

UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À
L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE EN MATHÉMATIQUES ET INFORMATIQUE
APPLIQUÉES

PAR
AYMEN KOUT

ÉVALUATION DE L'IMPACT DE L'INTRODUCTION DES ASPECTS SUR
LA TESTABILITÉ DES PROGRAMMES

DÉCEMBRE 2011

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire ou de cette thèse a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire ou de sa thèse.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire ou cette thèse. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire ou de cette thèse requiert son autorisation.

Évaluation de l'impact de l'introduction des aspects sur la testabilité des programmes

Aymen KOUT

SOMMAIRE

La qualité du logiciel a connu une importante amélioration durant les dernières décennies. L'adoption de nouvelles techniques et technologies et la prise de conscience de l'importance de l'assurance qualité a séduit les chercheurs du génie logiciel partout dans le monde. La testabilité logicielle tient une place proéminente dans le cadre des caractéristiques de la maintenabilité. Elle est considérée comme un facteur très important dans le modèle de qualité du standard ISO 9126.

La programmation orientée aspect apporte de nouveaux concepts permettant de repousser les limites de la programmation orientée objet. Elle offre, en fait, de nouvelles perspectives permettant une meilleure séparation des préoccupations transverses dans un programme. Le code correspondant à ces préoccupations est factorisé en des unités modulaires appelées aspects. Ceci devrait avoir des conséquences positives sur certains facteurs de la qualité des programmes. Les aspects introduisent, cependant, de nouvelles dimensions en termes de dépendances et de contrôle. Ils constituent aussi une nouvelle source de fautes.

L'objectif principal, dans ce mémoire, porte sur une évaluation spécifique de l'impact de l'introduction des aspects sur la testabilité des programmes. Cette évaluation est basée, entre autres, sur un ensemble de métriques spécifiques au code et d'autres spécifiques à l'effort de test (complexité des tests, effort de test, etc.). Il s'agit de faire une évaluation comparative sur le plan de la testabilité entre des implémentations Java et celles qui leur sont équivalentes en AspectJ.

Assessing the impact of introducing aspects on programs testability

Aymen KOUT

ABSTRACT

The concept of software quality has evolved in the last decades. The adoption of new techniques and technologies and the awareness of the importance of quality assurance techniques seduced the software engineering researchers all around the world. Software testability holds a prominent place in the maintainability characteristics. It is considered as a very important factor in the quality model of the ISO 9126 standard.

The aspect-oriented paradigm brings new concepts allowing to push away the limits of the object-oriented paradigm. In fact, it offers new perspectives allowing a better separation of cross-cutting concerns in a program. The code corresponding to these concerns is factorized in modular units called aspects. This should have positive consequences on many factors of software quality. However, aspects introduce new dimensions in terms of dependences and control. They also constitute a new source of faults.

The main objective, in this work, concerns a specific evaluation of the impact of the introduction of aspects on programs testability. This evaluation is based, among others, on a suite of specific code source level metrics and a suite of specific test case metrics (complexity, test effort, etc.). This work aims at making a comparative evaluation, from the point of view of software testability, between Java implementations and those who are equivalent in AspectJ.

REMERCIEMENTS

Merci,

À mon père, pour son soutien moral et matériel constant, son encouragement m'a été d'une aide précieuse.

À mes professeurs et directeurs de recherche Mourad BADRI et Linda BADRI, pour leur soutien intellectuel et financier, leur disponibilité et leur encadrement sans faille. Merci pour les plaisants moments passés au laboratoire. Merci également pour votre patience. Votre aide et votre expertise dans le domaine du génie logiciel m'ont beaucoup aidé pour mener à terme ce mémoire.

À tous les professeurs du département.

À mes frères, mes sœurs et mes proches parents qui m'ont assisté et encouragé durant toutes mes études malgré la distance qui nous sépare.

À mes amis pour la disponibilité dont ils ont fait preuve, le réconfort qu'ils m'ont apporté lors des moments difficiles, merci d'avoir été là pour moi lorsque le moral était au plus bas.

Cette étude a été rendue possible grâce à la contribution financière du CRSNG (Conseil de Recherches en Sciences Naturelles et en Génie du Canada) et à la fondation de l'UQTR.

TABLE DE MATIERES

CHAPITRE 1	10
INTRODUCTION	10
1.1 PROBLEMATIQUE.....	11
1.2 APPROCHE	12
1.3 ORGANISATION.....	14
CHAPITRE 2.....	16
ÉTAT DE L'ART : UN BREF APERÇU SUR LES TRAVAUX CONCERNANT LA TESTABILITÉ LOGICIELLE.....	16
2.1 L'IMPACT DE L'ASPECTUALISATION DES CLASSES OO	17
2.2 L'EVALUATION DE LA TESTABILITÉ DES PROGRAMMES.....	20
2.3 TEST DES PROGRAMMES OA.....	24
2.3.1 <i>Génération automatique des classes de test</i>	24
2.3.2 <i>Autres techniques de test des programmes OA</i>	26
CHAPITRE 3	28
MÉTRIQUES DE TESTABILITÉ	28
3.1 METRIQUES ORIENTEES OBJET.....	28
3.2 METRIQUES DE CAS TEST.....	32
CHAPITRE 4	36
ÉVALUATIONS EMPIRIQUES.....	36
4.1 UNE ANALYSE EMPIRIQUE D'UN MODELE DE TESTABILITÉ POUR LES PROGRAMMES ORIENTES OBJET.....	37
4.1.1 <i>Adaptation du modèle</i>	37
4.1.2 <i>Cadre expérimental.....</i>	39
4.1.3 <i>Étude de Cas.....</i>	42
4.1.4 <i>Conclusion</i>	47
4.2 UNE ANALYSE EMPIRIQUE DE L'IMPACT DE L'INTRODUCTION DES ASPECTS SUR LA TESTABILITÉ DES CLASSES OO	48
4.2.1 <i>Cadre expérimental.....</i>	48
4.2.2 <i>Étude de Cas.....</i>	50
4.2.3 <i>Conclusion</i>	97

4.3 UNE ANALYSE EMPIRIQUE DE LA TESTABILITE DES PROGRAMMES OO : TESTS GENERES AUTOMATIQUEMENT VS TESTS DEVELOPPE.....	98
4.3.1 <i>Cadre expérimental</i>	98
4.3.2 <i>Étude de Cas</i>	100
4.3.3 <i>Conclusion</i>	108
CHAPITRE 5.....	109
CONCLUSION	109
ANNEXES	110
REFERENCES BIBLIOGRAPHIQUES	131

LISTE DES TABLEAUX

Tableau 1 : Quelques caractéristiques des systèmes utilisés	42
Tableau 2 : Valeurs de corrélation entre le MTMOOP et les métriques de cas de test	43
Tableau 3 : Valeurs de corrélation entre les métriques de cas de test	44
Tableau 4 : Caractéristiques des systèmes Java utilisés	52
Tableau 5 : Quelques caractéristiques des systèmes AspectJ utilisés	52
Tableau 6.1 : Matrice de corrélation (Pearson) de Healthwatcher Java	56
Tableau 6.2 : Matrice de corrélation (Spearman) de Healthwatcher Java	57
Tableau 6.3 : Matrice de corrélation (Pearson) de Healthwatcher AspectJ	58
Tableau 6.4 : Matrice de corrélation (Spearman) de Healthwatcher AspectJ	59
Tableau 6.5 : Résultats du test statistique de Healthwatcher	63
Tableau 7.1 : Matrice de corrélation (Pearson) de Telestrada Java	64
Tableau 7.2 : Matrice de corrélation (Spearman) de Telestrada Java	65
Tableau 7.3 : Matrice de corrélation (Pearson) de Telestrada AspectJ	66
Tableau 7.4 : Matrice de corrélation (Spearman) de Telestrada AspectJ	67
Tableau 7.5 : Résultats du test statistique de Telestrada	71
Tableau 8.1 : Matrice de corrélation (Pearson) de Petstore Java	72
Tableau 8.2 : Matrice de corrélation (Spearman) de Petstore Java	73
Tableau 8.3 : Matrice de corrélation (Pearson) de Petstore AspectJ	74
Tableau 8.4 : Matrice de corrélation (Spearman) de Petstore AspectJ	75
Tableau 8.5 : Résultats du test statistique de PestStore	79
Tableau 9.1 : Matrice de corrélation (Pearson) de HSQLBD Java	80
Tableau 9.2 : Matrice de corrélation (Spearman) de HSQLBD Java	81
Tableau 9.3 : Matrice de corrélation (Pearson) de HSQLBD AspectJ	82
Tableau 9.4 : Matrice de corrélation (Spearman) de HSQLBD AspectJ	83
Tableau 9.5 : Résultats du test statistique de HSQLBD	87
Tableau 10.1 : Matrice de corrélation (Pearson) de JHotdraw Java	88
Tableau 10.2 : Matrice de corrélation (Spearman) de JHotdraw Java	89
Tableau 10.3 : Matrice de corrélation (Pearson) de AJHotdraw AspectJ	90
Tableau 10.4 : Matrice de corrélation (Spearman) de AJHotdraw AspectJ	91
Tableau 10.5 : Résultats du test statistique de (J, AJ) Hotdraw	95
Tableau 11 : Quelques caractéristiques des systèmes et des suites de tests utilisés	101

Tableau 12 : Valeurs de corrélation entre le MTMOOP et les métriques sélectionnées pour le système Ant	102
Tableau 13 : Valeurs de corrélation entre le MTMOOP et les métriques sélectionnées pour le système DBUnit	103
Tableau 14 : Valeurs de corrélation entre le MTMOOP et les métriques sélectionnées pour le système JFreeChart	104
Tableau 15 : Valeurs de corrélation entre le MTMOOP et les métriques sélectionnées pour le système POI	105
Tableau 16.1 : Matrice de corrélation (Pearson) de Healthwatcher Java (Toutes les classes)	111
Tableau 16.2 : Matrice de corrélation (Spearman) de Healthwatcher Java (Toutes les classes)	112
Tableau 16.3 : Matrice de corrélation (Pearson) de Healthwatcher AspectJ (Toutes les classes)	113
Tableau 16.4 : Matrice de corrélation (Spearman) de Healthwatcher AspectJ (Toutes les classes)	114
Tableau 17.1 : Matrice de corrélation (Pearson) de Telestrada Java (Toutes les classes)	115
Tableau 17.2 : Matrice de corrélation (Spearman) de Telestrada Java (Toutes les classes)	116
Tableau 17.3 : Matrice de corrélation (Pearson) de Telestrada AspectJ (Toutes les classes)	117
Tableau 17.4 : Matrice de corrélation (Spearman) de Telestrada AspectJ (Toutes les classes)	118
Tableau 18.1 : Matrice de corrélation (Pearson) de Petstore Java (Toutes les classes)	119
Tableau 18.2 : Matrice de corrélation (Spearman) de Petstore Java (Toutes les classes)	120
Tableau 18.3 : Matrice de corrélation (Pearson) de Petstore AspectJ (Toutes les classes)	121
Tableau 18.4 : Matrice de corrélation (Spearman) de Petstore AspectJ (Toutes les classes)	122
Tableau 19.1 : Matrice de corrélation (Pearson) de HSQLBD Java (Toutes les classes)	123
Tableau 19.2 : Matrice de corrélation (Spearman) de HSQLBD Java (Toutes les classes)	124
Tableau 19.3 : Matrice de corrélation (Pearson) de HSQLBD AspectJ (Toutes les classes)	125
Tableau 19.4 : Matrice de corrélation (Spearman) de HSQLBD AspectJ (Toutes les classes)	126
Tableau 20.1 : Matrice de corrélation (Pearson) de JHotdraw Java (Toutes les classes)	127
Tableau 20.2 : Matrice de corrélation (Spearman) de JHotdraw Java (Toutes les classes)	128
Tableau 20.3 : Matrice de corrélation (Pearson) de AJHotdraw AspectJ (Toutes les classes)	129
Tableau 20.4 : Matrice de corrélation (Spearman) de AJHotdraw AspectJ (Toutes les classes)	130

LISTE DES FIGURES

Figure 1.1 MTMOOP : HealthWatcher (OO vs OA)	60
Figure 1.2 TNofCalled : HealthWatcher (OO vs OA)	61
Figure 1.3 TNofNewData : HealthWatcher (OO vs OA)	61
Figure 1.4 TNofAssert : HealthWatcher (OO vs OA)	61
Figure 1.5 TNOO : HealthWatcher (OO vs OA)	61
Figure 1.6 TLOC : HealthWatcher (OO vs OA)	61
Figure 1.7 TRFC : HealthWatcher (OO vs OA)	61
Figure 1.8 TWMPC1 : HealthWatcher (OO vs OA)	61
Figure 1.9 TWMPC2 : HealthWatcher (OO vs OA)	61
Figure 2.1 MTMOOP : Telestrada (OO vs OA)	68
Figure 2.2 TNofCalled : Telestrada (OO vs OA)	69
Figure 2.3 TNofNewData : Telestrada (OO vs OA)	69
Figure 2.4 TNofAssert : Telestrada (OO vs OA)	69
Figure 2.5 TNOO : Telestrada (OO vs OA)	69
Figure 2.6 TLOC : Telestrada (OO vs OA)	69
Figure 2.7 TRFC : Telestrada (OO vs OA)	69
Figure 2.8 TWMPC1: Telestrada (OO vs OA)	69
Figure 2.9 TWMPC2 : Telestrada (OO vs OA)	69
Figure 3.1 MTMOOP : Petstore (OO vs OA)	76
Figure 3.2 TNofCalled : Petstore (OO vs OA)	76
Figure 3.3 TNofNewData : Petstore (OO vs OA)	76
Figure 3.4 TNofAssert: Petstore (OO vs OA)	77
Figure 3.5 TNOO : Petstore (OO vs OA)	77
Figure 3.6 TLOC : Petstore (OO vs OA)	77
Figure 3.7 TRFC : Petstore (OO vs OA)	77
Figure 3.8 TINVOK : Petstore (OO vs OA)	77
Figure 3.9 TWMPC : Petstore (OO vs OA)	77
Figure 4.1 MTMOOP : HSQLBD OO vs HSQLBD OA	84

Figure 4.2 TNofCalled : HSQLBD OO vs HSQLBD OA	85
Figure 4.3 TNofNewData : HSQLBD OO vs HSQLBD OA	85
Figure 4.4 TNofAssert : HSQLBD OO vs HSQLBD OA	85
Figure 4.5 TNOO : HSQLBD OO vs HSQLBD OA	85
Figure 4.6 TLOC : HSQLBD OO vs HSQLBD OA	85
Figure 4.7 TRFC : HSQLBD OO vs HSQLBD OA	85
Figure 4.8 TINVOK : HSQLBD OO vs HSQLBD OA	85
Figure 4.9 TWMPC : HSQLBD OO vs HSQLBD OA	85
Figure 5.1 MTMOOP : JHotDraw vs AJHotDraw	92
Figure 5.2 TNofCalled : JHotDraw vs AJHotDraw	92
Figure 5.3 TNofNewData: JHotDraw vs AJHotDraw	92
Figure 5.4 TNofAssert : JHotDraw vs AJHotDraw	93
Figure 5.5 TNOO : JHotDraw vs AJHotDraw	93
Figure 5.6 TLOC: JHotDraw vs AJHotDraw	93
Figure 5.7 TRFC : JHotDraw vs AJHotDraw	93
Figure 5.8 TINVOK : JHotDraw vs AJHotDraw	93
Figure 5.9 TWMPC : JHotDraw vs AJHotDraw	93

CHAPITRE 1

INTRODUCTION

Les systèmes industriels actuels sont de plus en plus complexes. La gestion (assurance et contrôle) de leur qualité répond à des enjeux techniques et économiques importants. La qualité des logiciels est un concept complexe. Elle possède plusieurs caractéristiques.

Comme toutes les caractéristiques de la qualité, la testabilité d'un logiciel est affectée par différents facteurs, comme la validité, le processus et les outils utilisés, la représentation des besoins, etc. [Bruntink04]. Dans ce mémoire, nous allons aborder ce qui a été écrit sur ce sujet. Nous essayerons également de répondre à quelques questions importantes pour mieux comprendre cet attribut qualité. Par exemple, comment définir la testabilité? Qu'est-ce qui rend un système plus testable qu'un autre? Ou bien encore, comment déterminer (prédir) la testabilité d'un logiciel?

