



Clever Age
Digital Architecture

Les frameworks, essentiels dans l'écosystème PHP

Bastien Jaillot, Xavier Lacot - 09 novembre 2010



Clever Garden
Digital Landscape

**FORUM PHP
2010**



- Création en 2001 à Paris par des managers expérimentés
- Plusieurs agences en France :
 - Paris
 - Lyon
 - Bordeaux
 - Nantes
- Valeurs fondatrices : Indépendance, Veille technologique, Conviction
 - Quelques chiffres :
 - CA 2009 : 6 M€
 - Effectif au 01/11/2010 : 85 personnes
 - Notre Mission : « Concevoir des systèmes informatiques flexibles en limitant la dépendance vis-à-vis des prestataires et des éditeurs »

▪ Bastien :

- Spécialiste Frameworks et CMS
- Plusieurs contributions (Jelix, inDefero, symfony, Drupal)
- Expert technique PHP chez Clever Age
- <http://twitter.com/bastnic>

▪ Xavier :

- Expert Frameworks, développeur symfony depuis fin 2005
- Plusieurs contributions (plugins, doc, patches, etc.)
- Leader PHP chez Clever Age
- En charge du pôle d'expertise de l'entreprise
- <http://twitter.com/xavierlacot>

- Développer avec PHP : les solutions possibles
- Choisir une architecture technique
- Les critères de choix d'un framework
- Les frameworks, d'un point de vue professionnel
- Plusieurs générations de frameworks
- Les défis de l'avenir des frameworks



CITE DES SCIENCES ET DE L'INDUSTRIE - 30, AVENUE CORENTIN-CARIEU - 75930 PARIS CEDEX

Développer avec php : les solutions possibles



FORUM PHP
2010



« PHP is just a stupid tool »

« PHP is not cool, this is what you do with it which is cool »

Rasmus Lerdorf, Forum PHP 2010

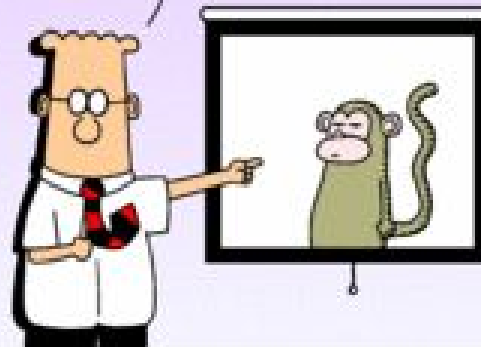
- Les projets Web couvrent un large panel
 - Gestion de contenus
 - Réseaux sociaux
 - Gestion de la relation client (CRM)
 - Communication
 - Solutions collaboratives
 - Partage de fichiers
 - Applications métiers spécifiques
- Dans chaque projet (Web), un développeur va s'arracher les cheveux

- Écarts de niveaux de complexité
 - 5 jours d'intégration html pour quelques pages Web
 - ... + de 1000 jours de développement en utilisant un framework pour construire un plateforme social « web 2.0 »
 - Très souvent, la « bonne » solution n'existe pas

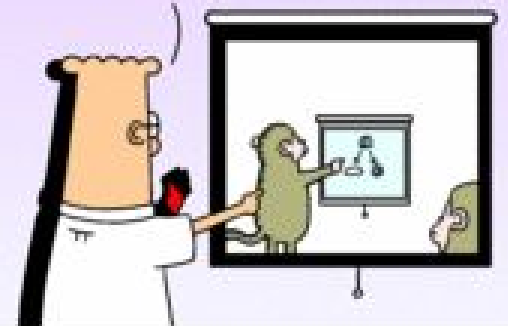
IF WE MIGRATE OUR ENTERPRISE APPLICATIONS TO THE WEB, AND OUTSOURCE OUR SALES AND PRODUCT DEVELOPMENT...



THE ENTIRE COMPANY CAN BE MANAGED BY ONE MONKEY.



PLUS A SECOND MONKEY TO LOOK AT THE POWERPOINT SLIDES FROM THE FIRST MONKEY.



