

# INF2015 – Développement de logiciels dans un environnement Agile

Révision de code

Jacques Berger

# Objectifs

Introduire une pratique de qualité

# Prérequis

Aucun

# Révision de code

L'analyse du code par un autre développeur dans le but de détecter des erreurs de programmation

# Révision de code

## Objectifs

Détection des erreurs  
Communication dans l'équipe  
Transfert de connaissances  
Formation

# Détection des erreurs

Différentes personnes voient différentes choses

Une erreur détectée lors de son écriture est nécessairement moins coûteuse à corriger

On veut en détecter le plus possible, le plus rapidement possible



# Communication dans l'équipe

On ne peut pas tout savoir, une autre personne peut avoir une vision différente (complémentaire) de la solution

Deux têtes valent mieux qu'une

Oblige les développeurs à communiquer entre eux, la communication c'est bon pour l'équipe

# Transfert de connaissances

La personne qui révisé le code connaît maintenant comment ça fonctionne

Évite qu'il n'y ait qu'une seule personne qui puisse travailler sur un bout de code



# Formation

Lorsqu'un sénior révise notre code, ses observations peuvent être très formatrices

Lorsqu'on révise le code d'un sénior, on peut apprendre plusieurs trucs intéressants

# Projet

La révision de code peut prendre un certain temps

Ce temps constitue un investissement si la révision de code est faite avec agilité

# Inspection formelle

Une forme de révision de code très structurée

## Déroulement

- Le développeur écrit du code

- Le code doit être révisé par plusieurs personnes

- On fait une réunion pour lister les problèmes

- Le développeur corrige les problèmes

- Si les changements sont importants, on peut refaire une (ou plusieurs) autre révision

# Inspection formelle

Nécessite beaucoup de temps

- L'organisation de la réunion (disponibilité)

- La préparation avant la réunion

- La réunion elle-même

Coûts de développement plus élevés

Le délai de la réunion nuit à l'avancement du projet

# Inspection formelle

Chaque participant doit orienter sa recherche sur un point en particulier (ex. Thread-safety)

Très bon taux de détection d'erreurs (60 à 65% des défauts)

Méthode efficace mais peu souple, pas très agile



# Révision informelle

Méthode moins formelle que l'inspection formelle

## Déroulement

- Le développeur écrit du code

- Un autre développeur (ou 2) révise le nouveau code

- Le développeur corrige les erreurs



# Révision informelle

Ne nécessite pas de réunion formelle avec beaucoup de participants, une discussion entre deux développeurs suffit

Le délai pour la révision est moins long puisqu'un seul développeur est impliqué

Le réviseur est choisi en fonction de son expertise et de sa disponibilité

# Révision informelle

Moins de délais, moins de coûts

Méthode favorisée dans les techniques agiles

Le taux de détection d'erreurs peut être assez bon  
(50% des défauts)

# Pair programming

Programmer à 2 personnes

Technique exigeante ayant beaucoup d'avantages et quelques inconvénients

Révision gratuite car elle est faite en même temps que l'écriture du code

Méthode privilégiée par eXtreme Programming

# Checklist

Programmer, c'est compliqué. Il faut avoir plusieurs préoccupations durant le développement

On peut se faire une liste de choses à vérifier durant une révision, une checklist, pour en oublier le moins possible

# Checklist

On peut mettre l'accent sur :

La performance

La validité des résultats

La sécurité

La lisibilité

# Checklist

La gestion des erreurs :

- Exceptions

- Codes de retour

- Erreurs systèmes

- Erreurs de bases de données

- Erreurs de communication (web, réseau)

- Récupération en cas d'erreur



# Checklist

Thread-safety

Normes de codification

Conventions de design

Qualité du design

Documentation

# Méthodes

Il existe plusieurs façons de faire

Certains préfèrent une révision électronique,  
d'autres préfèrent réviser une version imprimée

Certains préfèrent réviser avant de faire un  
commit, d'autres préfèrent réviser après

# Méthodes

Certains préfèrent réviser un diff, d'autres préfèrent réviser le fichier au complet

L'idéal, c'est d'utiliser un outil qui permet toutes ces possibilités, ainsi le réviseur peut choisir selon ses préférences

# Discussion

Peut-on faire de la révision de code lorsqu'on est le seul développeur de l'entreprise?

# Norme

Méthode normalisée pour effectuer une révision de code :

IEEE STD 1028-2008

IEEE Standard for Software Reviews and Audits

# Outils électroniques

Le diff de github

FishEye



# Conclusion

Pratique recommandée

Faites réviser votre code mais révisiez également le code des autres

Effectuer une révision de code, c'est une activité exigeante

Ça demande beaucoup de concentration

# Conclusion

Ne désespérez pas au début, l'on devient bon avec l'expérience et l'expérience s'acquiert lentement