La programmation orientée aspect apporte de nouveaux concepts permettant de repousser les limites de la programmation orientée objet. Elle offre, en fait, de nouvelles perspectives permettant une meilleure séparation des préoccupations transverses dans un programme. Le code correspondant à ces préoccupations est factorisé dans des unités modulaires appelées aspects. Ceci devrait avoir des conséquences positives sur certains facteurs de la qualité des programmes. Les aspects introduisent, cependant, de nouvelles dimensions en termes de dépendances et de contrôle. Ils constituent aussi une nouvelle source de fautes.

La testabilité logicielle tient une place proéminente dans le cadre des caractéristiques de la maintenabilité. Elle est considérée comme un facteur très important dans le modèle de qualité du standard ISO 9126 [Bruntink04]. Dans ce mémoire, nous allons nous intéresser à la testabilité logicielle dans le contexte *orienté objet* et *orienté aspect* en particulier.

1.1 Problématique

La programmation orientée aspect, comme mentionné précédemment, apporte de nouveaux concepts permettant de repousser les limites de la programmation orientée objet. Elle permet une meilleure séparation des préoccupations transverses dans un programme. Le code correspondant à ces préoccupations est factorisé en des unités modulaires appelées *aspects*. Ceci permet d'améliorer la modularité des systèmes et avoir des conséquences positives sur certains facteurs de leur qualité.

L'objectif principal, dans ce projet, porte sur une évaluation spécifique de l'impact de l'introduction des aspects sur la testabilité des programmes. Cette évaluation est basée, entre autres, sur un ensemble de métriques spécifiques au code (quelques aspects relatifs à la qualité des programmes – métriques orientées objet et aspect) et d'autres spécifiques à l'effort de test (taille des tests, complexité des tests, etc.).

Il s'agit, en particulier, de faire une évaluation comparative sur le plan de la testabilité des classes entre le code orienté objet (Java) de plusieurs applications ayant subies un refactoring aspect et leur code orienté aspect (AspectJ) après refactoring. Le but étant d'évaluer empiriquement l'impact de l'introduction des aspects sur la testabilité des classes du point de vue de l'effort du test unitaire. L'évaluation se fera aux deux principaux niveaux : code (quelques caractéristiques en lien avec la modularité et la testabilité en particulier) et l'effort de test (complexité des scénarios de test, taille, etc.).

1.2 Approche

Ce mémoire s'inscrit dans ce cadre et présente essentiellement les résultats d'une étude empirique (diverses expérimentations réalisées) sur plusieurs projets d'envergure. Pour chaque projet, nous avons considéré les deux versions de code : code orienté objet Java avant refactoring aspect et code AspectJ après refactoring. Cette étude consiste, entre autres, à comparer les deux implémentations à plusieurs niveaux (divers attributs). Nous avons, dans ce contexte, utilisé (après adaptation) le modèle développé par R. A. Khan et al. [Khan09] pour estimer la testabilité des classes des projets retenus. Pour toutes les études de cas considérées, nous avons effectué la génération complète des séquences de test JUnit [JUnit] aussi bien pour les versions Java que pour les versions AspectJ équivalentes. Enfin, nous avons fait une évaluation empirique comparative entre les codes des deux implémentations (Java et AspectJ), d'une part, et les scénarios de test complets relatifs aux deux implémentations, d'autre part.

Nous avons considéré, pour simplifier la problématique et comme première tentative dans ce contexte, la testabilité d'un point de vue « tests unitaires » des classes uniquement. Nous avons utilisé l'outil CodePro [CodePro] pour générer les tests unitaires pour l'ensemble des classes, pour les deux versions : Java (avant refactoring) et AspectJ (après refactoring). Cette étape consistait à développer les cas de test correspondant aux codes de base sans tenir compte des aspects. Puis, nous avons évalué comparativement, selon les multiples critères que nous avons retenus, l'ensemble des scénarios avant et après aspectualisation. Plusieurs méthodes statistiques ont été utilisées pour analyser les données collectées et interpréter les résultats obtenus.

Nous avons, par ailleurs, étendu notre étude en considérant deux scénarios : le premier dans lequel les séquences de test correspondant aux études de cas retenues ont été développées par des développeurs et le second scénario dans lequel les séquences de test ont été générées automatiquement en utilisant un outil. Dans cette partie, nous présenterons les résultats de l'étude empirique comparative réalisée sur des programmes

orientés objet dont les classes tests ont été développées par des développeurs, versus ces mêmes programmes, mais avec des classes tests générées automatiquement.

L'étude réalisée avait essentiellement pour objectifs de vérifier ou rejeter principalement les hypothèses suivantes :

- Le Modèle MTMOOP (MTMOOD [Khan09] adapté) permet de prédire la testabilité des classes OO.
- Pour les classes qui ont subi un refactoring aspect, leur testabilité (sans les préoccupations transverses) s'est améliorée (comparée à celle des classes Java avant le refactoring aspect). Il s'agit ici, d'évaluer empiriquement l'impact de l'introduction des aspects dans le code objet sur la testabilité des classes.
- Les suites de test générées automatiquement sont plus complètes que celles générées par des programmeurs.

Ces différentes hypothèses sont reliées à des enjeux importants et d'actualité dans le domaine du développement, de la maintenance et de l'évolution des logiciels de grande taille. Les nombreuses expérimentations effectuées ont été menées dans le cadre d'un environnement complet (composée de plusieurs outils) d'expérimentation que nous avons construit. Plusieurs expérimentations d'envergure ont été conduites sur des systèmes orientés objet (Java) et orientés aspect (AspectJ) de grande taille. Les résultats obtenus sont très positifs à plusieurs niveaux. Ils permettent également d'envisager plusieurs pistes de recherche dans le cadre de travaux futurs. Une publication importante [Kout11] portant sur une partie de ce travail a été effectuée. Une autre est prévue.

1.3 Organisation

Le présent mémoire est divisé en trois grandes parties. Dans un premier temps, et comme le domaine de la testabilité logicielle est très vaste et a fait l'objet de plusieurs types de travaux, nous ferons un bref survol des principaux travaux effectués dans ce domaine. Nous organiserons cette première partie comme suit : nous commencerons, dans un premier temps, par donner un aperçu sur quelques travaux importants qui ont porté sur l'évaluation de l'impact de l'introduction des aspects dans le code objet (aspectualisation des classes OO (orientées objet)). Nous ferons par la suite un survol des principaux travaux qui ont été effectués dans le domaine de la testabilité logicielle. Nous terminerons cette partie par une brève présentation des tests dans le contexte des programmes orientés aspect.

Dans un deuxième temps, nous présenterons les métriques de testabilité que nous avons sélectionnées pour la réalisation de notre étude. Nous ferons, dans cette partie, le tour des travaux qui ont porté sur ces métriques.

Enfin, dans un troisième temps, nous présenterons l'étude empirique que nous avons effectuée. Cette partie sera organisée autour des trois principaux groupes d'expérimentations effectuées, à savoir :

- Une analyse empirique d'un modèle de testabilité des programmes orientés objet. Nous présenterons le modèle original MTMOOD et son adaptation MTMOOP.
- Une analyse empirique de l'impact de l'introduction des aspects sur la testabilité des classes OO.
- Une évaluation comparative des résultats selon deux scénarios : (1) les tests sont développés par des développeurs, et (2) les tests sont générés automatiquement.

Nous terminerons ce mémoire par une conclusion globale dans laquelle, entre autres, nous poserons des questions de recherche restées ouvertes qu'il serait intéressant d'examiner dans le cadre de travaux futurs.

CHAPITRE 2

ÉTAT DE L'ART : UN BREF APERÇU SUR LES TRAVAUX CONCERNANT LA TESTABILITÉ LOGICIELLE

Dans ce chapitre, nous dressons un bref état de l'art de certains éléments relatifs à la testabilité logicielle, en général, et la testabilité dans le contexte des applications orientées objet et orientées aspect, en particulier.

La testabilité logicielle tient une place proéminente dans le cadre des caractéristiques de la maintenabilité. Elle est considérée comme un facteur très important dans le modèle de qualité du standard ISO 9126 [Bruntink04]. Comme toutes les caractéristiques de la qualité, la testabilité d'un logiciel est affectée par différents facteurs comme [Bruntink04] : la validité, le processus de développement et les outils utilisés, la représentation des besoins, etc. Par ailleurs, la migration vers de nouveaux paradigmes de programmation comme la programmation orientée aspect (POA) peut également affecter la testabilité d'un logiciel. Jusqu'à présent, il n'y a pas vraiment d'études poussées qui évaluent (ou prédisent) exactement la testabilité d'un logiciel OA (orienté aspect) comme on peut en trouver pour les programmes OO (orientés objet).

Dans cette partie, nous allons présenter sommairement ce qui a été écrit sur ce sujet jusqu'à présent. Nous essayerons de répondre à quelques questions importantes pour mieux comprendre cet attribut qualité; par exemple, quel est l'impact de l'aspectualisation des classes OO sur la testabilité? Qu'est-ce qui rend un système (ou un composant d'un système, comme une classe) plus testable qu'un autre? Ou bien encore, comment évaluer la testabilité d'un logiciel (ou d'une classe par exemple)?

2.1 L'impact de l'aspectualisation des classes OO

Ceccato et al. [Ceccato04] ont mené une étude quantitative pour mesurer l'impact de l'aspectualisation d'un programme. Dans leur article [Ceccato04], les auteurs se sont intéressés surtout à ce qu'ils appellent couplage implicite introduit par les aspects, pour essayer de déterminer les avantages et désavantages de la POA. Ils ont proposé une approche basée sur quelques métriques (taille (1), complexité (2), héritage (2), cohésion (1) et couplage (4)) qu'ils ont calculées sur de petits programmes OO et leurs équivalents AspectJ [AspectJ]. Les auteurs ont conclu qu'il y a eu une amélioration générale des valeurs de certaines métriques (donc des attributs relatifs à ces métriques). Le coût à payer de ces améliorations est, cependant, une augmentation des valeurs de la métrique CIM (Coupling on Intercepted Modules) et de la métrique CDA (Crosscutting Degree of an Aspect) en raison de l'interception de l'aspect à l'exécution des méthodes (couplage POA). Les résultats indiquent que les propriétés telles que la proportion du système touchée par un aspect et le degré de connaissance d'un aspect des modules dans lesquels il s'injecte sont capturées par les métriques de couplages (CDA et CIM). Ceccato et al. [Ceccato04] envisagent, alors, la définition d'un ensemble d'indicateurs POA communs, pour qu'il soit adopté par la communauté POA, afin de simplifier la comparaison des résultats obtenus par différentes équipes de recherche et d'avoir une méthode d'évaluation standard.

Plusieurs modèles de patrons de conception présentent des préoccupations transverses. Dans ce contexte, Hannemann et Kiczales (HK) [Hannemann02] ont entrepris une étude dans laquelle ils ont développé et ont comparé des implémentations Java et AspectJ des 23 patrons de conception GoF [GoF]. Ils revendiquent que le langage de programmation utilisé affecte la mise en œuvre des patrons de conception. De là, il est naturel d'explorer l'effet des techniques de la programmation orientée aspect sur l'implémentation des modèles GoF. Pour chacun des 23 patrons GoF, ils ont développé un exemple représentatif qui se sert du patron et qui implémente un exemple Java et un autre AspectJ. Deux sortes de rôles de patrons sont identifiés dans l'étude de HK : ceux qui ont un rôle

de définition et ceux qui ont un rôle de surimpression. Un rôle de définition définit une classe complètement participante. Autrement dit, les classes qui jouent un rôle de définition n'ont aucune fonctionnalité à l'extérieur du patron. Un rôle de surimpression peut être assigné aux classes participantes qui ont des fonctionnalités à l'extérieur du patron. Ces sortes de rôles sont utilisées par les auteurs pour analyser la structure des préoccupations transverses des patrons de conception. Dans l'étude de HK, le but des implémentations AspectJ [AspectJ] est de modulariser les rôles des patrons. Les auteurs ont conclu que la modularité est améliorée dans 17 des 23 cas et 12 implémentations de patron orientées aspect donnent une meilleure réutilisation. Le degré d'amélioration avec la POA varie selon chaque implémentation de patron.

Alessandro Garcia et al. [Garcia06] ont repris le travail de Hannemann et Kiczales [Hannemann02] et ont présenté une étude quantitative qui compare des solutions Java et leurs équivalents AspectJ pour les 23 patrons GoF. Ils ont utilisé des attributs logiciels comme critères d'évaluation, entre autres, la séparation de préoccupations, le couplage, la cohésion et la taille.

Alessandro Garcia et al. [Garcia06] ont tiré quelques conclusions de leur travail. Ils ont constaté que :

- L'utilisation des aspects améliore le couplage et la cohésion de quelques implémentations de patrons de conception.
- La plupart des solutions orientées aspect améliorent la séparation des préoccupations transverses reliées aux patrons de conception, même si seulement quatre implémentations orientées aspect ont montré une amélioration significative au niveau de la réutilisation des classes.
- L'aspectualisation des patrons de conception réduit le nombre d'attributs de 10 patrons, et diminue le nombre d'opérations et des paramètres respectifs de 12 patrons de conception.

- La relation entre les rôles des patrons et l'application spécifique des préoccupations est parfois si intense qu'il ne semble pas trivial de séparer ces rôles dans des aspects.
- L'utilisation des mesures de couplage, de cohésion et de la taille a été utile pour aider à la détection d'opportunités pour la réfactorisation orientée aspect des patrons de conception.

En plus, ils ont discuté l'adaptabilité des solutions analysées en ce qui concerne la séparation des préoccupations et la détermination d'un modèle qui peut, soi-disant, prédire la modularisation des patrons de conception avec les aspects.

Romain Delamare [Delamare09] a essayé d'évaluer l'impact de l'aspectualisation des classes OO. Dans un premier temps, il a mené une étude empirique faite en deux parties. La première partie porte sur 38 projets écrits avec AspectJ. L'auteur a mesuré différentes métriques pour voir le taux d'utilisation des aspects dans un programme OA. La deuxième partie de cette étude porte sur l'évaluation de la différence en termes de modularité et de testabilité entre les deux programmations OO et OA. Pour le faire, l'auteur a mesuré un ensemble de facteurs sur deux implémentations d'un même système (HealthWatcher), l'une en Java et l'autre en AspectJ.

Concernant la modularité, Dalemare [Delamare09] a calculé plusieurs métriques : le nombre de méthodes, le nombre d'attributs, le nombre de lignes de code et le manque de cohésion (selon la métrique LCOM). Mais, pour comparer la testabilité des deux implémentations, il a mesuré les métriques suivantes : CBO, RFC, RFC` et MPC [24].

Lors de cette étude, l'auteur a utilisé trois outils : Metrics [Metrics] qui calcule les métriques pour les systèmes Java, AjMetrics un outil qu'il a développé lui-même et qui calcule les métriques pour les systèmes AspectJ [AspectJ] et enfin, ABIS [Munoz08] un outil qui vérifie les interactions entre les aspects et le programme de base.

Les résultats obtenus montrent que les développeurs utilisent très peu de greffons. Ils écrivent des greffons très précis qui sont tissés à peu de points de jonction. Ils utilisent, en général, seulement la moitié de l'expressivité du langage AspectJ [AspectJ], et surtout, l'auteur conclut que AspectJ améliore la modularité, mais pas la testabilité, à cause du couplage qui selon lui a augmenté.

Romain Delamare [Delamare09] a fait une étude sur l'analyse de l'impact des aspects sur les cas de test. L'auteur s'est mis dans le cas où le programme de base est testé d'abord puis les aspects sont ajoutés. Sa stratégie consiste à détecter les cas de test impactés par l'introduction des aspects pour les réécrire. L'auteur et son équipe ont implémenté cette analyse dans un outil appelé Vidock. Cet outil analyse les cas de test et sélectionne l'ensemble des cas de test impactés. Il s'appuie sur AJDT (AspectJ Development Tools), un outil de développement pour AspectJ, et sur Spoon, un outil d'analyse statique pour Java. À la fin de cette étude, l'auteur conclut que Vidock est très précis et qu'il y a peu de cas de test impactés.

2.2 L'évaluation de la testabilité des programmes

La testabilité d'un logiciel est définie par le standard ISO comme « les attributs du logiciel qui ont un effet sur l'effort fourni pour valider le produit logiciel » [Hannemann02]. Cette définition a laissé les portes largement ouvertes aux chercheurs pour développer différentes théories et points de vue distincts sur la testabilité d'un logiciel.

