

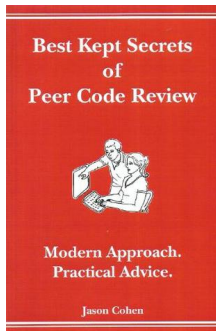
Systemes de la revue de code

Sergey Kirgizov

ESIREM

Littérature

- ▶ Wiki https://fr.wikipedia.org/wiki/Revue_de_code
- ▶ Livre de Jason Cohen, 2006



- ▶ "A curated list of tools, articles, books, and any other resource..."
<https://github.com/joho/awesome-code-review>

Exemples

OpenStack est un ensemble de logiciels open source permettant de déployer des infrastructures de cloud computing



openstack®

<https://review.openstack.org/>

Développement d'Android

Android est un système d'exploitation mobile, basé sur le noyau Linux et développé actuellement par Google. Lancé en juin 2007 à la suite du rachat par Google en 2005 de la startup "Android Inc" fondé à Palo Alto, en Californie, en octobre 2003 par Andy Rubin, Rich Miner, Nick Sears et Chris White



<https://android-review.google.com/>

Développement d'OpenMW

OpenMW est une réimplantation libre et moderne du moteur de jeu du jeu de rôle Morrowind permettant à qui possède les données originale du jeu d'y jouer. OpenWM a été écrit à partir de zéro, à l'aide de rétro-ingénierie sur l'implantation originale.



https://gitlab.com/OpenMW/openmw/merge_requests

Développement de MediaWiki

MediaWiki est un moteur de wiki pour le Web. Il est utilisé par l'ensemble des projets de la Wikimedia Foundation, ainsi que par l'ensemble des wikis hébergés chez Wikia et par de nombreux autres wikis.



<https://gerrit.wikimedia.org/>

Développement de la Dopamine

Très récent, 26 Août 2018!!!

Dopamine *c'est un framework pour prototypage rapide d'algorithmes d'apprentissage par renforcement*



<https://github.com/google/dopamine/pulls>

Développement du Coq

Coq est un assistant de preuve fondé sur le calcul des constructions (CoC abrégé en anglais) introduit par Thierry Coquand.

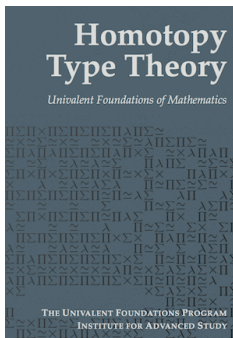


<https://github.com/coq/coq/pulls>

Rédaction de livres sur git

Plus que 70 auteurs !!

HoTT livre sur la théorie homotopique des types.



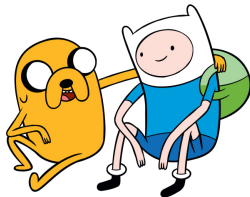
<https://github.com/HoTT/book/pulls>

Autres exemples de revue de code

- ▶ Évaluation par les pairs des articles scientifiques
- ▶ Les articles sur Wikipedia
- ▶ Processus législatif
- ▶ Rédaction de l'Encyclopédie en ligne des suites de nombres entiers (OEIS), par exemple
<https://oeis.org/history?seq=A000142>

Objectifs ?

Aider les gens à écrire du code



- ▶ améliorer la qualité du code
- ▶ favoriser la collaboration, le travail en équipe
- ▶ détecter et corriger les défauts (bogues mais aussi lisibilité) au plus tôt dans le cycle de vie du code pour économiser les coûts
- ▶ former des développeurs, etc

La revue de code peut prendre différentes formes, plus ou moins formelles.

La revue de code peut prendre différentes formes, plus ou moins formelles.

Il faut choisir le modèle le mieux adapté en fonction de la tolérance au risque du code audité.

Notification par email

Le système de gestion de versions peut être configuré pour envoyer un email décrivant chacune des modifications apportées à l'arborescence des sources. Les autres développeurs de l'équipe ont alors l'opportunité d'examiner ces changements.

Inspection formelle

https://en.wikipedia.org/wiki/Fagan_inspection

Michael Fagan, cadre chez IBM, a formalisé en 1976 une méthode d'inspection du code qui se base sur la convocation de plusieurs réunions d'audit.

Assez lourde, elle requiert en moyenne neuf heures-hommes pour 200 lignes de codes, elle est difficile à systématiser pour l'ensemble du code.

Tom Gilb et Dorothy Graham ont développé une autre méthode d'inspection...

Analyse par-dessus l'épaule

L'analyse par-dessus l'épaule (over the shoulder review) est une méthode informelle : l'auteur du code pilote la revue en présentant au relecteur les modifications qu'il a effectuées dans le code source.

Elle a l'avantage d'être simple et rapide à réaliser. L'un de ses inconvénients est que, puisque c'est l'auteur qui pilote la revue, il peut manquer un effet de bord qu'il n'avait pas déjà remarqué au moment de l'écriture.

Programmation en binôme

https://fr.wikipedia.org/wiki/Programmation_en_binôme

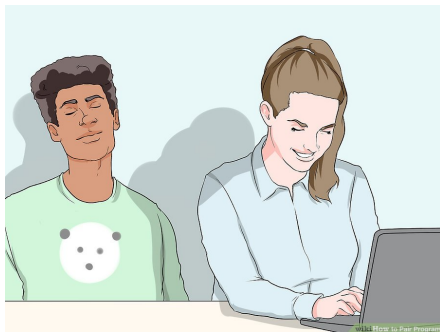


Image de <https://www.wikihow.com/Pair-Program>

Il existe des systèmes d'information permettant d'assister la revue de code.

