

# Soutenance de mémoire

## Master 2 MIAGE

Thibault Sartre

« Comment les flots de contrôle peuvent nous permettre de faire du refactoring de code »

Tuteur Académique : Prof. Emmanuel Hyon



# Sommaire

- Stage

- Présentation de la structure d'accueil
- Contexte et problématiques de la mission
- Résolution de la mission

- Mémoire

- Contexte et problématiques
- Qu'est ce que le refactoring ?
- Outil de refactoring existant (AutoRefactor)
- L'application possible des graphes de flots de contrôle dans le refactoring



# Partie 1 : Stage

# La structure d'accueil

- Structure d'accueil : le syndicat de l'encadrement CFE-CGC Orange
- Objectif : Travailler pour l'intérêt collectif de tous les personnels.
- Equipe d'accueil : Odyssée
- L'équipe Odyssée travaille sur la mise en place et la maintenance du SI du syndicat.



# Contexte et problématiques

- Base de données très riche en données.
- Mais très peu utilisée pour mener des actions ciblées.
- Peu de données sont analysées.
- Mission : Mettre en place un outil permettant de faciliter la mise en relation de n'importe quelles données pour ensuite les faire analyser et mener des actions syndicales adaptées.

# Résolution de la mission

## Méthode de travail :

- Utilisation de la méthode Agile :
  - Sprint d'une semaine
  - Présentation des nouvelles fonctionnalités tous les jeudis
- Utilisation de Trello :
  - Suivi de l'avancement du projet simplement
  - Choix des nouvelles fonctionnalités à développer pour le sprint tous les jeudis (cartes)





# Partie 2 : Mémoire

# Contexte et problématiques

- Augmentation de la puissance des ordinateurs mais ...
- ... diminution de la qualité du code.
- Maintenance et compréhension du code compliquées.
- Comment pourrait-on augmenter la qualité du code facilement?



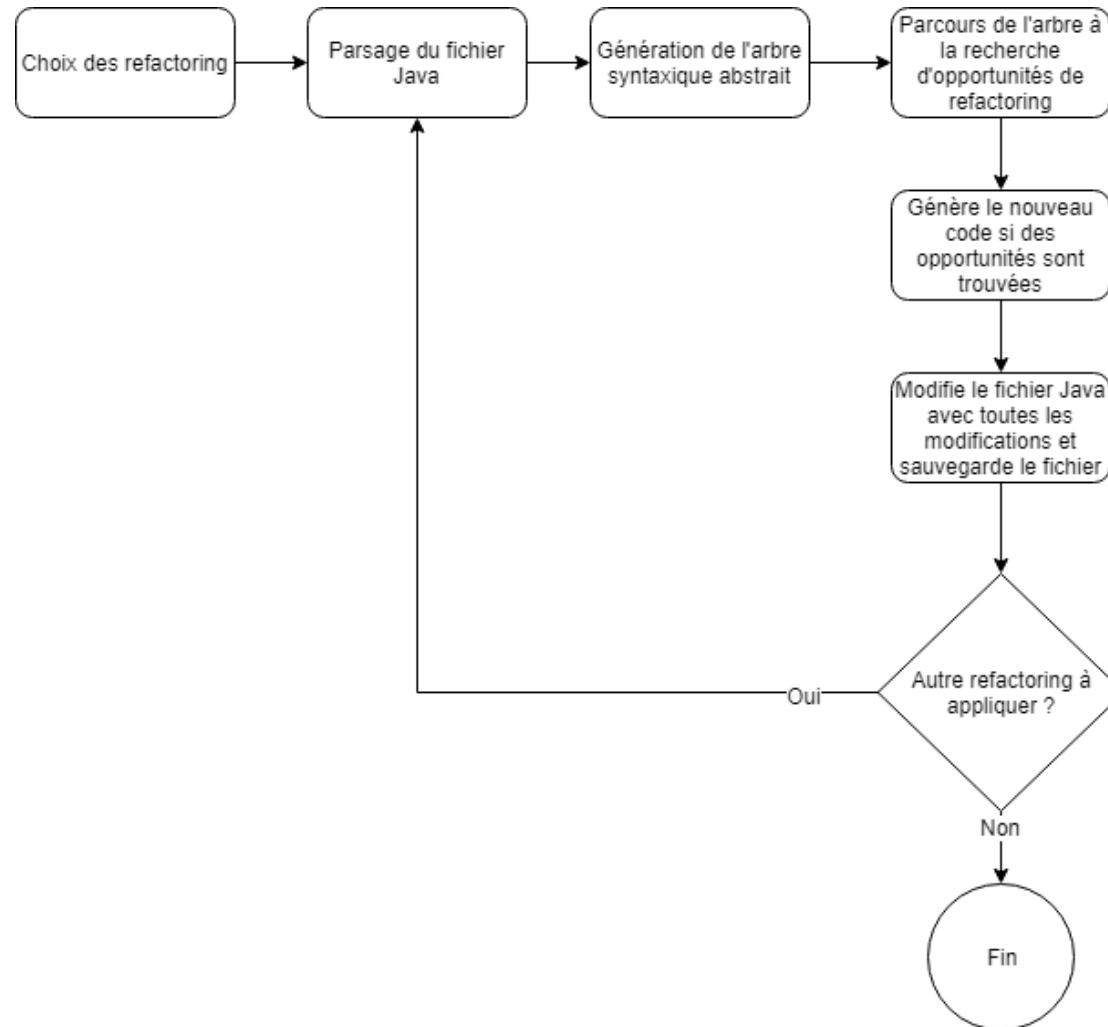
# Qu'est ce que le refactoring ?