Magiel Bruntink et al. [Bruntink04, Bruntink06] reprennent cette définition pour la traduire dans le langage d'un testeur. Ils citent : « la testabilité d'un système logiciel est indicative de l'effort nécessaire pour tester un système ». Ils utilisent cette définition comme point de départ pour avancer dans leurs recherches concernant plusieurs aspects relatifs à la testabilité OO. Ils ont examiné les facteurs ayant une influence sur la

testabilité des systèmes logiciels orientés objet. Leur étude s'est basée sur un point de départ de la littérature pour obtenir un modèle initial de testabilité et des métriques orientées objet existantes, liées à la testabilité. Par la suite, ces métriques ont été évaluées au moyen de cinq études de cas (commerciales et de source libre) de systèmes Java pour lesquelles des scénarios de test JUnit [JUnit] existent déjà. Le but de leur étude était d'identifier et d'évaluer un jeu de métriques qui peut être utilisé pour évaluer la testabilité des classes OO d'un système Java.

Voas et al., dans leur travail « A dynamic failure-based technique » [Voas92], présentent une technique dynamique pour une estimation statistique de trois caractéristiques qui affectent la conduite quantitative d'un programme :

- 1) la probabilité qu'une section particulière d'un programme soit exécutée,
- 2) la probabilité que la section particulière affecte l'état des données, et
- 3) la probabilité que l'état des données produites par cette section ait un effet sur l'output du programme.

Ces trois caractéristiques peuvent être utilisées pour prédire si les erreurs sont susceptibles d'être révélées par le test du logiciel. Ce qui nous intéresse dans cet article, c'est comment Voas et al. [Voas92] ont défini la testabilité logicielle. En fait, ils l'ont définie comme la probabilité qu'une section du logiciel échoue durant sa prochaine exécution pendant le test si elle contient des erreurs. C'est ce qu'ils ont appelé « la sensibilité aux erreurs ». Ils l'expliquent comme une combinaison de probabilités qu'une certaine partie du logiciel soit exécutée, la probabilité qu'une erreur se situe dans cette partie, et la probabilité d'avoir des résultats corrompus qui vont se propager jusqu'à l'output observable du programme.

T.M. Khoshgoftaar et al. [Khoshgoftaar00] adoptent le travail de Voas et al. [Voas92]. Leur travail porte sur une étude de cas de logiciels avioniques en temps réel. Ils partent de l'idée que la testabilité se définit comme la probabilité qu'un cas de test échoue dans un programme qui contient des erreurs.

Voas et Miller [Voas93] présentent une approche relative à la sensibilité d'erreur. Les renseignements sémantiques contenus dans la spécification d'un programme et les documents de conception sont analysés. Ils définissent une borne supérieure de la sensibilité de faute d'une composante qui est donnée par la quantité de perte d'information qui se produit dans celle-ci. La perte d'information peut apparaître de deux façons :

- La perte explicite d'information se produit parce que les valeurs des variables locales à la composante peuvent ne pas être visibles au niveau du système et ne peuvent pas ainsi être inspectées pendant le test.
- La perte implicite d'information est une conséquence du rapport de domaine/gamme (Domain/range ratio (DRR)) de la composante. Le DRR d'une composante est donné par le rapport de la cardinalité des entrées (Input) sur la cardinalité des sorties (Output).

McGregor et al. [McGregor96] ont essayé de déterminer la testabilité d'un système orienté objet. Ils introduisent la mesure « Visibilité de Composante » (VC), qui peut être considérée comme une version adaptée de la mesure de DRR. Le VC a été conçu pour être sensible aux caractéristiques de l'orienté objet comme l'héritage, l'encapsulation, la collaboration et les exceptions. En outre, un but important du VC est sa capacité d'être utilisé pendant les premières phases d'un processus de développement. Le calcul du VC exige ainsi des documents de spécification exacts et complets.

R.A. Khan et K. Mustafa [Khan09] ont examiné la testabilité logicielle orientée objet et ont présenté quelques informations appropriées sur la testabilité au niveau des classes. La testabilité a été identifiée comme un facteur clé de la qualité logicielle et l'accent est mis pour prévoir la testabilité des classes tôt dans le cycle de vie du développement logiciel. Un modèle (basé sur les métriques) pour les programmes orientés objet a été proposé (MTMOOD). Les auteurs ont proposé une suite de métriques orientées objet pouvant servir à l'estimation de la testabilité d'un système, et du moins pour localiser les parties de conception problématiques. L'identification tôt dans le processus de ces parties

pourrait améliorer la qualité du produit final et diminuer l'effort de test. Le modèle proposé a été évalué empiriquement en utilisant des logiciels industriels.

Freedman [Freedman91] propose « Le domaine de testabilité » qui est basé sur les notions d'observabilité et de contrôlabilité. L'observabilité capture le degré auquel une composante produit des sorties correctes pour des entrées données. La notion de contrôlabilité s'étend à la possibilité qu'une composante produise toutes les valeurs de son domaine de production indiqué. L'adaptation d'une composante (de sa spécification) pour qu'elle devienne observable et contrôlable peut être faite en présentant des extensions. Les extensions observables ajoutent des entrées pour représenter des états auparavant implicites dans la composante. Les extensions contrôlables modifient le domaine de production (output) de sorte que toutes les valeurs de production indiquées peuvent être produites. Freedman [Freedman91] propose de mesurer un index d'observabilité et de contrôlabilité pour pouvoir renseigner la testabilité.

Vishal Chowdhary [Chowdhary09] considère la testabilité dans le processus de développement. Il définit la testabilité comme le degré auquel les composantes et les systèmes sont conçus et implémentés d'une façon qu'il est relativement plus facile à les évaluer. Par ailleurs, Vishal Chowdhary [Chowdhary09] s'est concentré sur deux autres mesures à part l'observabilité et la contrôlabilité, qui sont la simplicité et la connaissance de résultats attendus. Il les a définis comme suit :

- Simplicité : une composante plus simple, moins chère à tester.
- Connaissance des résultats attendus : l'output observée est-il correcte?

Jungmayr [Jungmayr02] considère la testabilité d'un point de vue des tests d'intégration. Il se concentre sur les dépendances entre les composantes. Il propose la notion de *dépendances critiques* de test en utilisant des métriques pour les identifier, afin de les enlever par la suite en utilisant une réfactorisation du code (refactoring).

Du point de vue conception, Baudry et al. [Baudry03] analysent la testabilité à partir des diagrammes de classes UML. Ils considèrent la testabilité d'un autre point de vue. Ils avancent que « la testabilité est un facteur qualité qui est inutile s'il n'est pas disponible tôt dans le cycle de vie du logiciel » [Baudry03]. Ils proposent alors l'utilisation de la variation du couplage et des métriques d'interaction entre les classes pour caractériser la testabilité.

2.3 Test des programmes OA

2.3.1 Génération automatique des classes de test

Avec l'évolution des techniques et l'utilisation de la technologie aspect, l'évaluation des programmes OA (en particulier l'impact de l'introduction des aspects dans du code objet) reçoit maintenant beaucoup d'attention. Plusieurs méthodes pour tester les programmes orientés aspect ont vu le jour. Zhao et al. [Zhao07] ont proposé une méthode évaluant le code source tissé (ce qui semble être une voie faisable). Dans leur papier [Zhao07] ils discutent la faisabilité de cette approche et proposent une méthode OO pour le test. Ils considèrent des études de cas (programmes AspectJ) dans lesquelles le code décompilé et tissé est évalué.

AspectJ utilise le tissage compile-time. Ce tisseur utilise une méthode d'instrument bytecode pour obtenir le bytecode tissé. Ainsi, théoriquement le bytecode tissé a toutes les informations statiques et dynamiques du programme original incluant les classes de base et les aspects. Cependant, le bytecode est difficile à lire et à comprendre. Java exécute ses programmes en interprétation du Java bytecode. Donc, le bytecode n'est pas difficile à décompiler et le code décompilé est souvent très semblable au code original [Zhao07].

Le travail de Zhao et al. [Zhao07] consiste dans un premier temps à utiliser JODE [JODE] pour décompiler le bytecode tissé. JODE [JODE] est un paquetage Java

contenant un décompilateur et un optimisateur pour Java. Ce paquetage est disponible sous le GNOU GPL. Le décompilateur lit dans des fichiers de classe et produit quelque chose de semblable à la classe Java originale. Ensuite, les auteurs utilisent CodePro [CodePro] pour générer automatiquement les classes tests des programmes décompilés. CodePro AnalytiX [CodePro] est un outil qui génère automatiquement les classes tests des programmes Java et C++. Il a été intégré à Eclipse, Rational et WebSphere.

Pour commencer, Zhao et al. [Zhao07] ont décompilé un programme orienté aspect qui se compose d'un aspect et d'une classe. Par inspection du code tissé décompilé, ils ont constaté que :

- La qualité du code décompilé est bonne. Le code est bien structuré et a généralement besoin seulement de quelques modifications pour fonctionner correctement.
- Chaque *point de coupe* et chaque *advice* incluant la déclaration d'intertype ont une méthode correspondante dans le code d'aspect tissé. L'invocation est insérée dans certaines places appropriées du code de base.

Puisque le code décompilé est du code Java, Zhao et al. [Zhao07] ont généré automatiquement les classes tests JUnit [JUnit] grâce à CodePro AnalytiX [CodePro]. Puis, ils ont discuté l'efficacité de cette méthode pour tester un programme OA.

L'application de cette méthode sur d'autres exemples (programmes OA) a démontré sa faisabilité et son efficacité. L'avantage de cette méthode est que la décompilation et la génération des suites de test sont entièrement faites automatiquement grâce à des outils fiables et faciles à acquérir. Cependant, il n'est pas toujours évident de décompiler de gros programmes, cela peut provoquer des erreurs ou boguer l'outil (JODE [JODE]). En plus, lors de la découverte de fautes en testant le code décompilé, il peut être difficile ou même impossible dans certains cas (comme en analysant de gros systèmes), de localiser l'erreur au niveau du code source original, afin de la corriger.

2.3.2 Autres techniques de test des programmes OA

Romain Delamare [Delamare09] a proposé des solutions pour le test de programmes orientés aspect. Ces solutions ont été élaborées à la suite d'observations effectuées lors d'études empiriques de programmes orientés aspect. La troisième contribution de sa thèse est l'implémentation d'un outil appelé AdviceTracer qui permet d'écrire des cas de test qui se concentrent sur l'expression de points de coupe. Cet outil est utilisé pour spécifier des oracles qui vérifient la présence ou l'absence d'un greffon dans le flot d'exécution. L'auteur a mené une étude comparative qui consiste à l'utilisation de son outil avec JUnit et l'utilisation de JUnit seul. Il conclut que son outil améliore vraiment les résultats, en réduisant l'effort de test des points de coupe par rapport à JUnit.

Xuand et al. [Xuand06] ont présenté une approche de test incrémental pour les programmes orientés aspect. L'approche considère les aspects comme des modifications incrémentales des classes de base. Leur démarche est organisée en plusieurs étapes : construire les modèles d'états des classes de base; produire des cas abstraits de test des modèles de base puis instancier les scénarios de test abstraits pour former des suites de tests pour les classes de base; tester les classes de base; construire des modèles d'aspect et les tisser dans les modèles de base pour pouvoir produire des scénarios de test abstraits des modèles d'états tissés; enfin, produire des suites de tests pour le programme orienté aspect dans l'ensemble par la réutilisation, la modification et l'extension des scénarios de test des classes de base et des nouveaux scénarios de test abstraits instanciés pour pouvoir tester le programme orienté aspect.

Alexander et al. [Alexander04] ont proposé un modèle de faute pour la programmation orientée aspect. Le modèle inclut six types de fautes. Bien que ce modèle de faute ne constitue pas encore une approche de test entièrement développée, il est sans doute utile pour développer des outils de test et déterminer les stratégies de couverture et de critères.

Ceccato et al. [Ceccato05] ont proposé un processus de test incrémental qui permet de tester le code de base et les fonctionnalités transversales factorisées dans les aspects séparément. Dans leur approche, ils ont testé une application partiellement tissée. Ils tissent une préoccupation transversale par étapes dans une application partiellement tissée. Donc, ce qu'ils testent c'est le code de base et le code source tissé.

Ceccato et al. [Ceccato05] ont utilisé le test d'intégration et la technique de tests de régression pour appuyer leur démarche. Ils ont fait valoir l'utilisation de cette méthode incrémentale pour profiter des avantages de l'AOP lors de la phase des tests. Lors de l'intégration successive de nouveaux aspects, retester l'application de base peut être limité à quelques portions de code touchées par les aspects.

CHAPITRE 3

MÉTRIQUES DE TESTABILITÉ

L'objectif de ce chapitre est de présenter et de définir les métriques que nous allons utiliser par la suite pour réaliser nos expérimentations. Il y a plusieurs mesures proposées dans la littérature pour capturer les divers aspects liés à la qualité des systèmes OO. Nous donnons, dans ce chapitre, une brève définition de quelques métriques OO que nous avons sélectionnées pour nos études empiriques. Les paramètres utilisés sont représentés en cinq catégories : le couplage, la cohésion, l'héritage, la complexité et la taille. Par ailleurs, pour indiquer l'effort de test nécessaire pour une classe logicielle (notée C_s), nous avons utilisé diverses mesures pour quantifier la classe de test JUnit correspondante (notée C_t).

3.1 Métriques Orientées Objet

Pour capturer la testabilité des classes, nous avons décidé de mesurer pour chaque classe logicielle C_s , diverses caractéristiques. Dans ce mémoire, nous avons utilisé la suite de métriques pouvant prédire, selon nous, l'effort nécessaire à écrire les classes test C_t correspondant à C_s :

- *Métrique de couplage :*

- *CBO* : [Chidamber94] Nombre de couplages (degré de dépendance) entre une classe et toutes les autres classes du système (invocation de méthodes ou de variables). Afin d'améliorer la modularité et de favoriser l'encapsulation, le couplage interclasses doit être le plus limité possible. Plus le nombre de couples est élevé, plus les autres parties de la modélisation sont sensibles aux modifications et plus la maintenance est difficile. Par ailleurs, la mesure du couplage s'avère utile pour prévoir le niveau de complexité des tests pour les différentes parties d'une modélisation. Plus le couplage de classes inter-objets est important, plus les tests doivent être rigoureux.

- *Métriques d'héritage :*

- *DOIH* : C'est le nombre maximum de classes ancêtres de la classe pour atteindre la (une) racine. Le mécanisme d'héritage introduit par la technologie objet permet de garantir une meilleure réutilisation, structuration et encapsulation du code. Cependant, l'héritage doit être utilisé de manière minimale et suffisante. Car, d'une part, l'absence d'héritage est souvent synonyme de manque de réutilisation de code et, d'autre part, des arbres d'héritage très profonds affectent la maintenabilité, la testabilité et la lisibilité d'un code [Chidamber94].
- *NOCC* : C'est le nombre de classes enfants. Elle compte le nombre de classes, directement ou indirectement dérivée de la classe mesurée [Chidamber94].

- *Métriques de Halstead :*
 - *HDIFF* (Difficulté Halstead) : Cette mesure est l'une des métriques de « Halstead Science Software » [Halstead77]. Elle est calculée comme suit : "Nombre d'opérateurs uniques / 2 * Nombre d'opérandes / Nombre d'opérandes uniques".
 - *HEFF* (Effort Halstead) : Cette mesure est l'une des métriques de « Halstead Science Software » [Halstead77]. Elle est calculée comme suit : "Volume Programme Halstead * Difficulté Halstead".
- *Métriques de cohésion :*
 - *LCOM* : C'est le nombre de méthodes prises deux à deux (paires de méthodes) ne partageant pas des instances de variables de la classe. Moins il y'a de nombre de paires de méthodes partageants des instances de variables de la classe, moins la classe est cohésive. Si cette valeur est négative, LCOM est fixée (selon la définition des auteurs) à zéro. Une forte valeur indique un faible partage de données entre les méthodes de la classe d'où une perte d'unité structurelle de la classe, ce qui nuit grandement à la réutilisabilité de la classe [Basili96].
- *Métriques de Taille :*
 - *LOC* : C'est le nombre de lignes de code dans une classe. On suppose que plus une classe logicielle contient de lignes de code, plus sa classe test sera importante [Basili96].

