

A photograph of four people (three women and one man) looking at a laptop screen. They are all smiling and appear to be in a collaborative work environment. The woman on the far right is pointing at the screen. The background is slightly blurred, showing an office setting.

Alter Way

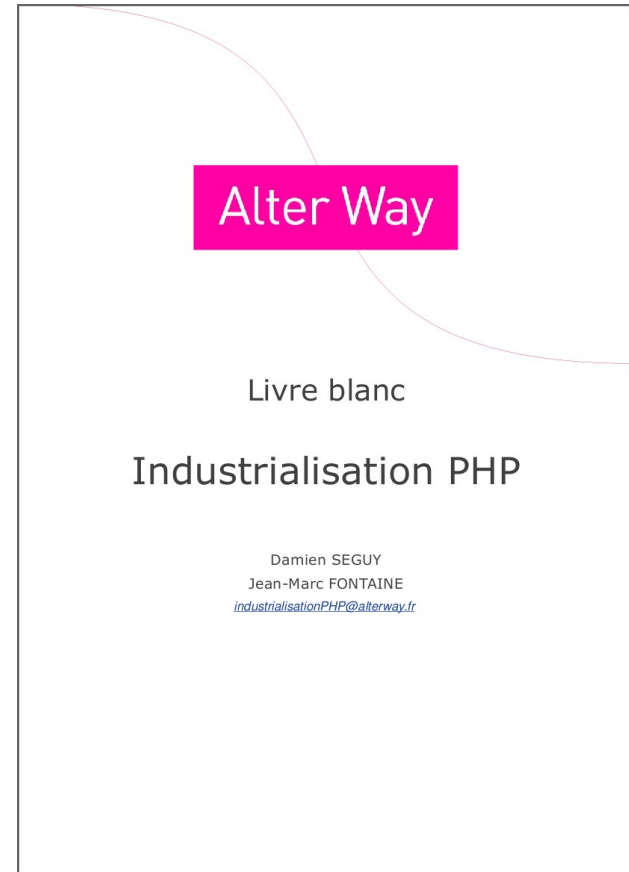
**Les revues de code ou
comment faire fructifier
son capital humain**

Qui suis-je ?

- **Jean-Marc Fontaine**
- **Consultant pour Alter Way Consulting**
- Responsable du centre de compétences PHP pour Alter Way Solutions
- Formateur / Professeur vacataire en université
- Membre Actif de l'AFUP
- Auteur du blog Industrialisation-PHP.com
- Co-auteur du livre blanc « Industrialisation PHP »

Livre blanc « Industrialisation PHP »

- **Co-écrit avec Damien Seguy**
- **Panorama des outils et méthodes d'industrialisation des développements PHP**
- **Publié en octobre 2009**
- **2 500 téléchargements**



Téléchargement : <http://www.alterway.fr/publications/livre-blanc-industrialisation-php>
Blog : <http://www.industrialisation-php.com/>

L'erreur est humaine



**Les développeurs font des erreurs, c'est inévitable.
La vraie question est quand souhaite-t-on les découvrir ?
En développement, en recette ou en production ?**

Revue de code : kezaiko ?

- **Une revue de code consiste à examiner le code de quelqu'un d'autre à la recherche de défauts ou d'améliorations potentielles**
- **La vérification (contrôle de la qualité pendant le développement) est plus efficace, et donc moins coûteuse, que la validation (contrôle de la qualité après le développement).**
- Proximité avec les méthodes agiles (Pair programming, responsabilité collective, courage)

Pas limitées au code source

- **Les revues ne sont pas limitées au code, on peut les appliquer également aux documents importants d'un projet comme les spécifications, les documents de conception, etc.**

Il n'y a pas des outils d'analyse pour ça ?

- **Complémentaires**
- **Adaptés aux problèmes de syntaxe et d'optimisation subtile**
- **Pas adaptés aux problèmes fonctionnels ou de logique**
- **Un humain peut identifier :**
 - les messages d'erreur peu clairs
 - les commentaires inappropriés
 - les valeurs en dur (Nombres magiques)
 - les manques de factorisation

Et les tests ?

