Projet de validation et maintenance des applications

(420-EAG-LI)

**Travail Pratique 2**

**Métriques, stratégie de tests, tests unitaires, jeux d’essais et Ant**



**Travail présenté à :**

Hichem Klabi

**Par :**

Pierre Montminy

Dyden Ung

Adrian Pinzaru

Edivalter DosSantos

10 mai 2016

Table des matières

[Introduction 2](#_Toc449446024)

[1. Calcul des métriques orientés objet (1er calcul) 3](#_Toc449446025)

[2. Programmation des classes 4](#_Toc449446026)

[3. Calcul des métriques orientées objet (2ième calcul) 5](#_Toc449446027)

[4. Spécifications et écriture des contrats des classes 6](#_Toc449446028)

[5. Jeux d’essais et tests unitaires 7](#_Toc449446029)

[6. Utilisation de l’outil Ant 8](#_Toc449446030)

# Introduction

# 1. Calcul des métriques orientés objet (1er calcul)

## 1.1 Calcul des métriques orientées objet provenant du diagramme de classes du TP1

CBO

Plus le chiffre est petit, donc il y a moins de dépendances avec les autres classes, donc c’est bon

Aggrégation entre les 3 classes (employé, service, sous-service)

NOC

En regardant le nombre total de classes, une classe qui a 3 enfants nous paraît adéquat compte tenue de la situation (énoncé).

5 classes utilisent l’entité employé.

Les classes administrateurs, superviseurs, responsables chaudière sont des employés, alors l’héritage est adéquat dans ce contexte.

Ajouter l’image (métrique calculée par Eclipse) classe employée

DIT

3 classes ont un seul parent, donc l’héritage n’est profond, ce qui fait que la conception n’est pas complexe.

Ajouter l’image (métriques des 3 classes)

Moyennes :

En résumé, c’est bon!!!

Améliorations (1.3)

La relation entre la classe sous-service et la classe service devient une composition (au lieu d’une aggrégation)

# 2. Programmation des classes

# 3. Calcul des métriques orientées objet (2ième calcul)

# 4. Spécifications et écriture des contrats des classes

# 5. Jeux d’essais et tests unitaires

# 6. Utilisation de l’outil Ant