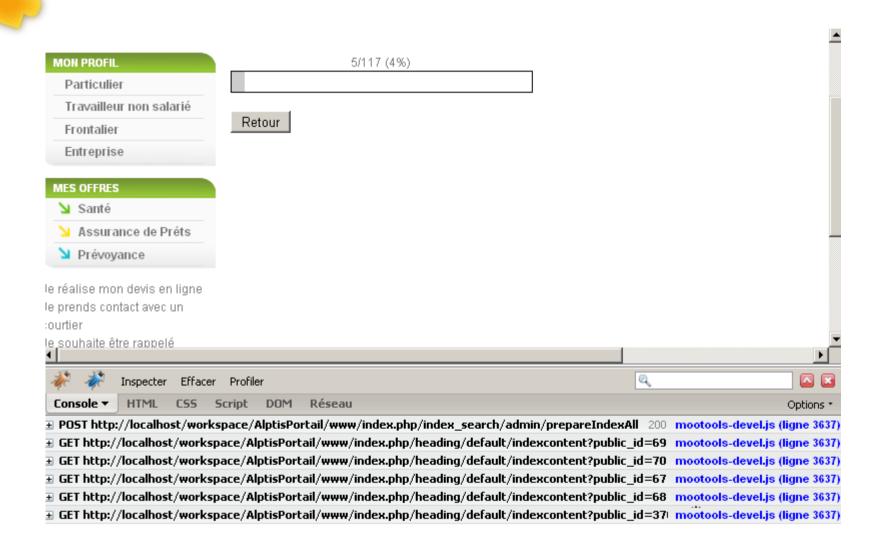
## [Tests] Firebug

- Inspection du HTML / CSS / Javascript
- Possibilité de modifier le HTML / CSS
- Monitoring des appels (images, requêtes ajax, css, ...)
- console.log (), l'autre « echo »

## [Tests] Firebug 2 (HTML / CSS)



## [Tests] Firebug 3 (Ajax, Requêtes)



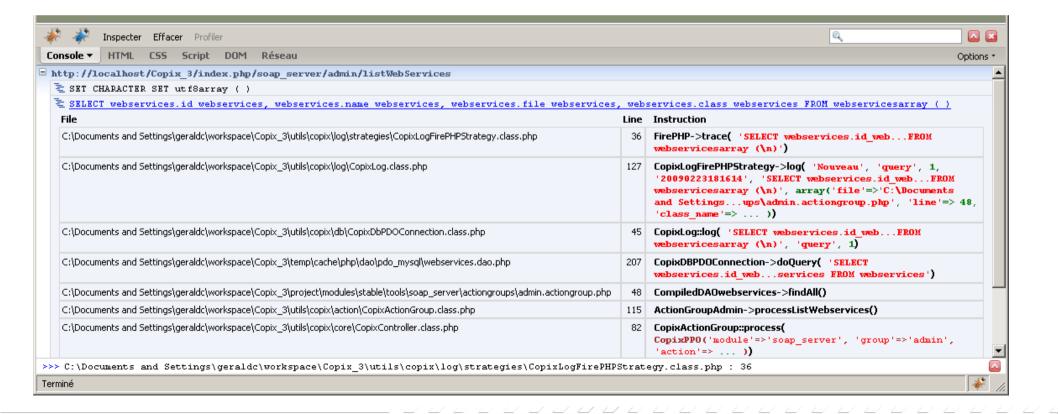




D Compagnon de jeu de FireBug pour PHP



\$fb->trace ('message');



## [Tests] dBug

- Affichez rapidement & clairement le contenu d'une variable
- Limitation avec PHP5 (variables privées masquées)

```
Alternative: xdebug_var_dump
ini_set ('xdebug.var_display_max_depth', 20);
ini_set ('xdebug.var_display_max_children', 128);
ini_set ('xdebug.var_display_max_data', 40960);
```

## [Tests] dBug 2

\$ar (array)							
conference	array						
	titre	Bonnes pratiques					
	Durée	75					
	Date	04/03/2009					
	Heure	12:15					
	DateTime	object					
		construct	[function]				
		format	[function]				
		modify	[function]				
		getTimezone	[function]				
		setTimezone	[function]				
		getOffset	[function]				
		setTime	[function]				
		setDate	[function]				
		setISODate	[function]				

```
array
  'conference' =>
    array
      'titre' => string 'Bonnes pratiques' (length=16)
      'Durée' => int 75
      'Date' => string '04/03/2009' (length=10)
      'Heure' => string '12:15' (length=5)
      'DateTime' =>
        object (DateTime) [1]
                   array
                      'conference' =>
                        array
                          'titre' => string 'Bonnes pratiques' (length=16)
                          'Durée' => int 75
                          'Date' => string '04/03/2009' (length=10)
                          'Heure' => string '12:15' (length=5)
                          'DateTime' =>
                            object(ConfDateTime)[1]
                              private ' date' => string '04/03/2009' (length=10)
                              private ' hour' => string '12:15' (length=5)
```