- Activité d'ingénierie logicielle qui consiste à modifier le code source d'une application.
- Point positif :
  - Améliore la qualité du code
  - N'altère pas le comportement du code vis-à-vis de l'utilisateur
- Point négatif :
  - Long et difficile à mettre en place

# AutoRefactor

- Projet développé par Jean-Noël Rouvignac [2]
- Outil de refactoring automatique pour le code Java
- Objectifs :
  - Faciliter la maintenance du code
  - Moderniser le code
  - Augmenter les performances des programmes

# AutoRefactor fonctionnement

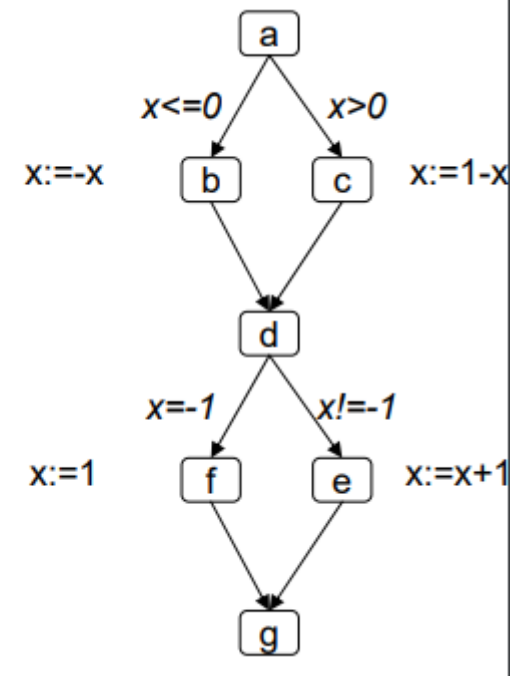
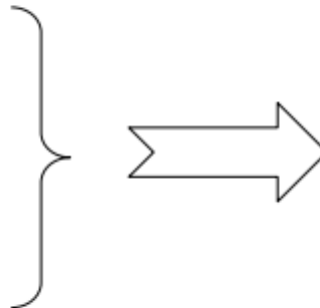


*Schéma représentant l'exécution d'AutoRefactor*

# Les graphes de flots de contrôle

- Représentation de tous les chemins possibles d'un programme durant son exécution.

```
begin  
  if (x<=0) then x:=-x  
  else x:=1-x;  
  if (x=-1) then x:=1  
  else x:=x+1;  
end
```



*Exemple de graphe de flots de contrôle [1]*

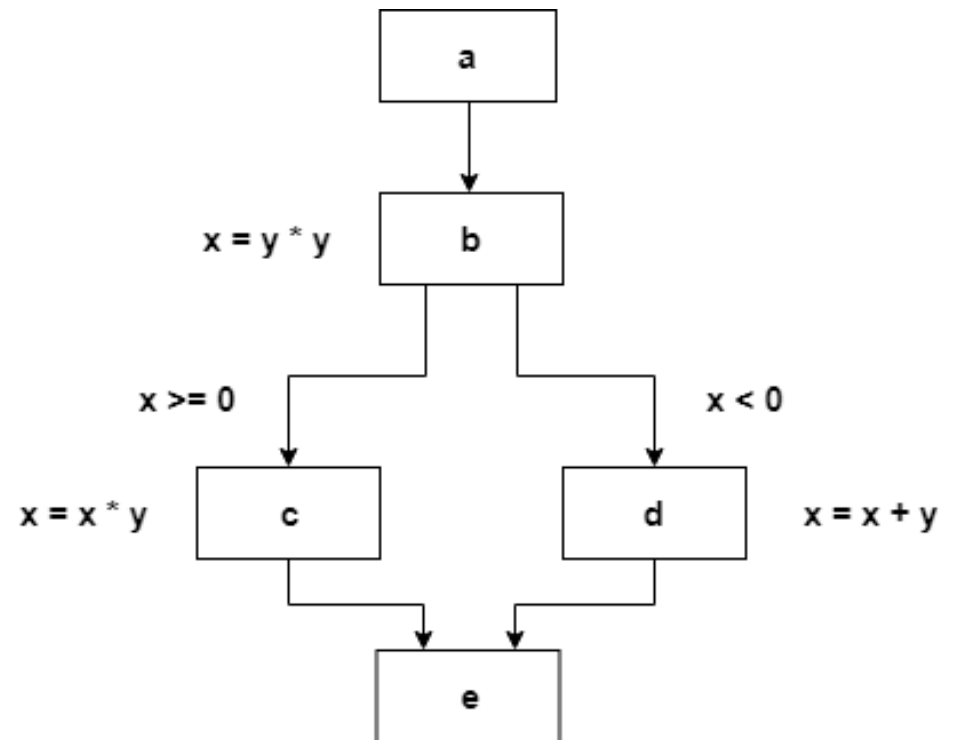
# Sont-ils utilisables dans le refactoring ?