- *NOO* : Compte le nombre d'opérations (nombre de méthodes). Si une classe a un nombre d'opérations élevé, il peut être judicieux d'examiner s'il serait approprié de la diviser en sous-classes. Sinon elle sera peut-être difficile à tester [Henderson96].
- *Métriques de complexité* :
 - *RFC* : C'est le nombre de méthodes (de la classe et d'autres classes) potentiellement appelées par une classe en réponse à un message. Une classe qui fournit un ensemble « réponse » plus grand est considérée comme plus complexe et comme nécessitant plus de tests qu'une classe dont la complexité de la modélisation générale est plus faible [Chidamber94].
 -
 - *WMPC* : C'est le nombre de méthodes définies dans une classe. Le nombre de méthodes et leur complexité permettent de prévoir le temps et l'effort requis pour développer et maintenir la classe. La complexité cyclomatique est liée au taux de fautes. Un WMPC élevé est synonyme d'un risque élevé de fautes dans la classe, mais aussi d'une compréhensibilité plus difficile [Chidamber94].

Ces métriques et leurs descriptions sont toutes disponibles dans Together [Together]. C'est l'outil que nous avons utilisé pour le calcul de ces dernières sur les différentes classes des systèmes que nous avons sélectionnés pour réaliser nos expérimentations.

3.2 Métriques de cas test

JUnit [JUnit] est un cadre simple pour écrire et exécuter des tests unitaires automatisés pour des classes Java [Java]. Les cas de test JUnit [JUnit] sont écrits en Java (ou générés automatiquement grâce à CodePro [CodePro]) par les testeurs. JUnit [JUnit] donne aux testeurs certains soutiens afin qu'ils puissent écrire ces cas de test plus facilement. Une utilisation typique de JUnit [JUnit] est de tester chaque classe C_s d'un programme au moyen d'une classe test C_t dédiée. Pour tester la classe C_s , nous avons besoin d'exécuter sa classe test C_t . JUnit [JUnit] fera le rapport du nombre des méthodes ayant réussi ou échoué l'épreuve de test.

Pour capturer la testabilité des classes, nous avons décidé de mesurer pour chaque classe test C_t , correspondante à une classe de logiciels C_s , diverses caractéristiques. Nous avons utilisé la suite de métriques de cas de test suivante :

- *TLOC* : C'est le nombre de lignes de code de la classe test C_t . Si une classe test C_t contient un nombre de lignes de code élevé, cela indique que l'effort fourni pour écrire cette classe test C_t est important [Bruntink04, Bruntink06].
- *TAssert* : Cette métrique donne le nombre d'invocations des méthodes assert JUnit se produisant dans le code d'une classe test C_t . L'ensemble des méthodes assert JUnit sont, en fait, utilisées par les testeurs pour comparer le comportement attendu de la classe sous test à son comportement actuel (réel). Cette métrique est utilisée pour indiquer un autre point de vue de la taille d'une suite de test. Elle est directement liée à la construction du cas de test [Bruntink04, Bruntink06].

- *TLocVariables* : Cette métrique donne le nombre total des variables locales qui sont déclarées dans les méthodes d'une classe test C_t . Ces variables sont utilisées par les testeurs pour supporter le test des méthodes de la classe C_s . Elle est utilisée pour indiquer l'effort nécessaire (création des objets nécessaires) pour tester les méthodes d'une classe.
- *TInvMethods* : Cette métrique donne le nombre de méthodes invoquées par une classe test C_t . Ces méthodes sont utilisées par les testeurs pour supporter les interactions entre les classes en cours de test et les classes avec qui elles collaborent. Cette métrique est utilisée pour indiquer l'effort requis pour tester les méthodes d'une classe couplée à d'autres classes.
- *THDiff* : Cette mesure est l'une des mesures de « Halstead Science Software » [Halstead77]. Cette métrique indique le niveau de difficulté d'une classe test C_t . Nous supposons que cela reflètera la difficulté de la classe test.
- *THEff* : Cette mesure est également l'une des mesures « Halstead Science Software ». Cette métrique donne l'effort nécessaire pour implémenter ou comprendre une classe test C_t . Elle est proportionnelle au volume et au niveau de difficulté de la classe test. Nous supposons que cela reflètera aussi la difficulté de la classe sous test et l'effort requis pour construire la classe de test correspondant.
- *TNOO* : (comme la traditionnelle métrique NOO [Henderson96] pour les classes logicielles) elle donne le nombre d'opérations dans une classe de test C_t .
- *TRFC* : Cette métrique donne la taille de la réponse prévue pour une classe test C_t (comme la traditionnelle métrique RFC [Chidamber94] pour les classes logicielles). Le TRFC d'une classe test C_t est le nombre de ses méthodes et des méthodes des autres classes qui sont appelées par les méthodes de C_t . Une classe C_t qui a un TRFC plus grand que d'autres sera considérée comme plus complexe. Ceci reflète une autre perspective de l'effort de test d'une classe.

- *TWMPC* : Cette métrique (comme la métrique WMPC traditionnelle [Chidamber94] pour les classes logicielles) donne la somme de la complexité de toutes les méthodes d'une classe test C_t (Weighted Methods Per Class : WMPC), où chaque méthode est pondérée par sa complexité cyclomatique. Nous supposons ici que si une classe de test a plus de méthodes qu'une autre, elle est susceptible d'être plus complexe.

L'approche utilisée dans ce mémoire est basée sur les travaux de Bruntik et al. [Bruntink04]. Deux métriques de cas de test (*TLOC* et *TAssert*) ont, en fait, déjà été utilisées par Bruntink et al. dans [Bruntink04, Bruntink06] pour indiquer la taille d'une suite de test. Bruntink et al. se sont basés pour définir ces indicateurs sur le travail de Binder [Binder94].

Nous supposons, dans ce mémoire, que ces deux mesures (*TLOC* et *TAssert*) sont des indicateurs de la testabilité des classes logiciels. Ces métriques reflètent différents facteurs du code source comme indiqué par Bruntink et al. [Bruntink04, Bruntink06] : les facteurs qui influencent le nombre de cas de tests nécessaires pour tester les classes d'un système, et les facteurs qui influencent l'effort nécessaire pour développer chaque cas de test individuellement. Ces deux catégories ont été mentionnées comme les facteurs de générations de cas de test et les facteurs de construction des cas de test.

Toutefois, en analysant le code source des classes tests JUnit [JUnit] des systèmes que nous avons sélectionnés pour nos expériences, nous avons constaté que certaines caractéristiques des classes tests ne sont pas capturées par ces deux métriques (comme l'ensemble des variables locales ou l'ensemble des méthodes appelées qui peuvent indiquer l'effort requis pour tester les méthodes de la classe ou de leurs interactions avec les autres classes avec lesquelles la classe sous test est couplée).

Puisque notre travail est de nature exploratoire, nous avons décidé d'étendre les deux métriques *TLOC* et *TAssert* en utilisant un ensemble complémentaire de métriques. Nous avons utilisé trois autres métriques (TNOO, TRFC et TWMPC) pour capturer des caractéristiques supplémentaires des classes de test.

Cette suite peut être utilisée, en fait, pour évaluer et comparer les classes tests (caractéristiques différentes). En utilisant ces paramètres, cela nous permettra de traiter des questions telles que : Quelles sont les grandes classes de test? Quelles sont les classes de test les plus complexes? Pourquoi certaines classes de test sont plus faciles à écrire que d'autres?

Nous supposons que l'effort nécessaire pour écrire une classe test C_t correspondante à une classe logiciel C_s est proportionnel aux caractéristiques mesurées par la suite des métriques de test utilisées.

CHAPITRE 4

ÉVALUATIONS EMPIRIQUES

Dans ce chapitre, nous présentons les conditions générales dans lesquelles nous avons effectué nos expérimentations ainsi que les démarches que nous avons établies et réalisées pour les valider. Les expérimentations, réalisées sur divers projets d'envergure, se regroupent en trois grandes familles visant à valider nos hypothèses.

La première section de ce chapitre porte sur une analyse empirique d'un modèle de testabilité des programmes orientés objet. Dans cette partie, nous allons présenter le modèle MTMOOD et son adaptation MTMOOP.

La deuxième grande section de ce chapitre porte sur une analyse empirique de l'impact de l'introduction des aspects sur la testabilité des classes OO. Dans cette partie, nous présenterons les résultats d'une étude comparative, au niveau de la testabilité, faite sur les programmes OO et leurs correspondants OA.

Dans la troisième et dernière partie de ce chapitre, nous présenterons les résultats d'une étude empirique comparative réalisée sur des programmes OO dont les classes tests ont été développées par des développeurs versus ces mêmes programmes avec des classes tests générées automatiquement.

4.1 Une analyse empirique d'un modèle de testabilité pour les programmes orientés objet

Nous présentons, dans cette section, un modèle de testabilité pour les programmes orientés objet. Le modèle est, en fait, une adaptation d'un modèle proposé dans la littérature pour évaluer la testabilité des conceptions orientées objet.

L'étude présentée dans cette section vise à explorer empiriquement la capacité du modèle à évaluer la testabilité des classes au niveau du code. Nous avons étudié la testabilité dans la perspective des tests unitaires et de l'effort de test requis.

Nous avons mis en place une étude empirique utilisant des données collectées à partir de deux systèmes logiciels Java [Java] pour lesquels des cas de test JUnit [JUnit] existaient déjà. Pour capturer la testabilité des classes, en termes d'effort de test nécessaire, nous avons utilisé différentes métriques pour quantifier les cas de test correspondants JUnit [JUnit]. Afin d'évaluer la capacité du modèle à prédire la testabilité des classes (les caractéristiques des classes tests correspondantes), nous avons utilisé la corrélation pour faire les tests statistiques.

4.1.1 Adaptation du modèle

Nous avons utilisé dans ce travail un modèle de testabilité, basé sur les métriques, pour les conceptions orientées objet (MTMOOD) proposé par Khan et al. [Khan09]. Le modèle a été développé pour prédire la testabilité des classes au niveau de la conception grâce à l'analyse de quelques propriétés orientées objet dans les diagrammes de classes des projets logiciels.

Le modèle comprend trois propriétés de conception orientée objet : encapsulation, héritage et couplage. L'encapsulation est définie comme une sorte d'abstraction qui impose une séparation nette entre l'interface externe d'un objet et son implémentation interne. L'héritage est défini comme une mesure de la relation « *est-un* » entre les classes. Le couplage est défini comme l'interdépendance d'un objet avec d'autres objets dans une conception.

L'importance relative de chaque conception est pondérée. Les auteurs ont utilisé une analyse de régression linéaire multiple pour obtenir les coefficients. L'efficacité du modèle dans la prédiction de la testabilité d'une conception a été validée en utilisant des données collectées à partir des diagrammes de classes de différents projets industriels de taille moyenne. En fait, les auteurs ont montré qu'en général, la testabilité estimée à l'aide du modèle MTMOOD avait une corrélation statistiquement significative avec l'évaluation faite par des évaluateurs indépendants.

Le modèle de calcul utilisé pour évaluer la testabilité d'une classe au niveau du design est (MTMOOD) :

$$\text{Testabilité} = -0,08 * \text{Encapsulation} + 1,12 * \text{Héritage} + 0,97 * \text{Couplage}.$$

Pour mesurer les propriétés de conception considérées, Khan et al. [Khan09] ont utilisé trois métriques au niveau des diagrammes de classes :

- La métrique ENM (métriques d'encapsulation) qui compte le nombre de toutes les méthodes définies dans une classe,
- La métrique REM (une métrique de réutilisation et d'héritage) qui compte pour chaque classe la profondeur de son arbre d'héritage dans la conception, et

- La métrique de CPM (une métrique de couplage) qui compte le nombre de classes avec lesquelles une classe est liée. Ces mesures sont toutes calculées à partir des diagrammes de classes.

Puisque notre travail est de nature exploratoire, nous avons adapté ce modèle au niveau du code en utilisant les métriques de niveau code source suivantes :

- NOO qui donne le nombre d'opérations dans une classe [Henderson96],
- DIT (ou DOIH) qui donne la profondeur de l'arbre d'héritage pour une classe [Chidamber94], et
- CBO qui donne le nombre de classes avec lesquelles une classe donnée est couplée [Chidamber94].

L'équation suivante donne la formule de calcul que nous avons utilisée pour évaluer la testabilité d'une classe au niveau du code source (MTMOOP) :

$$Testabilité = -0,08 * NOO + 1,12 * DIT + 0,97 * CBO.$$

4.1.2 Cadre expérimental

Le but de cette étude est d'explorer empiriquement la relation entre le modèle MTMOOP et la testabilité des classes dans les systèmes OO (en termes d'effort de test). Nous estimons la testabilité en utilisant le modèle MTMOOP au niveau des classes, et nous limitons l'effort de test aux tests unitaires des classes seulement. Afin d'obtenir des résultats significatifs, les données utilisées dans nos expériences ont été recueillies à partir de deux systèmes logiciels Java « *open source* ».

Cette sélection a été essentiellement basée sur le nombre de classes ayant subi des tests en utilisant l'outil JUnit [JUnit]. Pour nos expériences, nous avons sélectionné à partir de chacun des systèmes utilisés, seulement les classes pour lesquelles des cas de test JUnit [JUnit] ont été développées.

4.1.2.1 Collecte des données

Nous avons utilisé le modèle MTMOOP pour estimer la testabilité des classes pour lesquelles des cas de test JUnit [JUnit] ont été développés, et les métriques de cas de test pour quantifier les cas de test JUnit [JUnit].

Les métriques relatives au code source et les métriques de test ont été calculées en utilisant l'outil Borland Together [Together].

4.1.2.2 Hypothèses et analyse statistique

Nous présentons, dans cette section, la méthodologie de l'étude empirique que nous avons menée afin d'évaluer (explorer) la relation entre le modèle MTMOOP et la testabilité des classes (caractéristiques mesurées des classes de test correspondantes) [Kout11]. Nous avons effectué des tests statistiques à l'aide de corrélations. Les hypothèses (nulle et alternative) que nous avons vérifiées lors de nos expériences sont :

- H0 : Il n'y a pas de corrélation significative entre le modèle MTMOOP et la testabilité,
- H1 : Il existe une corrélation significative entre le modèle MTMOOP et la testabilité.

Dans cette expérimentation, rejeter l'hypothèse nulle montre qu'il y a une relation statistiquement significative entre le modèle MTMOOP et les métriques de cas de test (niveau de signification choisi $\alpha = 0,05$). Pour l'analyse des données recueillies, nous avons préféré une mesure non paramétrique de la corrélation. Nous avons utilisé le coefficient de corrélation de Spearman. Cette technique, basée sur les rangs des observations, est largement utilisée pour mesurer le degré de linéarité des relations entre deux variables (deux ensembles de données classées). Elle mesure le degré de rapprochement des grappes de données classées, autour d'une ligne droite.

Le coefficient de corrélation de Spearman peut prendre une valeur comprise entre -1 et +1. Une corrélation positive signifie que les rangs des deux variables augmentent ensemble. Une corrélation négative signifie que quand les rangs d'une variable augmentent, les rangs de l'autre variable diminuent. Une corrélation de +1 ou -1 se posera si la relation entre les rangs est exactement linéaire. Une valeur de corrélation proche de zéro signifie qu'il n'y a pas de relation linéaire entre les rangs. Nous avons utilisé le logiciel XLSTAT [XLSTAT] pour effectuer l'analyse statistique.

4.1.3 Étude de Cas

4.1.3.1 Systèmes sélectionnés

Les systèmes sélectionnés sont les suivants :

- ANT (www.apache.org) : un outil de « *build* » basé sur Java, avec des fonctionnalités similaires aux utilitaires « *make* » d'Unix,
- JFreeChart (<http://www.jfree.org/jfreechart>) : une bibliothèque de diagrammes gratuite pour la plate-forme Java [Java].

CLASSES LOGICIELLES - CLASSES TESTS						
SYSTÈMES	#Classes	MLOC	WMPC	#TClasses	MLOC	MWMP
ANT	713	89.85	17.1	115	153.52	30.37
JFC	496	137.73	28.09	230	231	46.08

TABLEAU 1 : QUELQUES CARACTÉRISTIQUES DES SYSTÈMES UTILISÉS.

Le tableau 1 récapitule quelques caractéristiques des systèmes analysés : Le nombre des classes logicielles de chaque système, la moyenne de lignes de code des classes logicielles, la moyenne de la complexité cyclomatique des classes logicielles, le nombre de classes test JUnit [JUnit] qui ont été développées pour chaque système, la moyenne des lignes de code et la moyenne de la complexité cyclomatique des classes logicielles pour lesquelles des classes test JUNIT ont été développées.