- Le développeur Web a le choix entre plusieurs types de produits :
 - Frameworks
 - Solution intégrée open source
 - Solution intégrée propriétaire (mauvais)
 - Framework propriétaire (très mauvais)

- Qu'est ce que je qualifie de « solution intégrée » ?
 - CMS, forums, outils de CRM, blogs, galerie photo, etc.
 - Construit comme une solution facilement utilisable plutôt qu'une plateforme de développement
 - Principalement utilisable comme archive compressée, puis capable de se mettre à jour en toute autonomie
 - Non basée sur un framework



- ...et de « frameworks » ?
 - Non «orienté fonctionnalité »
 - Conçu comme une plateforme de développement
 - DIY – do it yourself (fait le toi-même)
 - Pas d'« effet waouw »

Agavi 

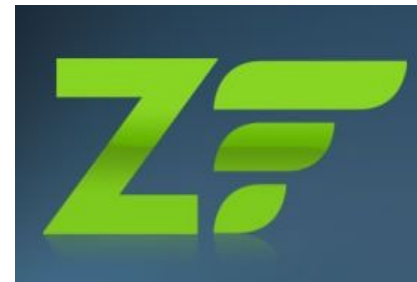
 yiiframework

symfony


CodeIgniter


JELIX

 Seagull







Choisir une architecture technique



Choisir une architecture technique

- Plusieurs considérations :

- Complexité du projet (charge de travail, fonctionnalité, scalabilité)
- Méthodologie de développement
- Y a-t-il beaucoup de personnalisation ?
 - Nouvelle fonctionnalité
 - Suppression de fonctionnalité
- Est-ce que des plugins/extensions existent, couvrent-ils tous les besoins ?

- Une solution intégrée est une bonne option :
 - Ça nous fait gagner du temps, de l'argent et évite la pression
 - Ça peut nous faire construire une meilleure application
 - Au moins le client bénéficiera de toutes les fonctionnalités que proposent l'outil
 - Ne pas réinventer la roue !
- Solutions parfaites pour les besoins communs

... le problème c'est que les clients ont rarement des besoins communs !

- Liberté d'action possible :
 - Modifier le code directement :
 - Attention aux mises à jour
 - Modifier le code directement + soumettre un ticket :
 - Plus de travail (les contributions ont besoin d'être propre et générique)
 - Pas sûr d'être acceptée (retour au point ci-dessus)
 - Créer des extensions quand c'est possible
 - Ces solutions ne sont pas conçues pour être étendues
 - Jeter cette solution et la re-coder en utilisant un framework
 - Profitez en pour définir les vrais besoins



FORUM PHP
2010



« Avec un framework on adapte l'outil aux besoins du client, alors qu'avec une solution intégrée on adapte les besoins du client à l'outil »



FORUM PHP
2010



"CMS are good to make simple websites that are mainly meant to be looked at, while frameworks are good to make complex websites that are meant to be interacted with"

Introducing The Magento Enterprise Edition

The eCommerce Platform for
Growth just got better.

Learn about
new upcoming
Enterprise
features



Magento Enterprise Edition | 1,000,000+ Downloads | Multi Store Retailing | The eCommerce Powerhouse

THE LATEST BRANDS TO GAIN COMPLETE FLEXIBILITY AND FREEDOM | [View Showcase](#)



Versatility and Freedom

Control every facet of your store, from merchandising to promotions and more. There are no limits to creativity with Magento.



Attract + Convert = Growth

Magento's SEO features and user experience will attract and convert more qualified customers, leading to business growth.



Expand Your Market

Reach more customers by creating targeted micro-sites using Magento's multi-store retailing functionality.