- Actuellement, aucun logiciel ne les utilise.
- Utilisation du test structurel ?
  - Couverture de toutes les décisions possibles
- Détection de code mort
- Suppression de code mort = Refactoring

# Exemple de refactoring

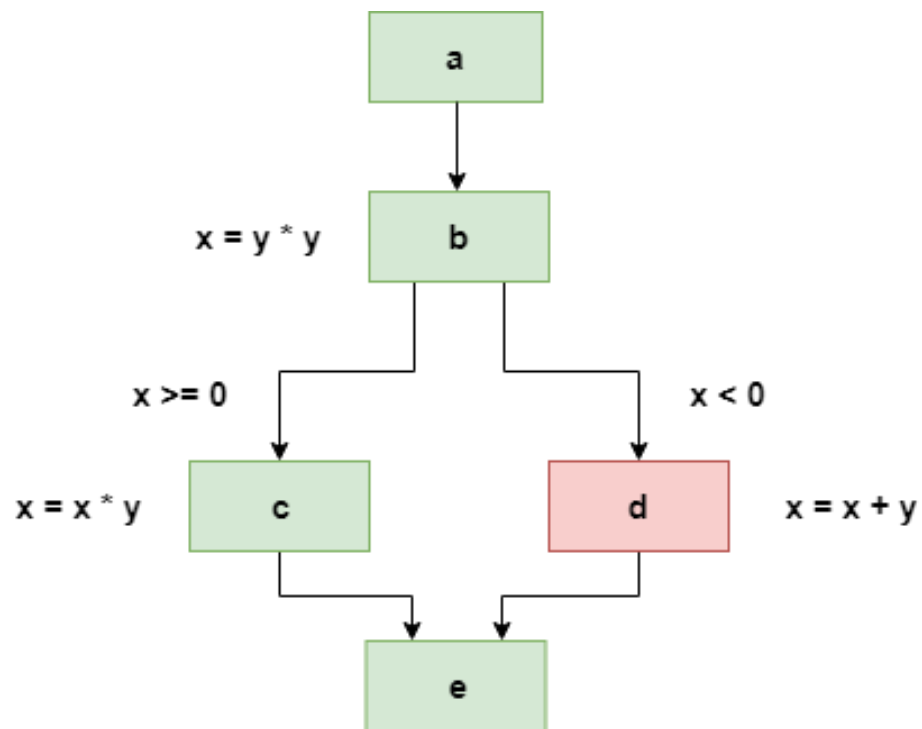
```
1  begin
2  x = y * y;
3  if(x >= 0){
4      x = x * y;
5  }else{
6      x = x + y;
7  }
8  end
```

*Exemple de programme à analyser*



Graph representing the program

# Exemple de refactoring



Graphe après analyse

```
1  begin
2  x = y * y;
3  if(x >= 0){
4      x = x * y;
5  }
6  //else{
7      //x = x + y;
8  //}
9  end
```

Code après le refactoring

# Algorithme proposé

- 1) Générer un graphe de flots de contrôle à partir du code.
- 2) Faire une couverture de toutes les décisions possibles.
- 3) Colorier en vert les blocs atteints et en rouge les blocs inatteignables.
- 4) Mettre en commentaire les parties du code qui sont inatteignables.



# Bilan mémoire

- La rédaction de ce mémoire m'a permis d'améliorer mes connaissances concernant le refactoring.
- J'ai appris énormément sur des sujets très intéressants comme le test structurel et l'utilisation des graphes de flots de contrôle.
- Et j'ai été surpris de découvrir qu'une personne était en train de développer un outil pour faire du refactoring avec les graphes de flot de contrôle.

# Conclusion

- Le code peut rapidement devenir illisible et difficilement maintenable.
- Mais il existe de nombreuses techniques pour faire du refactoring.
- Il pourrait être intéressant dans le futur de s'intéresser aux graphes de flots de contrôle pour détecter du code mort.

# Webographie

- [1] : Patrick Felix « [http ://dept-info.labri.fr/ felix/](http://dept-info.labri.fr/felix/) »
- [2]: Jean-Noël Rouvignac  
« <https://github.com/JnRouvignac/AutoRefactor> »