



07

Chapitre

Refonte de code (Refactoring)

1I2AC1 : Génie logiciel et Conception orientée objet

Régis Clouard, ENSICAEN - GREYC

« «Si déboguer c'est supprimer des bugs,
alors programmer ne peut être que de les ajouter. »

Edsger Dijkstra

Qu'est ce que la refonte de code ?

2

■ Définition

- La refonte de code est le processus qui consiste à changer le code d'un système logiciel de telle manière
 - ▶ qu'il **n'altère pas le comportement extérieur** du code
 - ▶ tout en **améliorant sa structure interne**

Pourquoi faire de la refonte de code ?

3

- Pour ne pas accumuler de **dette technique**
- Pour améliorer la logique de conception du logiciel
- Pour rendre le système plus simple à comprendre et donc à maintenir, étendre et vérifier
 - Principe KISS : « Kept It Simple, Stupid. »
 - ▶ « La perfection n'est atteinte, non pas lorsqu'il n'y a plus rien à ajouter, mais lorsqu'il n'y a plus rien à enlever. »
Antoine de Saint-Exupéry
- Pour aider à trouver les bugs (P. ex. les Schroedinbugs)

Quand faire de la refonte de code ?

4

1/ Lors du développement d'une fonctionnalité

- Induit une nouvelle façon de coder :
 - 1) Quick and dirty
 - 2) Refonte du code
 - 3) Ajout de la fonctionnalité dans le code commun
- Remarque : en Agilité l'ajout d'une fonctionnalité dans le code commun ne peut se faire que si son code est propre

Quand faire de la refonte de code ?

5

2/ Lors de la maintenance corrective

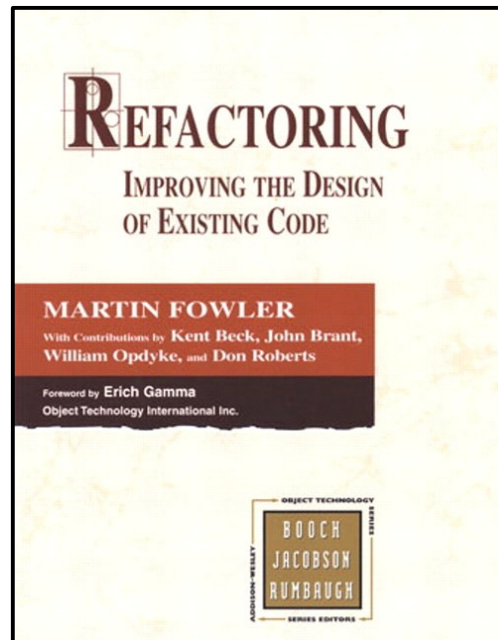
- Lors de la correction d'un bug en profiter pour améliorer le code
- Lors de la relecture du code
 - ▶ Application de la règle de boy-scouts : « *Laissez le campement plus propre que vous l'avez trouvé en arrivant* »

Mise en œuvre

- La refonte nécessite l'utilisation d'environnements de développement évolués (menu « *refactor* » d'*IntelliJ*)

Lectures

- Martin Fowler, Refactoring « *Improving the Design of Existing Code* », Addison-Wesley Professional, 1999.



- Martin Fowler, « *Refactoring Home Page* », <http://refactoring.com/catalog/index.html>

Démo

- Kata Ératosthène
- Algorithme pour trouver les nombres premiers $< n$
 - Ératosthène dessine sur le sable un tableau de taille n
 - Il met une pierre dans chaque case indiquant que c'est un nombre premier
 - Puis en commençant à 2, il supprime tous ses multiples en parcourant les cases de 2 en 2 et en supprimant la pierre
 - Il recommence avec la case suivante qui contient une pierre (ici 3) et il supprime ses multiples (en parcourant les cases de 3 en 3).

Ressources

- Plateforme Moodle
 - Rubrique : Ressources
 - ▶ Kata
 - ▶ Vidéo