Competitive, Low Cost of Ownership

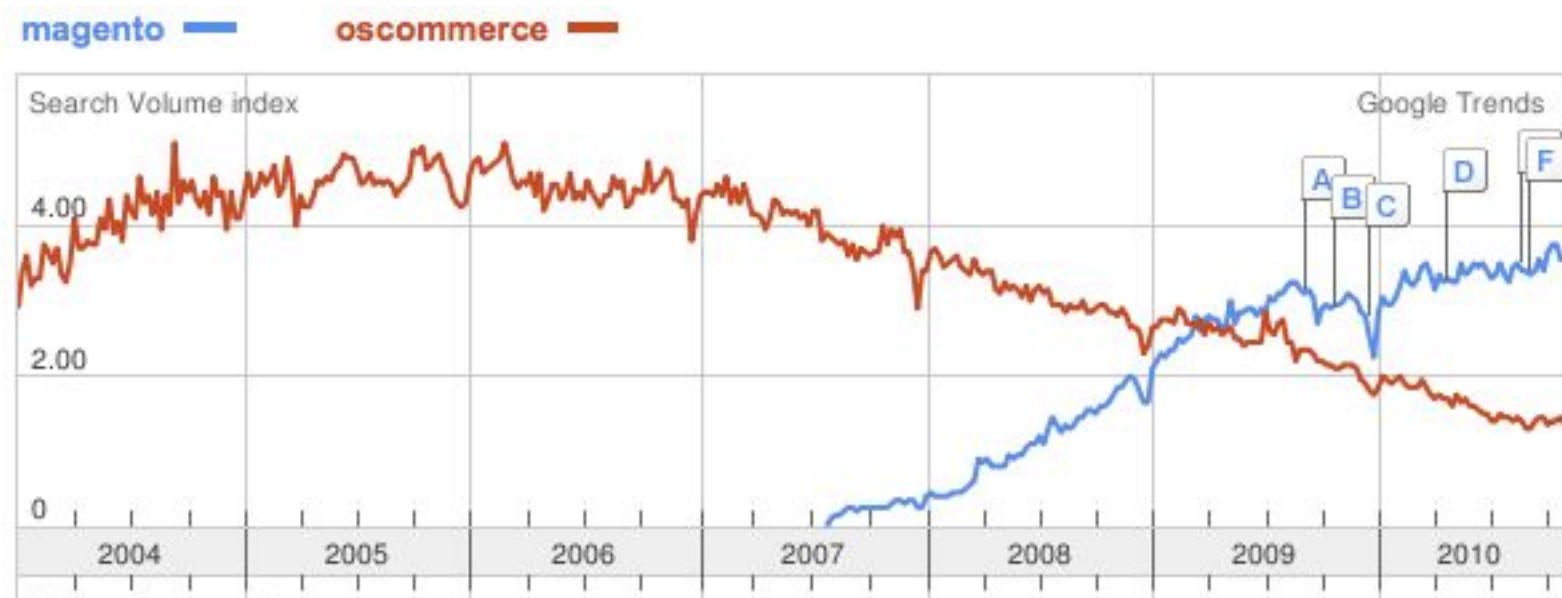
Magento's commercial open source business model yields a product that is superior for a fraction of the cost. [Learn more about our support subscription plans](#) backed by the source.



Video: Introducing the Magento Enterprise Edition



- **Magento Commerce rencontre un beau succès**
 - Basé sur Zend Framework
 - A écrasé OScommerce en 2 ans



▪ Règles d'or

- Aller à la simplicité – ne pas surcharger d'options inutiles
- Le back-office doit-être simple à utiliser
 - Éditeur WYSIWYG
 - Widgets adaptés
- Suivre les standards – « never hack up »
- Documentation !
- Ouvrir le développement : rechercher des contributeurs
- Release early, release often

- L'équation est simple :

Temps pour
bidouiller

vs.

Temps pour
construire

... Quelle est votre préférence dans la vie ?



Les critères de choix d'un framework

- Des objectifs généraux :
 - Accélérer les développements
 - Simplifier les mises à jour et les évolutions
 - Améliorer les performances (attention)
 - Favoriser l'internationalisation
 - Améliorer la qualité
 - Industrialiser la production
 - Ne plus « ré-inventer la roue »
 - Profiter du travail des autres

- Evaluation de frameworks pas évidentes
- Ne lisez pas l'avis des autres, faites vous le votre.
- A prendre en compte :
 - Risques pour l'utilisateur
 - Performances (applicative + de développement)
 - Sécurité
 - Modèle de conception
 - Outillage (CLI, journaux, scaffolding, tests)
 - Respect des standards / intégration briques externes
 - Extensibilité
 - i18n

Abstraction et Persistance des données
Multi SGBD
CRUD, Génération de code et d'interfaces
Validation des formulaires
Sécurité, authentification, autorisations
Gestion des urls (routage)
Langage de templates
Nombreux helpers
AJAX
Cache paramétrable et performant
Gestion des évolutions du modèle
Ecosystème de plugins

Gestion de l'internationalisation (i18n)
IDE
Réutilisation d'outils existants et reconnus
Documentation exhaustive
Support et maintenance
Automatisation de tâches
Tests unitaires et fonctionnels
Chargement de données de test
Licence souple d'utilisation
Web Services REST, SOAP
Montée en compétence rapide