Les premières observations que nous pouvons déjà faire sont :

- Uniquement un sous-ensemble de classes logicielles a été testé avec JUnit [JUnit],
- Le pourcentage des classes évaluées varie d'un système à un autre, et
- Les classes logicielles pour lesquelles des scénarios de test JUNIT ont été développés sont, en général, des classes de taille et de complexité importantes.

4.1.3.2 Résultats

Les tableaux 2 et 3 résument les résultats de l'analyse de corrélation. Le tableau 2 montre, pour chacun des systèmes sélectionnés et entre chaque paire de mesures distinctes (MTMOOP, métriques de cas test), les valeurs obtenues par le coefficient de corrélation de Spearman r_s . Nous avons également calculé le coefficient de corrélation de Spearman r_s pour chaque paire de métriques de cas de test (tableau 3). Les coefficients de corrélation de Spearman obtenus sont significatifs (à $\alpha = 0,05$) et ils sont mis en gras dans les deux tableaux. Cela signifie que pour les paires correspondantes des mesures, il existe une corrélation d'un niveau de confiance de 95 %.

ANT	TLOC	TAssert	TNOO	TRFC	TWMPC
MTMOOP	0.325	0.030	0.403	0.507	0.357
JFC	TLOC	TAssert	TNOO	TRFC	TWMPC
MTMOOP	0.283	0.236	0.260	0.335	0.265

TABLEAU 2 : VALLEURS DE CORRÉLATION ENTRE LE MTMOOP ET LES MÉTRIQUES DE CAS DE TEST.

ANT	TLOC	TAssert	TNOO	TRFC	TWMPC
	1 0.679 0.543 0.475 0.628				
		1 0.151 0.064 0.201			
			1 0.811 0.927		
				1 0.781	
					1
JFC	TLOC	TAssert	TNOO	TRFC	TWMPC
	1 0.848 0.739 0.839 0.740				
		1 0.586 0.713 0.581			
			1 0.762 0.996		
				1 0.762	
					1

TABLEAU 3 : VALEURS DE CORRÉLATION ENTRE LES MÉTRIQUES DE CAS DE TEST.

La première observation globale que nous pouvons faire est que les résultats obtenus soutiennent l'idée qu'il y a une relation significative statistiquement entre le modèle MTMOOP les métriques de cas de test utilisées.

Les coefficients de corrélation de Spearman obtenus entre le modèle MTMOOP et les métriques de cas de test sont globalement significatifs (à $\alpha = 0,05$) pour les deux systèmes sélectionnés (sauf pour le système ANT entre le modèle MTMOOP et la métrique TAssert). Nous pouvons raisonnablement rejeter l'hypothèse H_0 et accepter l'hypothèse H_1 .

Par ailleurs, les mesures ont une corrélation positive. Comme mentionné précédemment, une corrélation positive indique que les rangs d'une variable (MTMOOP dans notre cas) augmentent à mesure que les rangs de l'autre variable augmentent (métriques de cas de test). Ces résultats sont plausibles et n'ont rien d'étonnant sachant la définition du modèle MTMOOP et celles des métriques de cas de test.

L'autre constatation globale que nous pouvons faire est que les métriques de test sont également corrélées entre elles (tableau 3). Pour ANT, le modèle MTMOOP est significativement meilleur prédicteur des métriques de cas de test TRFC, TNOO que TLOC, TAssert et TWMP. La corrélation entre le modèle MTMOOP et la métrique de cas de test TAssert, en particulier, n'est pas significative. Pour JFreeChart, le modèle MTMOOP est meilleur prédicteur des métriques de cas de test TRFC et TLOC que TAssert, TNOO et TWMP. Les résultats pour JFreeChart montrent également que les valeurs de corrélation sont, globalement, plus faibles que les valeurs de corrélation obtenues dans le cas de l'ANT. Par ailleurs, pour les deux systèmes, ANT et JFreeChart, les valeurs de corrélation (bien que significatives) ne sont pas élevées.

L'étude réalisée et présentée dans cette section devrait être reproduite en utilisant plus de systèmes pour tirer des conclusions plus générales sur la relation entre le modèle MTMOOP et les métriques de cas de test. En fait, il y a un certain nombre de limitations qui peuvent affecter les résultats de l'étude ou limiter leur généralisation.

Les résultats obtenus sont basés sur le jeu de données que nous avons rassemblées des systèmes analysés. Comme mentionnés précédemment, nous avons utilisé seulement un sous-ensemble de classes et leurs scénarios de test JUnit [JUnit] correspondant (voir le Tableau 1) de chacun des systèmes analysés.

L'étude devrait être reproduite sur un plus grand nombre de systèmes orientés objet pour augmenter la généralité des résultats. De plus, les classes pour lesquelles des scénarios de test JUnit [JUnit] ont été développés sont des classes relativement grandes et complexes (voir le Tableau 1). Ceci est vrai pour les deux systèmes que nous avons analysés. Ceci peut affecter les résultats de notre étude. Cela dépend de la méthodologie suivie par les développeurs pour développer les classes tests et les critères qu'ils ont utilisés pour choisir les classes logicielles pour lesquelles ils ont développé des classes de test (aléatoirement, selon leur taille, leur complexité, ou bien selon d'autres critères). Les résultats peuvent être différents.

Il serait intéressant de reproduire cette étude en utilisant des systèmes pour lesquels des scénarios de test JUnit [JUnit] ont été développés pour un maximum de classes logicielles. Ceci permettra aussi d'observer des valeurs de corrélation entre le modèle MTMOOP et les métriques de cas de test pour différents types de classes (petites, moyennes et grandes). Aussi, en analysant le code source des classes tests JUnit [JUnit], nous avons observé que, dans de nombreux cas, ils ne couvrent pas toutes les méthodes des classes logicielles correspondantes. Ceci peut aussi affecter les résultats de l'étude et expliquer pourquoi les valeurs de corrélation obtenues sont plutôt relativement basses dans quelques cas. Il est aussi possible que le style de développement utilisé par les développeurs pour écrire les scénarios de test puisse affecter les résultats obtenus ou produire des résultats différents pour des applications spécifiques.

4.1.4 Conclusion

Dans cette section, nous avons présenté un modèle de testabilité basé sur les métriques pour les programmes orientés objet. Nous avons examiné empiriquement le potentiel du modèle MTMOOP dans la prédiction de la testabilité des classes au niveau du code. La testabilité a été examinée du point de vue des tests unitaires et en évaluant l'effort de test. Nous avons conçu une étude empirique en utilisant des données rassemblées de deux systèmes logiciels Java [Java] (Open source) pour lesquels des scénarios de test JUnit [JUnit] existent déjà. Pour capturer la testabilité des classes, nous avons utilisé diverses métriques pour évaluer quantitativement les différentes caractéristiques des cas de test JUnit [JUnit] correspondants. Pour évaluer la capacité du modèle au niveau de la prédiction de la testabilité des classes, nous avons utilisé des tests statistiques utilisant la corrélation. Les résultats réalisés soutiennent l'idée qu'il y a une relation statistiquement significative entre le modèle MTMOOP et les métriques de cas de test utilisés.

Cependant, l'étude réalisée devrait être reproduite en utilisant plusieurs autres systèmes pour tirer des conclusions plus générales. La contribution de cette étude devrait être vue comme exploratoire et indicative plutôt que conclusive. De plus, sachant que la testabilité logicielle est affectée par différents facteurs, il serait intéressant d'étendre la suite de métriques de cas de test utilisées pour mieux refléter l'effort de test.

4.2 Une analyse empirique de l'impact de l'introduction des aspects sur la testabilité des classes OO

La présente section porte sur une analyse empirique de l'impact de l'introduction des aspects sur la testabilité des classes OO. Dans cette partie, nous présenterons les résultats d'une étude comparative au niveau de la testabilité faite sur des programmes OO et les programmes correspondants OA.

4.2.1 Cadre expérimental

4.2.1.1 Collecte des données

Nous avons utilisé le modèle MTMOOP pour estimer la testabilité des classes pour lesquelles des cas de test JUnit [JUnit] ont été développés, et les métriques de cas test pour quantifier les cas de test JUnit [JUnit].

Les métriques relatives au code source des programmes et les métriques de test ont été calculées en utilisant l'outil Borland Together [Together].

4.2.1.2 Hypothèses et analyse statistique

Nous présentons, dans cette section, la méthodologie de l'étude empirique que nous avons menée afin d'évaluer l'impact de l'introduction des aspects sur la testabilité des programmes. Pour y arriver, nous avons mené une étude empirique basée sur le code source des classes OO des programmes en mesurant les valeurs de MTMOOP et d'autres métriques logicielles, d'une part. Puis, nous avons évalué empiriquement le code source des classes tests en nous basant sur notre suite de métriques de testabilité, d'autre part.

Nous avons effectué des tests statistiques basés sur les corrélations. Les hypothèses nulles et alternatives que nous avons vérifiées lors de nos expérimentations sont :

- H_0 : La testabilité des classes OO n'est pas améliorée après l'aspectualisation du code des préoccupations transverses.
- H_1 : La testabilité des classes OO est améliorée après l'aspectualisation du code des préoccupations transverses.

Dans cette étude, l'acceptation de H_1 signifie qu'il y a une amélioration de la testabilité des classes statistiquement significative après l'aspectualisation du code source des préoccupations transverses (le niveau de signification choisi est $\alpha=0,05$). Pour l'analyse des données recueillies, nous avons utilisé le coefficient de corrélation de Spearman et celui de Pearson.

La corrélation de Spearman basée sur les rangs des observations est largement utilisée pour mesurer le degré de linéarité des relations entre deux variables (deux ensembles de données classées). Elle mesure le degré de rapprochement des grappes de données classées, autour d'une ligne droite. Le coefficient de corrélation de Spearman peut prendre une valeur comprise entre -1 et +1. Une corrélation positive signifie que les rangs des deux variables augmentent ensemble. Une corrélation négative signifie que quand les rangs d'une variable augmentent, les rangs de l'autre variable diminuent. Une corrélation de +1 ou -1 se posera si la relation entre les rangs est exactement linéaire. Une valeur de corrélation proche de zéro signifie qu'il n'y a pas de relation linéaire entre les rangs.

Le coefficient de corrélation de Pearson est la corrélation la plus communément utilisée. Elle est adaptée à nos données quantitatives, malgré qu'elles soient discontinues. La valeur de la corrélation de Pearson est comprise entre -1 et 1, et elle mesure le niveau de relation linéaire entre deux variables. Les p-values calculées pour les coefficients de corrélation permettent de tester l'hypothèse nulle de corrélation non significativement différente de zéro entre les variables.

Cependant, il faut être prudent quant à l'interprétation des corrélations. En effet, si l'indépendance entre deux variables implique la nullité du coefficient de corrélation entre les variables, la réciproque n'est pas vraie : on peut avoir une corrélation proche de zéro entre deux variables parce que la relation n'est pas linéaire, ou parce qu'elle est complexe et nécessite la prise en compte d'autres variables. Le coefficient de corrélation de Spearman utilise les rangs des observations et non leur valeur en tant que telle. Ce coefficient est donc en théorie plus adapté aux données ordinaires sans beaucoup d'*æquo*. Comme pour le coefficient de Pearson, on peut aussi interpréter ce coefficient en termes de variabilité expliquée. Ici, il s'agit bien entendu de la variabilité des rangs.

Nous avons utilisé le logiciel XLSTAT [XLSTAT] pour effectuer l'analyse statistique.

4.2.2 Étude de Cas

4.2.2.1 Systèmes sélectionnés

Pour réaliser notre expérience, les données utilisées dans notre étude empirique ont été rassemblées à partir de quatre systèmes logiciels « *Open source* », Java et leurs équivalents AspectJ. Cette sélection a été essentiellement basée sur le nombre de classes aspectualisées ayant subi le test utilisant le Framework JUnit.

Les systèmes sélectionnés sont :

Healthwatcher (<http://www.comp.lancs.ac.uk/~greenwop/ecoop07/>) : Un système d'information basé sur le Web qui a été développé pour le Bureau de santé de la ville de Recife, au Brésil. Le système vise à améliorer la qualité des services fournis par les établissements de santé, permettant aux citoyens de déposer des plaintes au sujet des questions de santé, et des établissements de santé pour enquêter et prendre les mesures nécessaires.

Telestrada (<http://www.kevijnhoffman.com/icse2008/>) : Un système d'information pour les voyageurs. Il est en cours de développement pour l'administration de la route nationale brésilienne. Il permet à ses utilisateurs d'enregistrer et de visualiser des informations sur les routes brésiliennes.

Petstore (<http://www.kevijnhoffman.com/icse2008/>) : Une démo pour la plateforme J2EE qui est représentative des applications de commerce électronique existantes.

HSQLBD (<http://hsqldb.org/>) : Ce système est en réalité connu sous le nom de HyperSQL. HSQLDB (HyperSQL DataBase). C'est un moteur de base de données relationnelle SQL écrit en Java. Il supporte presque tout l'ANSI SQL-92 (format BNF) et le noyau SQL 2008 complet. Il offre un petit moteur de base de données rapide, multithread et transactionnelle avec mémoire et tableaux basés sur les disques et supporte les modes serveur embarqués. Il inclut des outils tels qu'un outil de ligne de commande SQL et des outils d'interrogation graphique.

JHotdraw (<http://sourceforge.net/projects/jhotdraw/>) : C'est un cadre graphique bidimensionnel pour les éditeurs de dessin structuré. Cet outil est écrit en Java et il a été re-factorisé en AspectJ.

Voici quelques caractéristiques des systèmes étudiés :

SYSTÈMES OO	CLASSES LOGICIELLES - CLASSES TEST					
	#Classes	MLOC	WMPC	#TClasses	MLOC	MWMP
Healthwatcher	96	69,18	8,19	88	156	20,29
Telestrada	233	14,69	2,67	223	39,3	6,2
Petstore	345	51,58	10,8	124	156,1	19,27
HSQLBD	379	193,18	34,7	199	700,25	54,2
JHotdraw	349	60,73	13,35	235	178,79	18,93

TABLEAU 4 : CARACTÉRISTIQUES DES SYSTÈMES JAVA UTILISÉS.

SYSTÈMES OA	CLASSES LOGICIELLES - CLASSES TESTS						
	#Classes	#Aspects	MLOC	WMPC	#TClasses	MLOC	MWMP
Healthwatcher	96	36	64,57	8,93	88	86,58	23
Telestrada	240	18	13,91	2,61	220	33,73	5,5
Petstore	345	38	49,67	5,3	168	39,6	6,15
HSQLBD	340	24	157,17	30,67	199	651,07	53,38
AJHotdraw	340	31	59,27	13,16	229	120,76	15,09

TABLEAU 5 : QUELSQUE CARACTÉRISTIQUE DES SYSTÈMES ASPCTJ UTILISÉS.

Les tableaux 4 et 5 récapitulent quelques caractéristiques des systèmes analysés : Le nombre des classes logicielles de chaque système, le nombre des aspects par système (Juste dans le Tableau 5), la moyenne des lignes de code des classes logicielles, la moyenne de la complexité cyclomatique des classes logicielles, le nombre de classes test JUNIT qui ont été développées pour chaque système, la moyenne des lignes de code et la moyenne de la complexité cyclomatique des classes logicielles pour lesquelles des classes test JUNIT ont été développées.

Les premières observations que nous pouvons déjà faire sont :

- Un grand sous-ensemble de classes logicielles (choisi aléatoirement) a été testé avec JUnit [JUnit],
- Le pourcentage des classes évaluées varie d'un système à un autre, et
- Les classes logicielles pour lesquelles des scénarios de test JUnit [JUnit] ont été développés sont, en général, des classes de taille et de complexité importantes.

4.2.2.2 Résultats

L'analyse des données est réalisée, dans un premier temps, grâce aux coefficients de corrélation de Pearson et celle de Spearman ensemble pour déterminer la validité des données recueillies et voir la relation entre les métriques sélectionnées et surtout la corrélation entre le modèle MTMOOP et notre suite de métriques de testabilité (TNofCalled, TNofNewData, TNofAssert, TNOO, TLOC, TRFC, TINVOK et TWMPC).

Rappelons que notre objectif est de voir l'impact de l'aspectualisation des classes OO sur la testabilité. Pour avancer dans notre sujet, nous avons sélectionné juste les classes qui ont été aspectualisées (programmes AspectJ) et leurs équivalentes dans la version Java de

chaque système. Puis, nous avons tracé les courbes respectives des valeurs obtenues du modèle MTMOOP et des métriques de cas de test recueillies sur ces classes.