- **Les tests automatisés (unitaires, fonctionnels, etc.) et manuels n'indiquent rien de la qualité et de la maintenabilité du code.**
- **Les tests trouvent les symptômes tandis que les revues de code trouvent les causes des problèmes.**

Buts

- **Amélioration de la qualité du code**
 - Découverte de bogues
 - Qualité de la conception
 - Application des bonnes pratiques
 - Respect des conventions
 - Suppression des redondances
 - Élimination des incohérences
- **Vérification de la conformité**
- **Vérification de l'exhaustivité**

Bénéfices indirects

- **Partage de la connaissance**
- **Formation des juniors**
- **Recherche de consensus**
- **Amélioration de la maîtrise collective du code**
- **Émergence d'idées neuves**

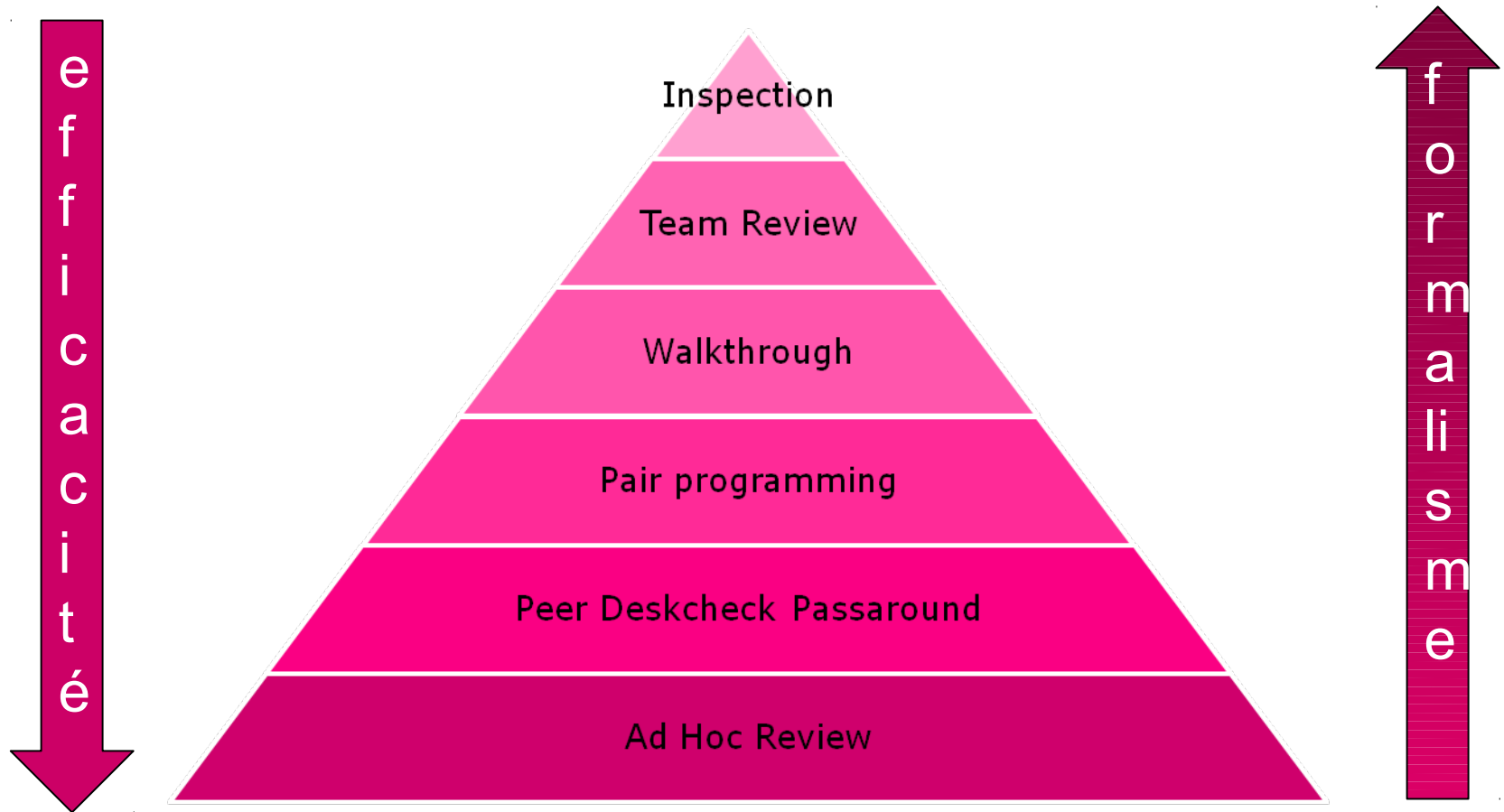
Objections habituelles

- **Coût**
- **Perte de temps**
- **Freins humains**
 - Fainéantise
 - Ego
 - Conflits personnels
- **Difficultés d'organisation**
- **Méthode non exhaustive**

C'est un truc expérimental ?!