- PHP « à l'ancienne » : mélange de PHP et de code SQL
- PHP « aujourd'hui » : utilisation d'outils d'abstraction de l'accès aux données :
 - Abstraction (ORM) : Relation Objet ↔ Relationnel
 - Persistance (DBAL) : abstraction de la base de données
- Le principe : des classes représentent les tables de la base de données et les entrées qu'elles contiennent

table Post	
id	integer(11)
title	varchar(100)
body	longtext
category_id	integer(11)
author	varchar(30)



```
class Post extends BasePost
{
}

$post = new Post();

$post->setTitle('Ceci est un message');
$post->setBody('Corps du post');
$post->setCategoryId(12);
$post->setAuthor('Michel');
```



FORUM PHP
2010



Abstraction et persistance des données

- Deux ORMs majeurs :
 - Propel,
 - Doctrine, plus récent, plus actif,
- Des solutions historiques :
 - PEAR_DB, AdoDB, Creole, etc.
- Ou anecdotiques :
 - dORM, junction, etc.
- Méfiez vous des performances (modes d'hydratation, cache de requêtes / resultsets), etc.
- Exigez un ORM !



Sauf cas très particulier (pas le votre),
refusez toujours les frameworks de
développement closed source.

Sinon, on se voit après la conférence et vous me faites un chèque en blanc ?

- Taux d'adoption
- Documentation
- Communauté
- Processus de développement
- Intégration dans le SI
- Il faut se faciliter la vie ! Privilégier :
 - les projets maintenus et fortement documentés
 - les architectures ouvertes (plugins, modules etc.)
- Éviter :
 - les choix à court terme
 - Le framework du copain ou du premier intégrateur venu





Les frameworks, d'un point de vue professionnel

- Plusieurs aspects intéressants en environnement professionnel :
 - Intégration continue
 - Gestion de sources
 - Mutualisation
 - Procédures de déploiement
 - Méthodologies de travail

Poste
développeur

Serveur de
gestion de
sources

#628

(last author before build: bclay)

28/01 16:13:06

Projet CommunautaireTrunk

- Espace de travail
- Derniers artefacts obtenus avec succès
 - fnac-communautaire-build628.tar.gz
- Changements récents
- Derniers résultats des tests (aucun échec)

Liens permanents

- Dernier build (#628), il y a 20 mn
- Dernier build stable (#628), il y a 20 mn
- Dernier build avec succès (#628), il y a 20 mn
- Dernier build en échec (#620), il y a 20 h

Tendance des résultats des tests



Chuck Norris doesn't need to know about class factory pattern. He can instantiate interfaces.



Serveur
d'intégration
continue

- Objectif : pouvoir maîtriser la qualité du projet au cours de son développement et de son cycle de vie
- Tester son application :
 - coûte du temps au début,
 - en fait gagner ensuite.
 - nécessite de la rigueur et de la constance
- Plusieurs outils :
 - PHPUnit : la référence
 - Lime (fourni avec symfony)
 - Atoum ?
- Tests fonctionnels : Selenium,



- Charges plus faciles à évaluer :
 - Prévoyez du temps d'initialisation
 - Recensez les tâches
 - Prévoyez du temps de recette
 - Examinez les contributions utilisables
 - Plugins, modules
 - Produits tiers
- Charges proportionnelles à la complexité du projet
 - Mais possible de capitaliser

- Méthodes agiles. Ex. Scrum :
 - Qu'est ce que j'ai fait hier, qu'est ce je fais aujourd'hui ?
 - Les difficultés rencontrés (en attente d'infos ? Difficultés techniques ?...)
 - Objectifs :
 - Rythme durable
 - Informe de l'existant et donne une vision de l'avancement (visibilité)
 - Améliorer l'esprit d'équipe (partage de connaissance, questions sur les besoins...)

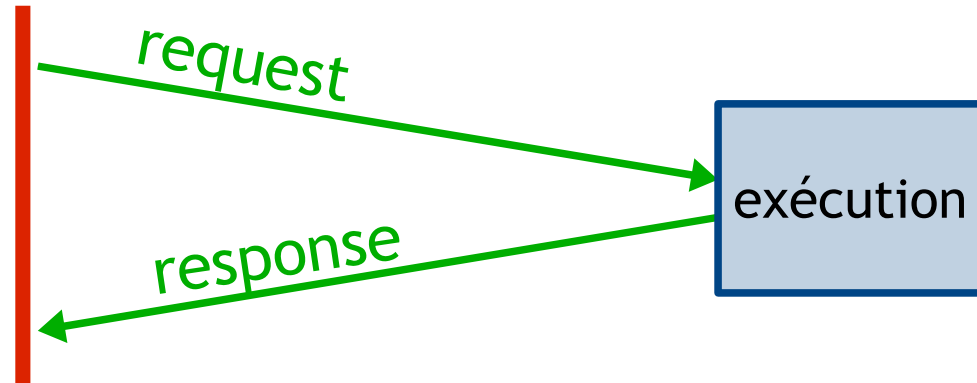
livrables
chifffrage
revue de sprint
livraisons fréquentes
étude du besoin
réunions de planning
scrum
sprint du produit priorisé (backlog)



Plusieurs générations de frameworks

- Le passé : des librairies

- Request
- Execution
- Response



- Le présent : RAD / MVC

- Symfony, Zend Framework, CakePHP, Code Igniter, Yii...