Objectifs des SI de la revue

- ▶ collecte et présentation des modifications apportées aux fichiers sources qui nécessitent une relecture
- ▶ gestion du workflow : la relecture devient une étape de l'intégration continue
- ▶ annotation des défauts et commentaires issus de la relecture, suivi des corrections de ces défauts
- ▶ statistiques et métriques, etc.

Des outils

- ▶ Gerrit
- ▶ GitLab
- ▶ Emails
- ▶ Github
- ▶ Bitbucket
- ▶ ...

Types de revue de code

Quand faut-il la faire, cette revue ?

- ▶ **avant l'intégration**
- ▶ après l'intégration

Types de revue de code

Quand faut-il la faire, cette revue ?

- ▶ **avant l'intégration**
- ▶ après l'intégration (pas vraiment une revue, plutôt post factum audit)

Types de revue de code

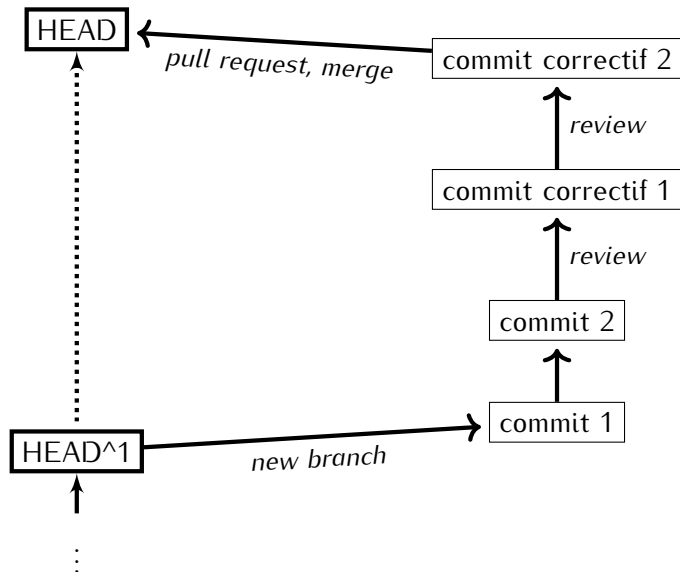
Quand faut-il la faire, cette revue ?

- ▶ **avant l'intégration**
- ▶ après l'intégration (pas vraiment une revue, plutôt post factum audit)

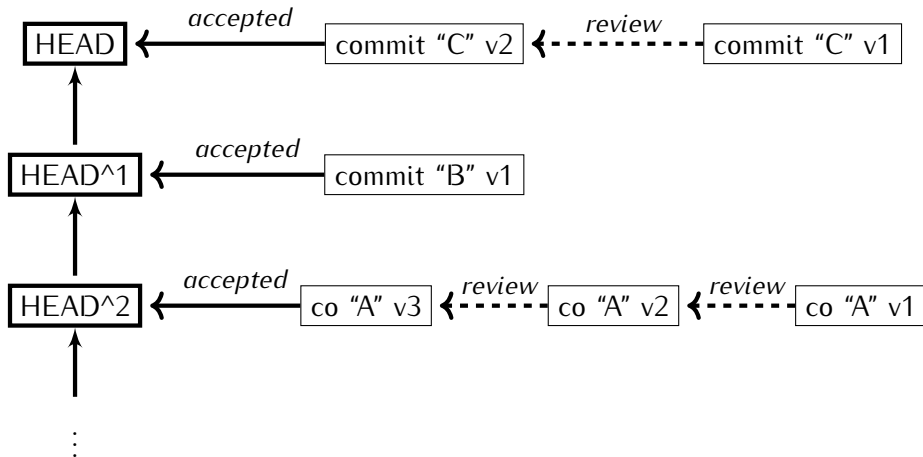
Revue de code et gestion de versions

- ▶ basée sur des patches (par e-mail, par exemple)
- ▶ basée sur les branches (pull-requests)
- ▶ basée sur les commits

Revue basée sur des branches



Revue basée sur des commits



Gerrit comme exemple

- ▶ C'est mieux de faire la revue régulièrement,

- ▶ C'est mieux de faire la revue régulièrement,
- ▶ ... à partir du tout début de projet.

Mes conseils

- ▶ C'est mieux de faire la revue régulièrement,
- ▶ ... à partir du tout début de projet.
- ▶ Petit commits. Plus simple à relire, plus simple à corriger.

Mes conseils

- ▶ C'est mieux de faire la revue régulièrement,
- ▶ ... à partir du tout début de projet.
- ▶ Petit commits. Plus simple à relire, plus simple à corriger.
- ▶ Ne pas prendre trop à cœur les commentaires des collègues.

Mes conseils

- ▶ C'est mieux de faire la revue régulièrement,
- ▶ ... à partir du tout début de projet.
- ▶ Petit commits. Plus simple à relire, plus simple à corriger.
- ▶ Ne pas prendre trop à cœur les commentaires des collègues.
- ▶ **Faire la revue !**

- ▶ C'est mieux de faire la revue régulièrement,
- ▶ ... à partir du tout début de projet.
- ▶ Petit commits. Plus simple à relire, plus simple à corriger.
- ▶ Ne pas prendre trop à cœur les commentaires des collègues.
- ▶ **Faire la revue !**
- ▶ Savoir comment expliquer la (non-)nécessité d'une revue d'un point de vue commercial.

Video !



<https://www.youtube.com/watch?v=lge345hJKJk>

Questions ?