- **Les revues de code sont pratiquées par tous les acteurs importants**
 - Google
 - Cisco
 - Intel
 - NASA
 - BNP Paribas
 - UPS
- **Chez Google rien n'est commité sans être revu au préalable.**

Les types de revues de code



Organiser une revue de code



Quand effectuer une revue de code ?

pre-commit

- Assure la qualité du code commité
- Possibilité de blocage si les auditeurs ne sont pas disponibles

post-commit

- Pas de blocage
- Présence dans le dépôt de code pas encore revu
- Possibilité de mitiger ce problème en utilisant une branche dédiée

Planifier une revue de code

- **Entre 3 et 7 personnes**
- **Rôles**
 - Auteur
 - Inspecteur
 - Modérateur
 - Lecteur
 - Secrétaire
 - Vérificateur
- **Les rôles sont cumulables sauf pour l'auteur qui ne peut éventuellement être que lecteur**
- **Choisir un endroit calme**
- **L'équipe doit rester isolée durant toute la revue**

Sélectionner le code à étudier

- **Systématique**
- **À la demande du développeur**
- **Parties problématique de l'application**
- **Couverture de code**
- **Expérience**
- **Hasard**

Préparer une revue

- **Réunion de présentation**
 - Informelle
 - Présente rapidement le code qui sera revu et les objectifs de la revue de code
 - Peut être supprimée si l'équipe est familière du code
- **Définition des règles, standards et spécifications en vigueur**
- **Le lecteur doit se familiariser avec le code**
- **Les inspecteurs doivent étudier le code à la recherche de problèmes et d'opportunités d'optimisations**

Déroulement d'une réunion de revue

Le modérateur conduit la revue



```
graph TD; A[Le modérateur conduit la revue] --> B[Le lecteur décrit avec ses mots le code par petits blocs]; B --> C[Les inspecteurs font leurs commentaires, posent des questions, proposent des améliorations]; C --> D[Le secrétaire note toutes les remarques dans un journal des défauts]; D --> E[Les inspecteurs notent la qualité du code audité];
```

Le lecteur décrit avec ses mots le code par petits blocs

Les inspecteurs font leurs commentaires, posent des questions, proposent des améliorations

Le secrétaire note toutes les remarques dans un journal des défauts

Les inspecteurs notent la qualité du code audité

Après la revue de code

- **L'auteur modifie son code en fonction du journal des défauts**
- **Le modérateur rédige un compte-rendu de revue**
- **Le vérificateur s'assure que le code a été retravaillé comme convenu**
- **Livrables**
 - Code retravaillé
 - Journal des défauts
 - Compte-rendu de revue
- **Possibilité de générer des statistiques avec le temps**

Évaluer le fruit des revues de code



Region	East	West
Y1 Qtr1	4,562	7,485
Y1 Qtr2	6,076	9,959
Y1 Qtr3	23,229	28,625
Y1 Qtr4	9,505	13,654
Y2 Qtr1	17,336	31,557
Y2 Qtr2	9,289	17,718
Y2 Qtr3	10,527	24,067
Y2 Qtr4	80,524	133,064
Previous Year		
Y1 Qtr1	7,632	3,250
Y1 Qtr2	9,635	4,141
Y1 Qtr3	30,241	12,582
Y1 Qtr4	10,585	4,457
Y2 Qtr1	31,017	13,208
Y2 Qtr2	16,264	6,926
Y2 Qtr3	21,974	9,357
Y2 Qtr4		



Region	Y1 Qtr1	Y1 Qtr2	Y1 Qtr3	Y1 Qtr4	Y2 Qtr1
North	29,388	23,688	22,589	45,506	
South	31,251	24,754	23,790	47,554	
East	13,078	10,541	10,052	20,250	
West	23,176	19,282	18,387	37,042	

Sales Growth (%) Compared to Previous Year	Qtr1	Qtr2	Qtr3	Qtr4
North	12.30	15.99	12.30	2.3
South	7.82	11.60	12.02	-1.3
East	19.19	32.34	101.35	55.8
West	70.38	64.56	12.50	11.1