Dans ces graphiques, nous avons combiné les résultats des versions Java et AspectJ pour des fins de comparaison entre les deux versions de chaque programme. Ces figures nous permettront de :

- Visualiser la relation entre le MTMOOP et les autres métriques. (on rappelle que le modèle MTMOOP est calculé sur les classes logicielles des programmes tandis que les métriques de testabilité sont calculées sur les classes tests JUNIT générées par CodePro [CodePro]), et,
- Schématiser l'impact de l'aspectualisation des classes OO sur chaque attribut mesuré.

La dernière partie de cette étude d'impact est un test t (test statistique) sur les données que nous avons collectées (et tracées) précédemment. Ce test va nous permettre d'accepter une des deux hypothèses définies au début de cette étude.

Le test t (ou test de Student) désigne un ensemble de tests d'hypothèse paramétriques où la statistique calculée suit une loi de Student lorsque l'hypothèse nulle est vraie. Un test t peut être utilisé notamment pour tester statistiquement les valeurs obtenues par la différence de la valeur d'une variable à un moment t_1 et la valeur de cette même variable au moment t_2 .

Le test t utilisé est le suivant :

$$T = \frac{\bar{X}}{\frac{\sigma}{\sqrt{n}}}, \text{ qui peut prendre cette forme aussi, } T = \sqrt{n} * \frac{\bar{X}}{\sigma}$$

Où :

\bar{X} est la moyenne des données dans l'échantillon.

n est la taille de notre échantillon.

σ est l'Écart-type dans l'échantillon.

Notons que X est la différence entre deux mesures identiques (la même métrique) calculées sur la même classe avant et après aspectualisation.

Notons que notre X va déterminer le signe de notre résultat t , alors :

Si $X > 0$, cela veut dire que la valeur obtenue en calculant une métrique donnée sur la classe aspectualisée est plus petite que la valeur obtenue en calculant cette même métrique sur la classe Java originale (avant aspectualisation).

Si $X < 0$, cela veut dire que la valeur obtenue en calculant une métrique donnée sur la classe aspectualisée est plus grande que la valeur obtenue en calculant cette même métrique sur la classe Java originale (avant aspectualisation).

Si $X = 0$, cela veut dire tout simplement qu'en aspectualisant une classe donnée, il n'y a pas eu d'impact au niveau de l'attribut mesuré sur cette classe.

Pour vérifier si le résultat de notre test t est significatif ou pas, nous nous sommes basés sur le Tableau de Student en complétant les paramètres de cette équation :

$$|T| > t_{\alpha/2, n-2}$$

Avec α comme degré de signification désiré. Dans notre cas, nous avons choisi 5 %.

- Projet 1 : Healthwatcher

Variables	HDiff	HEff	FO	LCOM1	LCOM3	LOC	NOA	NOCC	RFC	WMPC	CBO	DOIH	NOO	MTMOOP	TNofCalled	TNofNewData	TNofAssert	TNOO	TLOC	TRFC	TINVOK	TWMPC
HDiff	1	0,695	0,668	-0,311	-0,297	0,647	0,244	-0,052	0,740	0,779	0,122	0,182	0,500	0,102	0,651	0,569	0,530	0,604	0,656	0,666	0,589	0,647
HEff	0,695	1	0,604	-0,148	0,051	0,871	0,129	-0,042	0,742	0,789	0,358	0,014	0,360	0,329	0,625	0,511	0,460	0,593	0,787	0,730	0,817	0,732
FO	0,668	0,604	1	-0,149	-0,099	0,736	0,233	-0,058	0,869	0,803	0,397	0,323	0,697	0,371	0,673	0,492	0,662	0,758	0,811	0,860	0,813	0,847
LCOM1	-0,311	-0,148	-0,149	1	0,454	0,039	0,907	0,816	-0,021	0,179	-0,114	-0,096	0,621	-0,158	-0,030	0,018	0,020	-0,003	-0,035	-0,020	-0,060	-0,021
LCOM3	-0,297	0,051	-0,099	0,454	1	0,165	0,697	0,217	0,173	0,206	-0,059	0,205	0,444	-0,066	0,134	-0,079	0,299	0,076	0,114	0,096	0,125	0,095
LOC	0,647	0,871	0,736	0,039	0,165	1	0,226	-0,088	0,789	0,812	0,567	0,121	0,503	0,534	0,666	0,520	0,539	0,745	0,887	0,861	0,850	0,865
NOA	0,244	0,129	0,233	0,907	0,697	0,226	1	-0,004	0,520	0,581	-0,460	0,287	0,785	-0,481	0,435	0,394	0,592	0,253	0,218	0,272	0,226	0,232
NOCC	-0,052	-0,042	-0,058	0,816	0,217	-0,088	-0,004	1	-0,100	-0,068	-0,097	-0,326	-0,044	-0,119	-0,151	-0,120	-0,133	-0,123	-0,127	-0,115	-0,055	-0,112
RFC	0,740	0,742	0,869	-0,021	0,173	0,789	0,520	-0,100	1	0,945	0,149	0,453	0,798	0,131	0,874	0,699	0,880	0,765	0,839	0,861	0,799	0,837
WMPC	0,779	0,789	0,803	0,179	0,206	0,812	0,581	-0,068	0,945	1	0,110	0,246	0,847	0,072	0,772	0,614	0,749	0,699	0,804	0,820	0,836	0,796
CBO	0,122	0,358	0,397	-0,114	-0,059	0,567	-0,460	-0,097	0,149	0,110	1	0,079	-0,106	0,997	0,195	0,030	-0,020	0,522	0,557	0,518	0,329	0,562
DOIH	0,182	0,014	0,323	-0,096	0,205	0,121	0,287	-0,326	0,453	0,246	0,079	1	0,354	0,134	0,641	0,310	0,711	0,585	0,410	0,443	-0,066	0,435
NOO	0,500	0,360	0,697	0,621	0,444	0,503	0,785	-0,044	0,798	0,847	-0,106	0,354	1	-0,141	0,622	0,448	0,736	0,559	0,566	0,633	0,597	0,598
MTMOOP	0,102	0,329	0,371	-0,158	-0,066	0,534	-0,481	-0,119	0,131	0,072	0,997	0,134	-0,141	1	0,203	0,025	-0,011	0,523	0,544	0,504	0,279	0,549
TNofCalled	0,651	0,625	0,673	-0,030	0,134	0,666	0,435	-0,151	0,874	0,772	0,195	0,641	0,622	0,203	1	0,827	0,912	0,807	0,827	0,816	0,552	0,797
TNofNewData	0,569	0,511	0,492	0,018	-0,079	0,520	0,394	-0,120	0,699	0,614	0,030	0,310	0,448	0,025	0,827	1	0,711	0,466	0,547	0,522	0,478	0,499
TNofAssert	0,530	0,460	0,662	0,020	0,299	0,539	0,592	-0,133	0,880	0,749	-0,020	0,711	0,736	-0,011	0,912	0,711	1	0,665	0,666	0,690	0,512	0,658
TNOO	0,604	0,593	0,758	-0,003	0,076	0,745	0,253	-0,123	0,765	0,699	0,522	0,585	0,559	0,523	0,807	0,466	0,665	1	0,946	0,963	0,535	0,966
TLOC	0,656	0,787	0,811	-0,035	0,114	0,887	0,218	-0,127	0,839	0,804	0,557	0,410	0,566	0,544	0,827	0,547	0,666	0,946	1	0,989	0,749	0,991
TRFC	0,666	0,730	0,860	-0,020	0,096	0,861	0,272	-0,115	0,861	0,820	0,518	0,443	0,633	0,504	0,816	0,522	0,690	0,963	0,989	1	0,742	0,998
TINVOK	0,589	0,817	0,813	-0,060	0,125	0,850	0,226	-0,055	0,799	0,836	0,329	-0,066	0,597	0,279	0,552	0,478	0,512	0,535	0,749	0,742	1	0,729
TWMPC	0,647	0,732	0,847	-0,021	0,095	0,865	0,232	-0,112	0,837	0,796	0,562	0,435	0,598	0,549	0,797	0,499	0,658	0,966	0,991	0,998	0,729	1,000

Tableau 6.1 : Matrice de corrélation (Pearson) de Healthwatcher Java.

Variables	HDifIT	HEIT	FO	LCOM1	LCOM3	LOC	NOA	NOCC	RFC	WMPC	CBO	DOIH	NOO	MTMOOP	TNCalled	TNNcwData	TNAsset	TNOO	TLOC	TRFC	TINVOK	TWMP
HDifIT	1	0,973	0,783	-0,433	-0,492	0,163	0,748	-0,006	0,828	0,811	-0,257	0,260	0,757	-0,262	0,630	0,672	0,845	0,429	0,331	0,501	0,753	0,361
HEIT	0,973	1	0,767	-0,144	-0,139	0,270	0,835	0,023	0,879	0,865	-0,239	0,329	0,823	-0,246	0,687	0,710	0,885	0,509	0,430	0,576	0,761	0,444
FO	0,783	0,767	1	-0,416	-0,428	0,207	0,505	0,241	0,842	0,795	0,010	0,149	0,764	0,013	0,550	0,603	0,651	0,342	0,393	0,540	0,758	0,433
LCOM1	-0,433	-0,144	-0,416	1	0,718	0,231	0,688	0,403	0,204	0,251	-0,384	0,168	0,454	-0,435	0,111	0,072	0,261	0,031	0,121	0,077	0,069	0,031
LCOM3	-0,492	-0,139	-0,428	0,718	1	0,235	0,907	0,346	0,161	0,214	-0,420	0,268	0,409	-0,419	0,075	-0,073	0,243	-0,025	0,061	0,018	0,029	-0,028
LOC	0,163	0,270	0,207	0,231	0,235	1	0,137	-0,060	0,201	0,253	0,591	0,217	0,230	0,564	0,543	0,465	0,231	0,694	0,715	0,678	0,197	0,694
NOA	0,748	0,835	0,505	0,688	0,907	0,137	1	0,052	0,824	0,833	-0,519	0,402	0,842	-0,523	0,516	0,494	0,869	0,253	0,229	0,339	0,686	0,186
NOCC	-0,006	0,023	0,241	0,403	0,346	-0,060	0,052	1	0,146	0,217	-0,124	-0,523	0,267	-0,154	-0,276	-0,193	-0,187	-0,185	-0,063	0,004	0,096	-0,005
RFC	0,828	0,879	0,842	0,204	0,161	0,201	0,824	0,146	1	0,958	-0,307	0,378	0,955	-0,307	0,624	0,611	0,885	0,317	0,379	0,500	0,840	0,357
WMPC	0,811	0,865	0,795	0,251	0,214	0,253	0,833	0,217	0,958	1	-0,334	0,220	0,990	-0,349	0,542	0,510	0,837	0,334	0,376	0,533	0,853	0,395
CBO	-0,257	-0,239	0,010	-0,384	-0,420	0,591	-0,519	-0,124	-0,307	-0,334	1	-0,050	-0,364	0,997	0,105	0,065	-0,329	0,380	0,379	0,276	-0,251	0,400
DOIH	0,260	0,329	0,149	0,168	0,268	0,217	0,402	-0,523	0,378	0,220	-0,050	1	0,232	-0,022	0,605	0,494	0,578	0,269	0,263	0,172	0,178	0,114
NOO	0,757	0,823	0,764	0,454	0,409	0,230	0,842	0,267	0,955	0,990	-0,364	0,232	1	-0,378	0,508	0,470	0,823	0,283	0,355	0,506	0,846	0,361
MTMOOP	-0,262	-0,246	0,013	-0,435	-0,419	0,564	-0,523	-0,154	-0,307	-0,349	0,997	-0,022	-0,378	1	0,101	0,065	-0,326	0,355	0,357	0,251	-0,253	0,373
TNofCalled	0,630	0,687	0,550	0,111	0,075	0,543	0,516	-0,276	0,624	0,542	0,105	0,605	0,508	0,101	1	0,952	0,746	0,802	0,806	0,799	0,558	0,726
TNoData	0,672	0,710	0,603	0,072	-0,073	0,465	0,494	-0,193	0,611	0,510	0,065	0,494	0,470	0,065	0,952	1	0,685	0,761	0,745	0,759	0,514	0,685
TAsset	0,845	0,885	0,651	0,261	0,243	0,231	0,869	-0,187	0,885	0,837	-0,329	0,578	0,823	-0,326	0,746	0,685	1	0,405	0,402	0,505	0,827	0,340
TNOO	0,429	0,509	0,342	0,031	-0,025	0,694	0,253	-0,185	0,317	0,334	0,380	0,269	0,283	0,355	0,802	0,761	0,405	1	0,929	0,905	0,300	0,938
TLOC	0,331	0,430	0,393	0,121	0,061	0,715	0,229	-0,063	0,379	0,376	0,379	0,263	0,355	0,357	0,806	0,745	0,402	0,929	1	0,949	0,409	0,965
TRFC	0,501	0,576	0,540	0,077	0,018	0,678	0,339	0,004	0,500	0,533	0,276	0,172	0,506	0,251	0,799	0,759	0,505	0,905	0,949	1	0,574	0,969
TINVOK	0,753	0,761	0,758	0,069	0,029	0,197	0,686	0,096	0,840	0,853	-0,251	0,178	0,846	-0,253	0,558	0,514	0,827	0,300	0,409	0,574	1	0,406
TWMP	0,361	0,444	0,433	0,031	-0,028	0,694	0,186	-0,005	0,357	0,395	0,400	0,114	0,361	0,373	0,726	0,685	0,340	0,938	0,965	0,969	0,406	1