- Le futur

- Plus découplé ?
- Distribué ?
- Cloudifié ?

- Peu de frameworks jusqu'en 2005 (RoR)
- **PEAR**
 - Débuté en 1999
 - Collection de bibliothèques de haute qualité (à l'époque) pour promouvoir la réutilisabilité en PHP
 - Encore actif
- **Smarty**
 - Moteur de template, 1.0 en 2004
 - Encore actif

▪ Frameworks MVC

- Zend Framework
- symfony
- Cake PHP
- Code Igniter
- Yii
- Jelix
- Pluf
- etc.



- Version 1.11.0 (version 1.0 en 2007)
- Bibliothèque de composant accompagné d'une brique MVC
- Soutenue par la société Zend
- De nombreux IDE et bundles aident à son développement
- Documentation très complète
- Composants à jour sur les APIs les plus connues (Google, Amazon, Flickr)
- Conférence demain 10/11/2010 à 9h45 par Gauthier Delamarre



- Version stable 1.4 (version 1.0 en 2007)
- Framework « Full stack » MVC
- Soutenue par la société Sensio
- Documentation orientée sous forme de livres
- La star : le « Jobeet », génial pour apprendre le framework
- Symfony2 en route...
- Conférence demain 10/11/2010 à 9h présentée par Hugo Hamon

- Version actuelle 1.3.5 (version 1.0 en 2006)
- Compatible PHP4
- Implémentation de Ruby on Rails en PHP
- Conférence demain 10/11/2010 à 11h par Aurélien Vairet et Pierre Martin

- Version 1.2 (version 1.0 en 2008)
- Développement communautaire
- Projet français
- Documentation complète
- Système d'installation et de mise à jour de modules
- Projet présenté par Laurent Jouanneau et Olivier Demah

- Fin du 1er cycle de vie des frameworks
- Bientôt :
 - Zend framework 2
 - Symfony2
- Nouvelles tendances architecturales
 - Injection de dépendance
 - Utilisation des namespaces
 - Retrait de singletons
 - Plus légers
- Abandon progressive des frameworks n'évoluant plus



Les défis de l'avenir des frameworks

▪ Les performances

- Les perfs actuelles ne sont pas satisfaisantes
- Sites aux fonctionnalités de plus en plus riches
- Solution actuelle : multiplication des machines

▪ Solutions envisagées

- Compilation ? (cf. hiphop-php)
- Meilleur fonctionnement distribué (cloud)
- ESI (Edge Side Includes)
- Tirer profit de bases NoSQL

- La pénétration du marché reste faible

zend framework 1.00 drupal 8.50 wordpress 28.5



- Adoption par des projets OSS (PHPBB, des projets de CMS ?, etc.)
- Attention : maîtrise des cycles de vie des projets

- **Défis :**
 - Plus grande adoption « grand public »
 - Adoption par des projets OSS
 - PHPBB avec Symfony2,
 - des projets de CMS ?
 - Interopérabilité des contributions ?
- **Attention : maîtrise des cycles de vie des projets**
 - Changer de version majeure → réécrire le produit ?
 - Exemple de Magento

- **Intégration de nouveaux types d'outils « modernes »**
 - Bases NoSQL – MongoDB, Redis, Cassandra, etc.
 - Indexes (recherche) – Lucene Solr, ElasticSearch, etc.
 - Messaging Queues (AMQP, XMPP)
 - Debuggers plus poussés
 - Industrialisation plus facile
 - Cloud
- **Choix d'une nouvelle architecture possibles**
 - BC break
 - Attention à ne pas « JAVAiser » PHP



FORUM PHP
2010



Questions



- Récompense des projets Web innovants
 - HTML5
 - OpenData
 - Mobilité
 - etc.
- Appel à candidatures :
candidat_awards@clever-age.com
- Bientôt sur <http://awards.clever-age.com/>