

La revue de code

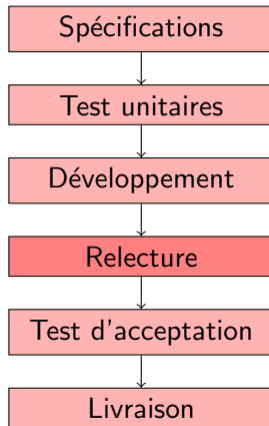
Kevin Hagner

9 septembre 2014

Rendre un code fiable

Différentes vérifications :

- ▷ Tests unitaires.
- ▷ Tests d'intégrations.
- ▷ Revue de code.



Inspection en comité

Faire des réunions concrètes où tous les développeurs examinent un code source commun.

Avantages notables

- ▶ Assiduité garantie de tous les collaborateurs.
- ▶ Très didactique.

Défauts notables

- ▶ Souvent partielle.
- ▶ Peut être pénible pour le développeur « évalué ».
- ▶ Long, donc cher.

Notification post-commit

Un message contenant les modifications est envoyé automatiquement après chaque commit à tous les développeurs du projet.

Avantages notables

- ▷ Rapide.
- ▷ Indéférent de la localité des développeurs.

Défauts notables

- ▷ Pas sûr : le mail peut être mis de côté.
- ▷ Le contexte est manquant.

Notification pré-commit

La modification est soumise à relecture et réellement intégrée qu'une fois validée par un nombre défini de développeurs.

Avantages notables

- ▶ L'intégralité du code doit être relue.
- ▶ Tous les développeurs ont une vision assez globale du projet.

Défauts notables

- ▶ Rend le processus d'intégration lent.
- ▶ Fastidieux dans le cas de commits à haute fréquence.

Analyse par dessus l'épaule

Le développeur fait valider son code par un relecteur avant le commit.

Avantages notables

- ▶ Plutôt rapide.
- ▶ Ne nécessite pas d'outils particulier.

Défauts notables

- ▶ Relecteurs limités.
- ▶ Relecture dirigée par le développeur.
- ▶ Informelle, donc potentiellement négligée.
- ▶ Peut gêner le relecteur qui faisait quelque chose d'autre.

Programmation en binôme

Deux développeurs codent sur le même PC, donc en concertation continue.

Avantages notables

- ▶ Revue de code en continue.
- ▶ Efficace pour éviter les gros problèmes de conception.

Défauts notables

- ▶ L'examineur est très impliqué dans le code.
- ▶ Coûteux, car deux développeurs sur la même chose.

Difficultés

- ▶ L'égo des développeurs.
- ▶ Trouver une méthode efficace (date, durée des réunions, logiciels, etc.)
- ▶ Manque de soutien de la hiérarchie.

Erreurs fréquentes

- ▷ Relecture non assidue.
- ▷ Reporter les revues car considérées comme non-prioritaires.
- ▷ Revue effectuée toujours par la même personne.
- ▷ Revue trop grosses.
- ▷ Non-transmission du compte-rendu du relecteur au développeur.
- ▷ Rigueur décroissante par le développeur qui pense que la réflexion sur son code se fera après coup.

Bonnes pratiques

- ▷ Documenter les procédés de relecture.
- ▷ Y consacrer du temps dans les estimations de prix.
- ▷ Mettre en place des checklists.
- ▷ Faire une revue de son propre code avant le commit.

Outils

- ▶ Grep.
- ▶ Phabricator ? Review Board ?
- ▶ Code Smell : PHP Code Sniffer, Jenkins, Sonar, etc.

Des questions ?



Contactez - moi sur spyzone.fr