Tableau 6.2 : Matrice de corrélation (Spearman) de Healthwatcher Java

Variables	HDiff	HEff	FO	LCOM1	LCOM3	LOC	NOA	NOCC	RFC	WMPC1	CBO	DOIH	NOO	MTMOOP	TNofCalled	TNofNewData	TNofAssert	TNOO	TLOC	TRFC	TINVOK	TWMPG1
HDiff	1	0,706	0,600	-0,324	-0,173	0,720	-0,112	-0,148	0,597	0,621	0,595	0,091	0,112	0,510	0,537	0,495	-0,221	0,707	0,744	0,601	0,163	0,707
HEff	0,706	1	0,615	-0,153	0,085	0,970	0,073	-0,053	0,626	0,791	0,550	-0,087	0,290	0,425	0,780	0,636	-0,134	0,899	0,960	0,689	0,066	0,896
FO	0,600	0,615	1	-0,160	-0,081	0,642	-0,335	-0,165	0,856	0,377	0,960	0,445	0,005	0,897	0,507	0,478	-0,310	0,926	0,890	0,767	0,178	0,925
LCOM1	-0,324	-0,153	-0,160	1	0,527	0,003	0,879	0,714	-0,012	0,280	-0,217	-0,148	0,769	-0,269	0,026	-0,025	0,388	0,069	-0,039	0,149	0,186	0,073
LCOM3	-0,173	0,085	-0,081	0,527	1	0,195	0,785	0,218	0,229	0,340	-0,122	0,113	0,602	-0,145	0,252	0,195	0,511	0,181	0,128	0,536	0,760	0,197
LOC	0,720	0,970	0,642	0,003	0,195	1	0,211	-0,103	0,659	0,858	0,567	-0,075	0,422	0,434	0,852	0,718	-0,029	0,945	0,985	0,771	0,158	0,943
NOA	-0,112	0,073	-0,335	0,879	0,785	0,211	1	-0,003	-0,214	0,530	-0,443	-0,543	0,811	-0,533	0,384	0,245	0,666	0,212	0,139	0,366	0,398	0,219
NOCC	-0,148	-0,053	-0,165	0,714	0,218	-0,103	-0,003	1	-0,026	-0,100	-0,163	0,045	-0,056	-0,124	-0,174	-0,195	-0,133	0,017	-0,055	0,053	0,075	0,019
RFC	0,597	0,626	0,856	-0,012	0,229	0,659	-0,214	-0,026	1	0,371	0,901	0,642	-0,065	0,893	0,513	0,540	-0,247	0,957	0,952	0,851	0,293	0,958
WMPC1	0,621	0,791	0,377	0,280	0,340	0,858	0,530	-0,100	0,371	1	0,243	-0,423	0,779	0,069	0,822	0,611	0,293	0,954	0,946	0,856	0,306	0,956
CBO	0,595	0,550	0,960	-0,217	-0,122	0,567	-0,443	-0,163	0,901	0,243	1	0,621	-0,179	0,978	0,363	0,396	-0,444	0,949	0,932	0,778	0,166	0,947
DOIH	0,091	-0,087	0,445	-0,148	0,113	-0,075	-0,543	0,045	0,642	-0,423	0,621	1	-0,622	0,766	-0,209	0,049	-0,440	0,027	0,062	0,278	0,487	0,037
NOO	0,112	0,290	0,005	0,769	0,602	0,422	0,811	-0,056	-0,065	0,779	-0,179	-0,622	1	-0,340	0,574	0,352	0,613	0,612	0,507	0,680	0,441	0,619
MTMOOP	0,510	0,425	0,897	-0,269	-0,145	0,434	-0,533	-0,124	0,893	0,069	0,978	0,766	-0,340	1	0,230	0,323	-0,501	0,945	0,939	0,786	0,187	0,944
TNofCalled	0,537	0,780	0,507	0,026	0,252	0,852	0,384	-0,174	0,513	0,822	0,363	-0,209	0,574	0,230	1	0,915	0,421	0,879	0,913	0,779	0,263	0,879
TNofNewData	0,495	0,636	0,478	-0,025	0,195	0,718	0,245	-0,195	0,540	0,611	0,396	0,049	0,352	0,323	0,915	1	0,375	0,690	0,756	0,630	0,241	0,691
TNofAssert	-0,221	-0,134	-0,310	0,388	0,511	-0,029	0,666	-0,133	-0,247	0,293	-0,444	-0,440	0,613	-0,501	0,421	0,375	1	-0,063	-0,087	0,120	0,312	-0,056
TNOO	0,707	0,899	0,926	0,069	0,181	0,945	0,212	0,017	0,957	0,954	0,949	0,027	0,612	0,945	0,879	0,690	-0,063	1	0,981	0,857	0,244	1,000
TLOC	0,744	0,960	0,890	-0,039	0,128	0,985	0,139	-0,055	0,952	0,946	0,932	0,062	0,507	0,939	0,913	0,756	-0,087	0,981	1	0,819	0,199	0,980
TRFC	0,601	0,689	0,767	0,149	0,536	0,771	0,366	0,053	0,851	0,856	0,778	0,278	0,680	0,786	0,779	0,630	0,120	0,857	0,819	1	0,709	0,868
TINVOK	0,163	0,066	0,178	0,186	0,760	0,158	0,398	0,075	0,293	0,306	0,166	0,487	0,441	0,187	0,263	0,241	0,312	0,244	0,199	0,709	1	0,255
TWMPG1	0,707	0,896	0,925	0,073	0,197	0,943	0,219	0,019	0,958	0,956	0,947	0,037	0,619	0,944	0,879	0,691	-0,056	1,000	0,980	0,868	0,265	1

Tableau 6.3 : Matrice de corrélation (Pearson) de Healthwatcher AspectJ

Variables	HDifT	HEifT	FO	LCOM1	LCOM3	LOC	NOA	NOCC	RFC	WMPC1	CBO	DOIH	NOO	MTMOOP	TnofCalled	TnofNewData	TnofAssert	TNOO	TLOC	TRFC	TINVOK	TWMPCT
HDifT	1	0,901	0,618	-0,406	-0,429	0,547	-0,289	-0,378	0,629	0,186	0,700	0,404	-0,120	0,694	0,287	0,463	-0,293	0,576	0,629	0,608	0,445	0,576
HEifT	0,901	1	0,721	-0,123	-0,046	0,749	-0,274	-0,357	0,790	0,156	0,810	0,558	-0,132	0,796	0,335	0,525	-0,292	0,762	0,826	0,790	0,415	0,762
FO	0,618	0,721	1	-0,284	-0,341	0,454	-0,508	-0,102	0,816	-0,077	0,956	0,557	-0,211	0,923	0,261	0,361	-0,353	0,744	0,741	0,701	0,127	0,744
LCOM1	-0,406	-0,123	-0,284	1	0,764	0,261	0,666	0,375	0,188	0,466	-0,327	0,183	0,597	-0,368	0,309	0,073	0,705	0,376	0,291	0,397	0,301	0,376
LCOM3	-0,429	-0,046	-0,341	0,764	1	0,382	0,936	0,340	0,190	0,433	-0,359	0,231	0,587	-0,346	0,223	0,032	0,521	0,229	0,156	0,254	0,308	0,229
LOC	0,547	0,749	0,454	0,261	0,382	1	0,237	-0,125	0,607	0,606	0,484	0,206	0,399	0,424	0,651	0,594	0,270	0,919	0,919	0,920	0,192	0,919
NOA	-0,289	-0,274	-0,508	0,666	0,936	0,237	1	0,074	-0,356	0,669	-0,564	-0,591	0,826	-0,604	0,510	0,216	0,833	0,520	0,540	0,565	0,269	0,520
NOCC	-0,378	-0,357	-0,102	0,375	0,340	-0,125	0,074	1	-0,165	0,098	-0,152	-0,222	0,192	-0,143	-0,182	-0,279	0,019	0,289	0,044	0,255	-0,056	0,289
RFC	0,629	0,790	0,816	0,188	0,190	0,607	-0,356	-0,165	1	-0,017	0,878	0,755	-0,252	0,870	0,256	0,408	-0,402	0,752	0,859	0,738	0,133	0,752
WMPC1	0,186	0,156	-0,077	0,466	0,433	0,606	0,669	0,098	-0,017	1	-0,123	-0,479	0,884	-0,201	0,617	0,310	0,733	0,996	0,925	0,980	0,193	0,996
CBO	0,700	0,810	0,956	-0,327	-0,359	0,484	-0,564	-0,152	0,878	-0,123	1	0,671	-0,337	0,988	0,156	0,340	-0,486	0,744	0,741	0,701	0,127	0,744
DOIH	0,404	0,558	0,557	0,183	0,231	0,206	-0,591	-0,222	0,755	-0,479	0,671	1	-0,647	0,722	-0,090	0,254	-0,627	0,059	0,261	0,058	0,222	0,059
NOO	-0,120	-0,132	-0,211	0,597	0,587	0,399	0,826	0,192	-0,252	0,884	-0,337	-0,647	1	-0,411	0,654	0,311	0,875	0,940	0,860	0,937	0,178	0,940
MTMOOP	0,694	0,796	0,923	-0,368	-0,346	0,424	-0,604	-0,143	0,870	-0,201	0,988	0,722	-0,411	1	0,091	0,327	-0,551	0,611	0,662	0,579	0,135	0,611
TnofCalled	0,287	0,335	0,261	0,309	0,223	0,651	0,510	-0,182	0,256	0,617	0,156	-0,090	0,654	0,091	1	0,855	0,640	0,675	0,824	0,663	0,192	0,675
TnofNewData	0,463	0,525	0,361	0,073	0,032	0,594	0,216	-0,279	0,408	0,310	0,340	0,254	0,311	0,327	0,855	1	0,289	0,573	0,721	0,575	0,281	0,573
TnofAssert	-0,293	-0,292	-0,353	0,705	0,521	0,270	0,833	0,019	-0,402	0,733	-0,486	-0,627	0,875	-0,551	0,640	0,289	1	0,322	0,438	0,325	0,369	0,322
TNOO	0,576	0,762	0,744	0,376	0,229	0,919	0,520	0,289	0,752	0,996	0,744	0,059	0,940	0,611	0,675	0,573	0,322	1	0,934	0,986	0,192	1,000
TLOC	0,629	0,826	0,741	0,291	0,156	0,919	0,540	0,044	0,859	0,925	0,741	0,261	0,860	0,662	0,824	0,721	0,438	0,934	1	0,923	0,162	0,934
TRFC	0,608	0,790	0,701	0,397	0,254	0,920	0,565	0,255	0,738	0,980	0,701	0,058	0,937	0,570	0,663	0,575	0,325	0,986	0,923	1	0,281	0,986
TINVOK	0,445	0,415	0,127	0,301	0,308	0,192	0,269	-0,056	0,133	0,193	0,127	0,222	0,178	0,135	0,192	0,281	0,369	0,192	0,162	0,281	1	0,192
TWMPCT	0,576	0,762	0,744	0,376	0,229	0,919	0,520	0,289	0,752	0,996	0,744	0,059	0,940	0,611	0,675	0,573	0,322	1,000	0,934	0,986	0,192	1

Tableau 6.4 : Matrice de corrélation (Spearman) de Healthwatcher AspectJ

Pour ce système, nous avons analysé 35 classes qui ont été touchées (par des modifications) lors du passage de la version OO à la version OA. Les résultats des corrélations montrent que le coefficient de Pearson donne de meilleurs résultats. Dans la version Java de Healthwatcher, la métrique MTMOOP corrèle significativement avec les métriques TNOO, TLOC, TRFC et TWMP. Cependant, TNofAssert ne corrèle pas significativement avec le MTMOOP avec le coefficient de corrélation de Pearson. En calculant le coefficient de corrélation de Spearman, nous avons obtenu un coefficient de -0.326. Pour ce système, il n'existe pas de corrélation significative entre le MTMOOP et les métriques TNofCalled, TNofNewData et TINVOK.

Dans la version AspectJ de Healthwatcher, le modèle MTMOOP corrèle fortement avec les métriques TNofAssert, TNOO, TLOC, TRFC et TWMP que ce soit pour Pearson ou pour Spearman. Cependant, il n'y a pas de corrélation significative entre le MTMOOP et les métriques TNofCalled, TNofNewData et TINVOK.

Puisque nous avons vérifié la validité de nos données, on peut procéder à la comparaison de la testabilité (des classes) des deux versions de Healthwatcher (Java et AspectJ). Nous avons commencé par dessiner les courbes représentatives des données recueillies de chaque système. Du côté des classes logicielles, le MTMOOP évolue de cette façon :

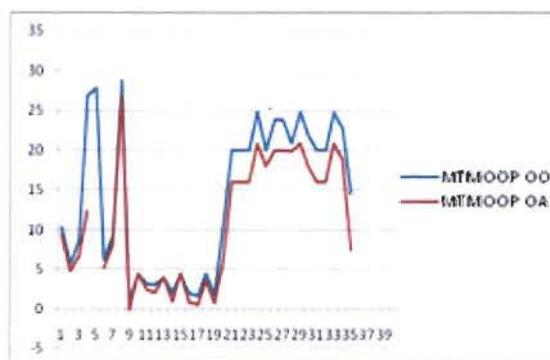


Figure 1.1 MTMOOP : HealthWatcher (OO vs OA)

Du côté des classes tests JUNIT, voici l'évolution de chaque métrique :

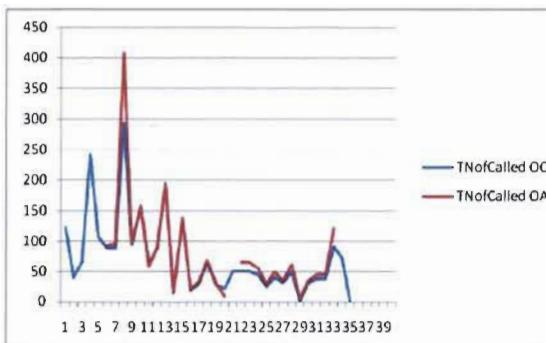


Figure 1.2 TNofCalled : HealthWatcher (OO vs OA)

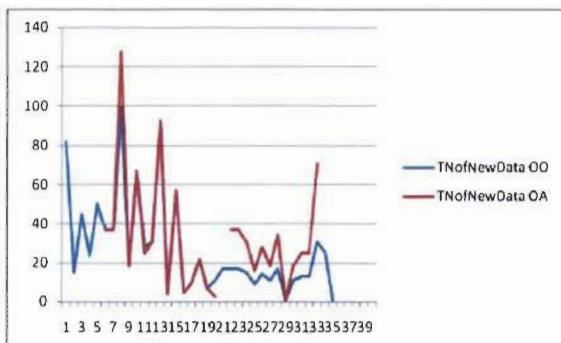


Figure 1.3 TNofNewData : HealthWatcher (OO vs OA)

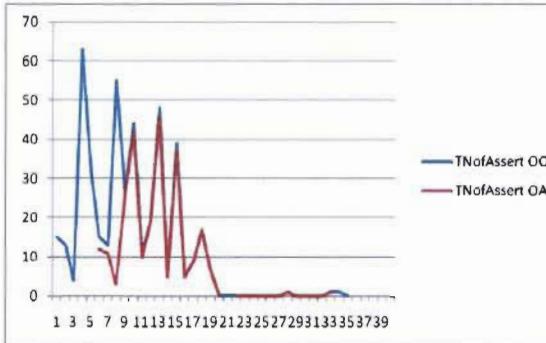


Figure 1.4 TNofAssert : HealthWatcher (OO vs OA)

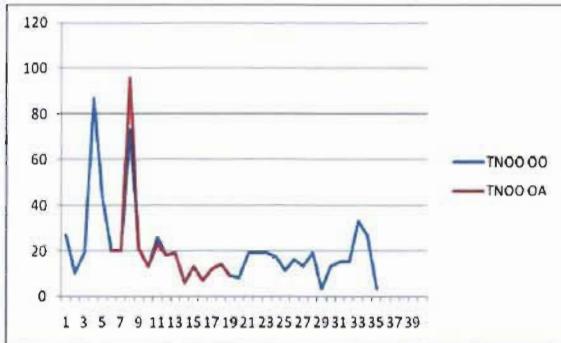


Figure 1.5 TNOO : HealthWatcher (OO vs OA)

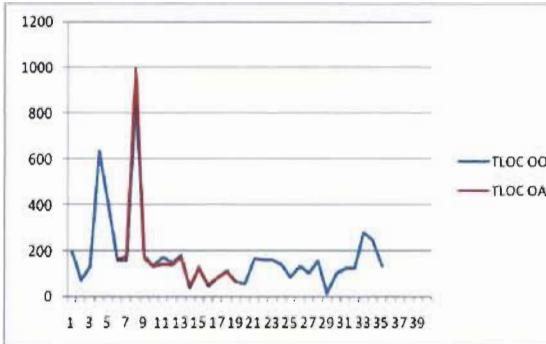


Figure 1.6 TLOC : HealthWatcher (OO vs OA)

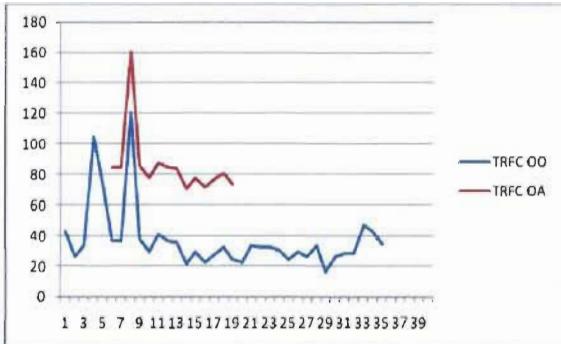


Figure 1.7 TRFC : HealthWatcher (OO vs OA)

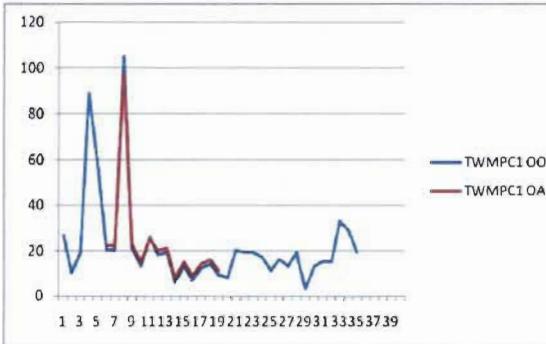


Figure 1.8 TWMPc1 : HealthWatcher (OO vs OA)

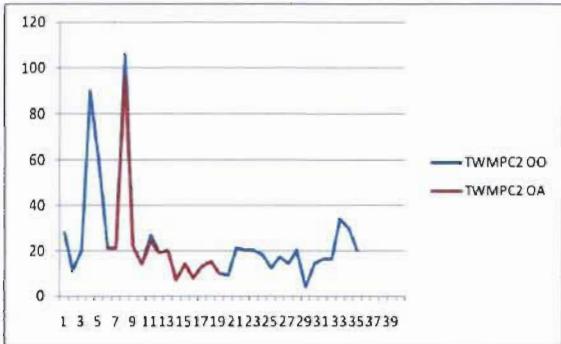


Figure 1.9 TWMPc2 : HealthWatcher (OO vs OA)

Pour déterminer s'il y a une amélioration ou non de la testabilité des classes OO après leur aspectualisation (refactoring aspect), nous allons d'abord commencer par analyser les courbes précédentes en sélectionnant deux classes à la fois du même système. Par exemple, en tenant compte des classes 1 et 2 de HealthWatcher OO, le modèle MTMOOP montre que la classe 2 est plus facile à tester que la classe 1, et cela est vérifié par toutes les métriques du côté des classes tests. Si on compare les classes 2 et 3, la prédiction du modèle MTMOOP est vérifiée dans 7 métriques de tests sur les 8. Il y a juste la métrique TNofAssert qui indique qu'il y a moins de « *assert* » dans la classe 3 par rapport à la classe 2, par contre le modèle MTMOOP indique que la classe 2 nécessite moins d'effort de test que la classe 3, et ainsi de suite pour les autres classes. À part quelques exceptions, les fréquences de toutes les courbes sont assez semblables, elles ont presque la même allure.