\$ar (array)							
conference	array						
	titre	Bonnes pratiques					
	Durée	75					
	Date	04/03/2009					
	Heure	12:15					
	DateTime	object					

## **Optimisation**

- En temps et en heure
- OpCode
- Caches
- Memcache
- Querycache
- Compression de la sortie
- Xdebug..... encore
- Compression de la sortie

- Complexifie le code inutilement
- 🥮 Souvent mal ciblée
- Risque d'être revu lors de l'affinement des spécifications / règles de gestion

```
public function getVisibility ($pPublicId, & $pInherited){
   $this-> calcInheritedStuff ($pPublicId);
   $pInherited = self::$ visibility[$pPublicId]['inherited'] === $pPublicId ? false : self::
$ visibility[$pPublicId]['inherited'];
   return self::$ visibility[$pPublicId]['visibility'];
private function calcInheritedStuff ($pPublicId){
  if (array key exists ($pPublicId, self::$ visibility)){
      return:
   $path = array reverse ($this->getHeadingPath ($pPublicId));
  foreach ($path as $position=>$elementId){
     //Utile de le calculer uniquement si pas déjà connu
     if (! array key exists ($elementId, self::$ visibility)){
        $element = $this->get ($elementId);
         //visibilité
        if ($element->show in menu hei !== self::VISIBILITY INHERITED){
           //elle n'est pas héritée
            self::$ visibility[$element->public id hei] = array ('inherited'=>$element->public id hei,
'visibility'=>$element->show in menu hei);
         }else{
           //elle est héritée, on va trouver de ou.
            if ($position === 0){
              //si c'est le premier élément, on ne cherche pas plus loin
               self::$ visibility[$element->public id hei] = array ('inherited'=>null,
'visibility'=>self::VISIBILITY NOT VISIBLE);
            }else{
              //ce n'était pas le premier élément, on prend le parent
               self::$ visibility[$element->public id hei] = array ('inherited'=>self::
$ visibility[$path[$position-1]]['inherited'], 'visibility'=>self::$ visibility[$path[$position-1]]
['visibility']);
//.... et une suite d'autres calculs
```

```
public function getTree ($origin = 0, $depth = 100, $noCache = false){
  static $cache = array ();
  if (isset ($cache[$origin][$depth])){
     return $cache[$origin][$depth];
  $toReturn = array ();
  if( !count ($headings = $this->getHeadingChildren ($origin)) ){
     return $toReturn;
  $headingsToLoad = array ();
  $toLoad = array ();
  foreach( $headings as $public id hei => $item ){
     $item->children = null; // will be set to 0 if there are no children, or an array containing them
     if ( $depth > 1 && $item->type hei=='heading'){
        //On rajoute les enfants a charger de facon groupée (pour éviter X requêtes)
        $toLoad[] = $item;
        $headingsToLoad[] = $item->public id hei;
     } else {
        //si ce n'est pas une rubrique cela ne sert à rien de chercher des enfants
        $item->children = array();
     $toReturn[] = $item;
  //demande de pré-chargement des enfants
  $this-> loadHeadingChildren ($headingsToLoad);
  //on parcours les enfants préchargés et on les rajoute a l'arbre.
  foreach ($toLoad as $item){
     $item->children = $this->getTree ($item->public id hei, $depth-1, true);
   //si on doit le mettre en cache, on le fait
  if (!$noCache){
      return $cache[$origin][$denth][$nOn]vPublished] = $toReturn
```

## [Optimisation] Cache

- On peut demander à une application de calculer 1000 fois 1000 résultats, mais pas 1000 fois 1 résultat...
- Les contenus dynamiques « statiques »
- Les contenus qui changent peu souvent
- Les contenus communs a une population

## [Optimisation] opCode



Zend Platform

XCache

Eaccelerator (Turk MMCache)

## [Optimisation] MemCache

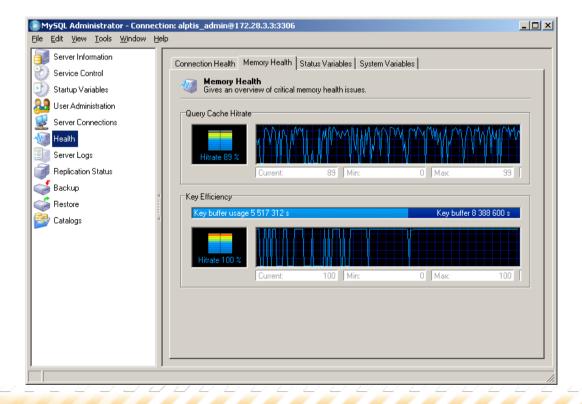
- Un serveur de « mémoire » capable de stocker des données php de n'importe quel type.
- Les données PHP complexes sont sérialisées

```
$memcache_obj = memcache_pconnect ('localhost', 11211);
if (($value = memcache_get ($memcache_obj, 'var_key_2')) !== false){
    _dump ($value);
}else{
    _dump ($value = _ppo (array ('a'=>'vA', 'b'=>'vb')));
    memcache_set ($memcache_obj, 'var_key_2', $value, 0, 30);
}
```