Top Sales Persons	Name	Sales

Évaluer le fruit des revues de code

- **L'efficacité des revues doit être évalué par rapport au temps gagné et pas au nombre de défauts trouvés.**
- **Éviter les mesures individuelles**

Mesures de base

Taille

- Nombre de lignes à étudier
- Nombres de lignes étudiées

Effort

- Planification
- Présentation
- Préparation
- Revue
- Modification

Défauts

- Nombre de défauts majeurs trouvés
- Nombre de défauts mineurs trouvés
- Nombre de défauts majeurs corrigés
- Nombre de défauts mineurs corrigés

Divers

- Durée de la revue
- Nombre d'inspecteurs
- Évaluation de la qualité du code

Mesures avancées

Défauts

- Nombre de défauts par unité de travail inspectée
- Nombre total de défauts trouvés
- Nombre total de défauts corrigés

Effort

- Nombre d'heure consacrées à la revue dans son ensemble
- Nombre d'heure moyen pour trouver un défaut (hors correction)
- Nombre d'heure moyen par unité de travail inspectée

Couverture

- Pourcentage du code prévu qui a été effectivement revu
- Pourcentage de défaut considérés comme majeurs

Rythme

- Nombre moyen d'unités de travail couvertes par revue
- Nombre moyen d'heures de préparation individuelle par unités de travail
- Nombre moyen d'heures pour corriger et vérifier un défaut

Outiller ses revues de code



Outiller ses revues de code

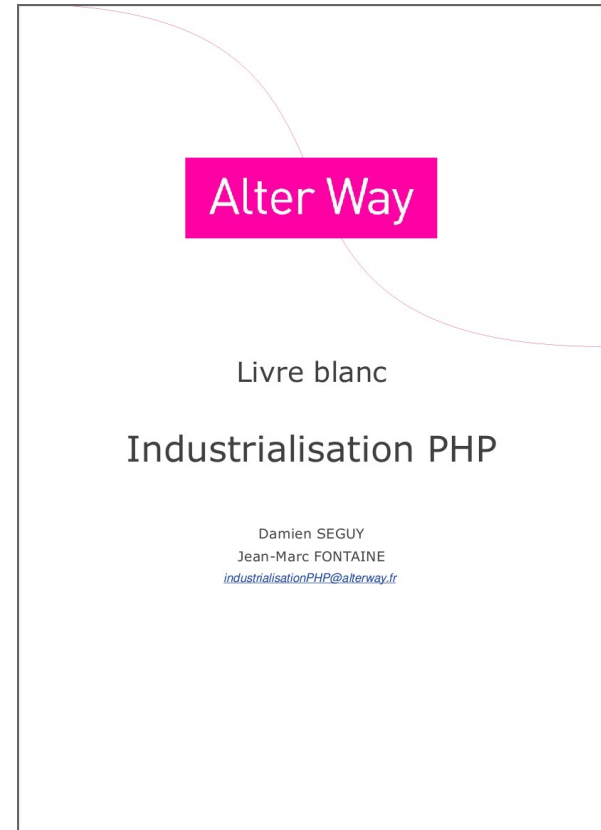
- **Outils Open Source**
 - Review Board
 - Rietveld
- **Outils commerciaux**
 - SmartBear Code Collaborator
 - Atlassian Crucible

10 bonnes pratiques

- **Ne pas étudier plus de 300 lignes à la fois**
- **Adopter un rythme de 300 à 500 lignes étudiées par heure**
- **Ne pas dépasser 90 minutes pour une revue**
- **Les inspecteurs doivent étudier le code avant la réunion de revue**
- **Établir des objectifs quantifiables et recueillir des mesures**
- **Utiliser des checklists**
- **Vérifier que les problèmes trouvés sont effectivement corrigés**
- **Développer la culture de la revue de code**
- **Jouer sur la pression sociale**
- **Éviter le sentiment de surveillance**

Merci !

- **Me contacter :**
jean-marc.fontaine@alterway.fr
- **Donnez votre avis sur cette conférence :**
<http://joind.in/2075>
- **Me suivre sur Twitter :**
(@jmfontaine)
- **Des questions ?**



Présentation : <http://www.slideshare.net/jmf/>

Livre blanc : <http://www.alterway.fr/publications/livre-blanc-industrialisation-php>

Blog : <http://www.industrialisation-php.com/>