

- Frameworks PHP

Retours d'expériences

Cas d'utilisation

Critères de choix

Panorama



- Définition (Framework Applicatif)
- Éléments de conception
- Engouement soudain
- Apports & Risques
- Critères de choix
- Panorama
- Avenir des frameworks PHP



- Un ensemble de classes qui coopèrent et permettent des conceptions réutilisables dans des catégories spécifiques de logiciels.
- Un framework définit la structure globale de l'application, le partitionnement en classes et objets, la façon dont ces classes collaborent entre elles.

- Le framework permet au développeur de se concentrer sur les particularités de l'application à développer, car le framework maîtrise les décisions de conception courantes dans son domaine



- Un cadre de travail
- Des outils
- Une méthodologie
- Des normes de développement
- Des recommandations



Dans les contextes WEB

- Authentification & Gestion des profils
- Indépendance vis à vis de l'interface
- Accès aux données (indépendance et rapidité)
- Montée en charge
- Internationalisation
- Réutilisation du code / capitalisation
- Paramétrage



- API / Système d'échange inter-modules
- Sécurité (cross scripting, code injection, sql injection, vols de session)



Le cas des frameworks spécialisés tels les CMF

- Cycles de publication
- Rubriquage
- Arborescence de navigation
- Systèmes d'installation
- Espaces collaboratifs
- ...



- Une direction « entreprise » du langage
 - Il y a 4 ans, image négative.
 - Aujourd'hui vraie alternative, solution comme une autre
- Services informatiques prescripteurs
 - Besoin de reconnaissance des compétences
 - Facilité d'utilisation, d'installation, faibles coûts
- Projets pilotes « non critiques » concluants
 - Budgets plus faibles => tendre vers l'efficience

- Un marché de plus en plus concurrentiel
 - Crise informatique, Multiplicité des acteurs
- Vers l'industrialisation des développements
- Axe de différenciation



- Encouragement à la qualité logicielle
- Recommandation / utilisation des Designs Patterns
- Organisation commune
 - Restriction dans l'emplacement des traitements
 - Restriction dans l'emplacement des fichiers
- Forcent à la réflexion « post-projet »

Attention: Ils n'en sont toutefois pas garants



- Maintenance
 - Identification rapide des portions de code qui rentrent en jeu
- Granularité du code et des équipes
 - Découpage en modules, en fonctionnalités
 - Organisation commune
- Gains de temps en conception
 - « Pré-conception » technique, rôle des acteurs

- Réalisations d'actions systématiques
- Maîtrises des comportements
- Possibilité de gérer une cinématique



- Capitaliser
 - Organiser
 - Permettre la génération / généralisation de code
 - Assurer une compatibilité / interopérabilité des développements inter-projets
 - Aider le découpage applicatif
- => Faciliter le travail en équipe
- Montée en charge
 - Accélération des développements

- Langage simple et souple
 - Langage de script
 - Qualité des scripts existants très diverse
- Du site vitrine au langage intégré au SI
 - Multi-base
 - Léger
 - Rapide
- Accompagner la montée en puissance / la volumétrie croissante des applications

- Complexité apparente
- Des réponses à des besoins différents
- Refactoring libre
 - Soucis d'amélioration
 - Pas de contrainte de compatibilité
- Un besoin spécifique et simple



- Ancienneté / Durée dans le temps
- Système de modules / plugins
- Internationalisation
- Caches / Rapidité
- Documentation
- Langage natif ?
- Visibilité



- Nombre d'intervenants (développeurs, utilisateurs, Organismes de soutien)
- Références **Affinité globale**
- Licence
- Installation / Déploiement

- Maîtriser tous les aspects
- Connaissance parfaite de ses besoins
 - Meilleure spécialisation
- Choix de son environnement de travail



- Maîtriser l'évolution
- Documentation
- Tout faire/refaire (homemade)
 - Effet tunnel (masquer l'évolution globale des outils, oublier l'objectif initial)
- Définir une formule qui fonctionne
- Complexité gratuite



- Horde
- Seagull
- Xaraya
- Blueshoes
- PhpMVC
- Copix
- Pear ?



- Framework technique, non applicatif



- <http://www.horde.org>
- Langage natif : Anglais
- IMP, Chora, Hermes, Jonah, Kronolith, Mnemo, Nag, Rakim, Sork, Trean, Turba, Whups

- Forte intégration avec le framework technique Pear
- Ancienneté et notoriété



- <http://seagull.phpkitchen.com>
- Langage natif : Espagnol
- Documentation en ligne (Wiki)
- Authentification, Cache, Internationalisation, Indépendance vis à vis de l'interface (Flexy)
- Blog, Galleries d'images, CSV Manager, RDF agregator, Agenda



- <http://www.phpmvc.net>
- Utilisation du framework technique Pear
- MVC 2
- Documentation en ligne

- Implémentation du framework Struts en PHP.



- <http://www.copix.org>
- Langage natif : Français
- Authentification & gestion des permissions, Cache, Internationalisation, Indépendance de l'interface (Smarty ou PHP), Accès aux données, Paramètres, Installation avec dépendance.

CMS basé sur un système de portlets (sondages, nouvelles, agenda, moteur de recherche, formulaires, ...)



- Mimer les frameworks des autres langages ?
 - Exemple de Struts
 - Risque d'effacer les particularités de PHP
 - Bénéfice de l'existant, faible prise de risque
- Adapter les concepts en profitant de la puissance de PHP
 - Prise de risque plus forte
- PHP 5, de nouvelles possibilités (destructeurs, exceptions, XML, autoloader)