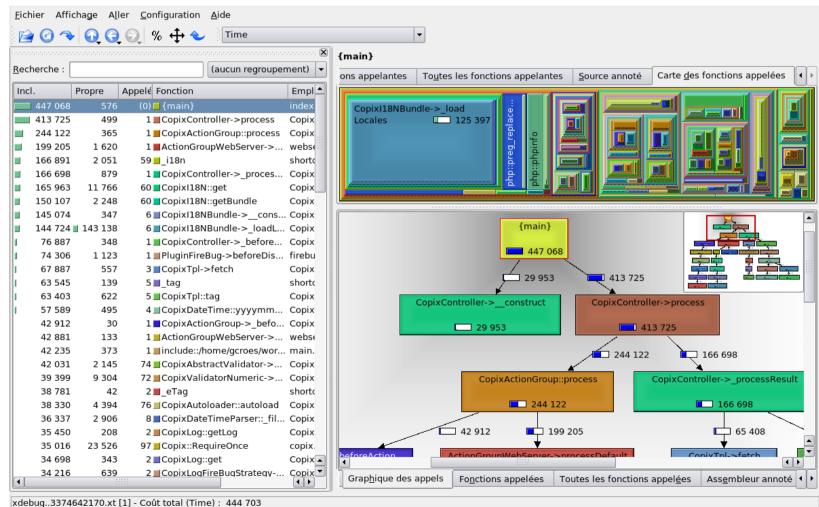
## [Optimisation] QueryCache

 Si votre site interroge votre base de données systématiquement avec les mêmes requêtes
 set global query cache size = 102400000;



## [Optimisation] Xdebug





## [Optimisation] Compression de la sortie

```
    ob_gzhandler
    if (substr_count($_SERVER['HTTP_ACCEPT_ENCODING'],
        'gzip')){
        ob_start ("ob_gzhandler");
    }
}
```

```
php.ini
    zlib.output_compression = 1
```

## Sécurité

- L'internaute est le mal
- Configuration
- SQLInjection
- Protection des sessions

## [Sécurité] L'internaute est le mal

- Ne faites jamais confiance a l'utilisateur
- Échappez les données avant de les afficher
- Ne vous basez pas dessus pour rechercher des fichiers
- La base de données « est » l'utilisateur.
- Vérifiez les données / les droits a chaque page.

## [Sécurité] Configuration



display\_errors

log\_errors

## [Sécurité] SQLInjection

- @ add\_slashes
- mysql\_real\_escape\_string
- \$\text{\text{Binding}}
  \$ps = \$db->prepare ('select \* from conference where
  sujet = :sujet');
  \$rs = \$ps->execute (array (':sujet'=>\$ GET['sujet']));

## [Sécurité] Sessions

- Un identifiant « donné » par le client
- Utilisez l'adresse IP de provenance
- Utilisez un deuxième cookie pour protéger les données

## [Sécurité] Sessions 2

```
session start ();
if (array key exists ('session key',
$ SESSION)){
   if (!isset ($ COOKIE['sessionKey']) ||
$ COOKIE['sessionKey'] !== $sessionKey){
     die ('Session Error');
}else{
   $ SESSION['sessionkey'] = uniqid
('sessionkey');
  setcookie ('sessionKey', $sessionKey);
```

## [Production] Configuration des postes

- display\_errors false
- log\_errors
- @ error\_reporting E\_ALL
- Expose php non
- max\_execution\_time limité
- OpCode actif
- Xdebug désactivé
- Eviter les extensions inutiles

## Ce qui manque à cette présentation

- phpUnderControl
- phing
- 🥮 Capistrano
- … et bien d'autres





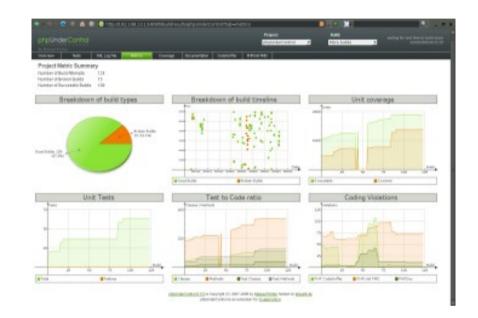
Intégration continue

**Tests** 

Métriques

**Documentation** 

Normes



# Questions?