Du côté de la comparaison du code source des programmes OO vs OA, le modèle MTMOOP indique que le HealthWatcher OA est plus facile à tester que sa version OO. Comme on peut le voir clairement dans les graphiques du MTMOOP, les courbes rouges qui correspondent à la version OA de HealthWatcher, sont largement au-dessous des courbes bleues qui correspondent à la version OO du même système. D'après notre suite de métriques de testabilité et les graphiques correspondants, on remarque qu'il n'y a pas de grosses différences du côté complexité des classes tests. Il y a juste TRFC, où on voit que les valeurs obtenues dans la version OA sont plus élevées.

Par ailleurs, en effectuant notre test statistique t définit plus haut (voir les résultats dans le tableau 6), nous avons conclu globalement que les différences entre les valeurs des métriques respectives calculées sur les deux versions de HealthWatcher ne sont pas significatives en majorité.

Ce résultat statistique peut expliquer l'incohérence que nous avons remarquée entre la figure 1.1 et les figures 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8 et 1.9.

Variables	Valeur de T	Seuil de Student	#Classes
HDiff	-2,8331769	2,438	35
HEff	-0,51744239	2,438	35
LCOM1	0,48362235	2,539	19
LCOM3	1,28072157	2,539	19
LOC	2,06276764	2,438	35
NOA	0,89548132	2,438	35
NOCC	0	2,438	35
RFC	-3,68200788	2,438	35
WMPC1	-0,40060642	2,438	35
CBO	3,49236002	2,438	35
DOIH	-4,10362244	2,438	35
NOO	0,5289947	2,438	35
MTMOOP	1,83331738	2,438	35
TNofCalled	-0,41618922	2,457	30
TNofNewData	-1,75425633	2,457	30
TNofAssert	1,71559084	2,457	30
TNOO	0,72783538	2,567	17
TLOC	1,19879785	2,567	17
TRFC	-5,16056303	2,567	17
TINVOK	-8,36982481	2,567	17
TWMPC	1,14412881	2,567	17

Tableau 6.5 : Résultats du test statistique de Healthwatcher

- Projet 2 : Telestrada

Variables	HDiff	HEff	FO	LCOM1	LCOM3	LOC	NOA	NOCC	RFC	WMPC1	CBO	DOIH	NOO	MTMOOP	TnofCalled	TnofNewData	TnofAssert	TNOO	TLOC	TRFC	TINVOK	TWMPC1
HDiff	1	0,800	-0,068	0,576	0,583	0,833	0,652	0,453	0,612	0,832	-0,011	0,065	0,452	-0,030	0,889	0,638	0,724	0,891	0,895	0,888	0,103	0,891
HEff	0,800	1	0,319	0,337	0,039	0,944	0,337	0,098	0,809	0,774	0,387	0,180	0,423	0,374	0,911	0,792	0,848	0,845	0,879	0,841	-0,022	0,845
FO	-0,068	0,319	1	-0,390	-0,835	0,284	-0,431	-0,380	0,299	-0,034	0,935	-0,113	-0,108	0,930	0,143	0,419	0,272	0,000	0,071	-0,005	-0,246	0,000
LCOM1	0,576	0,337	-0,390	1	0,987	0,557	0,951	0,506	0,575	0,738	-0,350	0,814	0,949	-0,361	0,547	-0,061	0,383	0,687	0,636	0,691	0,506	0,687
LCOM3	0,583	0,039	-0,835	0,987	1	0,307	0,983	0,542	0,315	0,687	-0,841	0,678	0,960	-0,843	0,403	-0,394	0,139	0,643	0,550	0,651	0,542	0,643
LOC	0,833	0,944	0,284	0,557	0,307	1	0,540	0,248	0,897	0,889	0,379	0,355	0,647	0,365	0,947	0,683	0,862	0,930	0,945	0,928	0,156	0,930
NOA	0,652	0,337	-0,431	0,951	0,983	0,540	1	0,734	0,378	0,733	-0,404	0,295	0,738	-0,423	0,548	-0,030	0,158	0,677	0,631	0,682	0,382	0,677
NOCC	0,453	0,098	-0,380	0,506	0,542	0,248	0,734	1	0,064	0,560	-0,311	-0,105	0,455	-0,340	0,255	-0,128	0,492	0,370	0,330	0,382	0,681	0,370
RFC	0,612	0,809	0,299	0,575	0,315	0,897	0,378	0,064	1	0,835	0,479	0,674	0,765	0,477	0,814	0,558	0,744	0,829	0,830	0,831	0,314	0,829
WMPC1	0,832	0,774	-0,034	0,738	0,687	0,889	0,733	0,560	0,835	1	0,115	0,436	0,827	0,097	0,863	0,454	0,866	0,927	0,910	0,932	0,501	0,927
CBO	-0,011	0,387	0,935	-0,350	-0,841	0,379	-0,404	-0,311	0,479	0,115	1	0,104	0,048	0,999	0,226	0,458	0,391	0,112	0,167	0,111	0,016	0,112
DOIH	0,065	0,180	-0,113	0,814	0,678	0,355	0,295	-0,105	0,674	0,436	0,104	1	0,762	0,124	0,285	-0,060	0,141	0,398	0,350	0,405	0,423	0,398
NOO	0,452	0,423	-0,108	0,949	0,960	0,647	0,738	0,455	0,765	0,827	0,048	0,762	1	0,041	0,555	0,013	0,579	0,689	0,645	0,698	0,620	0,689
MTMOOP	-0,030	0,374	0,930	-0,361	-0,843	0,365	-0,423	-0,340	0,477	0,097	0,999	0,124	0,041	1	0,213	0,452	0,368	0,099	0,154	0,099	0,009	0,099
TnofCalled	0,889	0,911	0,143	0,547	0,403	0,947	0,548	0,255	0,814	0,863	0,226	0,285	0,555	0,213	1	0,793	0,761	0,978	0,992	0,975	0,098	0,978
TnofNewData	0,638	0,792	0,419	-0,061	-0,394	0,683	-0,030	-0,128	0,558	0,454	0,458	-0,060	0,013	0,452	0,793	1	0,527	0,665	0,718	0,657	-0,211	0,665
TnofAssert	0,724	0,848	0,272	0,383	0,139	0,862	0,458	0,492	0,744	0,866	0,391	0,141	0,579	0,368	0,761	0,527	1	0,753	0,769	0,758	0,408	0,753
TNOO	0,891	0,845	0,000	0,687	0,643	0,930	0,677	0,370	0,829	0,927	0,112	0,398	0,689	0,099	0,978	0,665	0,753	1	0,996	1,000	0,242	1,000
TLOC	0,895	0,879	0,071	0,636	0,550	0,945	0,631	0,330	0,830	0,910	0,167	0,350	0,645	0,154	0,992	0,718	0,769	0,996	1	0,995	0,186	0,996
TRFC	0,888	0,841	-0,005	0,691	0,651	0,928	0,682	0,382	0,831	0,932	0,111	0,405	0,698	0,099	0,975	0,657	0,758	1,000	0,995	1	0,261	1,000
TINVOK	0,103	-0,022	-0,246	0,506	0,542	0,156	0,382	0,681	0,314	0,501	0,016	0,423	0,620	0,009	0,098	-0,211	0,408	0,242	0,186	0,261	1	0,242
TWMPC1	0,891	0,845	0,000	0,687	0,643	0,930	0,677	0,370	0,829	0,927	0,112	0,398	0,689	0,099	0,978	0,665	0,753	1,000	0,996	1,000	0,242	1

Tableau 7.1 : Matrice de corrélation (Pearson) de Telestrada Java

Variables	HDiff	HEff	FO	LCOM1	LCOM3	LOC	NOA	NOCC	RFC	WMPC1	CBO	DOIH	NOO	MTMOOP	TnofCalled	TnofNewData	TnofAssert	TNOO	TLOC	TRFC	TINVOK	TWMPCL																			
HDiff	1	0,885	-0,162	0,122	0,224	0,717	0,633	0,374	0,614	0,672	-0,100	-0,091	0,279	-0,063	0,677	0,608	0,505	0,689	0,679	0,689	-0,046	0,689																			
HEff		0,885	1	-0,043	-0,014	0,335	0,921	0,457	0,309	0,798	0,751	0,115	0,182	0,399	0,167	0,819	0,652	0,618	0,825	0,810	0,825	0,136	0,825																		
FO			-0,162	-0,043	1	-0,269	-0,783	0,094	-0,507	-0,439	0,154	-0,087	0,922	-0,092	0,208	0,901	0,199	0,387	0,276	0,094	0,183	0,094	-0,207	0,094																	
LCOM1				0,122	-0,014	-0,269	1	1,000	0,053	0,626	0,393	0,198	0,425	-0,391	0,465	0,641	-0,374	0,089	-0,170	0,369	0,243	0,170	0,243	0,393	0,243																
LCOM3					0,224	0,335	-0,783	1,000	1	0,335	0,875	0,395	0,335	0,631	-0,803	0,791	0,894	-0,783	0,335	-0,112	0,224	0,671	0,335	0,671	0,395	0,671															
LOC						0,717	0,921	0,094	0,053	0,335	1	0,325	0,310	0,938	0,829	0,262	0,364	0,607	0,316	0,892	0,635	0,716	0,917	0,889	0,917	0,318	0,917														
NOA							0,633	0,457	-0,507	0,626	0,875	0,325	1	0,619	0,187	0,321	-0,606	-0,065	0,305	-0,568	0,286	0,242	0,264	0,330	0,325	0,330	0,000	0,330													
NOCC								0,374	0,309	-0,439	0,393	0,395	0,310	0,619	1	0,248	0,437	-0,437	-0,105	0,382	-0,434	0,248	-0,093	0,372	0,315	0,310	0,315	0,681	0,315												
RFC									0,614	0,798	0,154	0,198	0,335	0,938	0,187	0,248	1	0,915	0,297	0,500	0,735	0,346	0,870	0,533	0,797	0,942	0,885	0,942	0,409	0,942											
WMPC1										0,672	0,751	-0,087	0,425	0,631	0,829	0,321	0,437	0,915	1	0,035	0,435	0,717	0,078	0,801	0,415	0,734	0,907	0,830	0,907	0,504	0,907										
CBO											-0,100	0,115	0,922	-0,391	-0,803	0,262	-0,606	-0,437	0,297	0,035	1	0,000	0,184	0,993	0,292	0,405	0,293	0,197	0,262	0,197	-0,092	0,197									
DOIH												-0,091	0,182	-0,092	0,465	0,791	0,364	-0,065	-0,105	0,500	0,435	0,000	1	0,560	0,046	0,363	0,046	0,410	0,463	0,364	0,463	0,423	0,463								
NOO													0,279	0,399	0,208	0,641	0,894	0,607	0,305	0,382	0,735	0,717	0,184	0,993	0,292	0,405	0,293	0,197	0,262	0,197	-0,092	0,197									
MTMOOP														-0,063	0,167	0,901	-0,374	-0,783	0,316	-0,568	-0,434	0,346	0,078	0,993	0,046	0,204	1	0,357	0,466	0,301	0,261	0,323	0,261	-0,091	0,261						
TnofCalled															0,677	0,819	0,199	0,089	0,335	0,892	0,286	0,248	0,870	0,801	0,292	0,363	0,357	1	0,827	0,651	0,971	0,990	0,971	0,273	0,971						
TnofNewData																0,608	0,652	0,387	-0,170	-0,112	0,635	0,242	-0,093	0,533	0,415	0,405	0,046	0,264	0,466	0,827	1	0,315	0,702	0,780	0,702	-0,250	0,702				
TnofAssert																	0,505	0,618	0,276	0,369	0,224	0,716	0,264	0,372	0,797	0,734	0,293	0,410	0,871	0,301	0,651	1	0,728	0,710	0,728	0,455	0,728				
TNOO																	0,689	0,825	0,094	0,243	0,671	0,917	0,330	0,315	0,942	0,907	0,197	0,463	0,700	0,261	0,971	0,702	0,728	1	0,977	1,000					
TLOC																		0,679	0,810	0,183	0,170	0,335	0,889	0,325	0,310	0,885	0,830	0,262	0,364	0,664	0,323	0,990	0,780	0,710	0,977	1	0,977	0,319	0,977		
TRFC																		0,689	0,825	0,094	0,243	0,671	0,917	0,330	0,315	0,942	0,907	0,197	0,463	0,700	0,261	0,971	0,702	0,728	1	0,977	1,000				
TINVOK																			-0,046	0,136	-0,207	0,393	0,395	0,318	0,000	0,681	0,409	0,504	-0,092	0,423	0,514	-0,091	0,273	-0,250	0,455	0,370	0,370	1	0,370		
TWMPCL																				0,689	0,825	0,094	0,243	0,671	0,917	0,330	0,315	0,942	0,907	0,197	0,463	0,700	0,261	0,971	0,702	0,728	1	0,977	1,000	0,370	1

Tableau 7.2 : Matrice de corrélation (Spearman) de Telestrada Java

Variables	HDiff	HEff	FO	LCOM1	LCOM3	LOC	NOA	NOCC	RFC	WMPC1	CBO	DOIH	NOO	MTMOOP	TNofCalled	TNofNewData	TNofAssert	TNOO	TLOC	TRFC	TINVOK	TWMPC1
HDiff	1	0,834	0,022	0,790	0,693	0,865	0,744	0,546	0,655	0,880	0,013	0,151	0,597	-0,017	0,709	0,340	0,515	0,727	0,714	0,740	0,546	0,727
HEff	0,834	1	0,352	0,703	0,308	0,949	0,570	0,238	0,828	0,879	0,286	0,277	0,581	0,264	0,928	0,643	0,778	0,936	0,927	0,939	0,238	0,936
FO	0,022	0,352	1	-0,261	-0,781	0,249	-0,386	-0,330	0,334	0,125	0,854	-0,063	-0,049	0,843	0,527	0,820	0,786	0,436	0,497	0,424	-0,330	0,436
LCOM1	0,790	0,703	-0,261	1	0,911	0,837	0,873	0,344	0,835	0,870	-0,136	0,842	0,968	-0,153	0,629	0,006	0,268	0,740	0,692	0,750	0,344	0,740
LCOM3	0,693	0,308	-0,781	0,911	1	0,581	0,983	0,542	0,494	0,723	-0,674	0,678	0,960	-0,685	0,124	-0,629	-0,295	0,318	0,231	0,341	0,542	0,318
LOC	0,865	0,949	0,249	0,837	0,581	1	0,736	0,410	0,895	0,971	0,276	0,425	0,771	0,253	0,909	0,517	0,721	0,943	0,925	0,951	0,410	0,943
NOA	0,744	0,570	-0,386	0,873	0,983	0,736	1	0,722	0,522	0,787	-0,329	0,337	0,745	-0,350	0,479	-0,048	0,124	0,526	0,492	0,545	0,722	0,526
NOCC	0,546	0,238	-0,330	0,344	0,542	0,410	0,722	1	0,130	0,536	-0,251	-0,105	0,455	-0,288	0,121	-0,170	0,080	0,123	0,111	0,152	1,000	0,123
RFC	0,655	0,828	0,334	0,835	0,494	0,895	0,522	0,130	1	0,880	0,521	0,726	0,832	0,515	0,836	0,438	0,731	0,935	0,896	0,935	0,130	0,935
WMPC1	0,880	0,879	0,125	0,870	0,723	0,971	0,787	0,536	0,880	1	0,238	0,469	0,858	0,213	0,807	0,344	0,659	0,872	0,843	0,884	0,536	0,872
CBO	0,013	0,286	0,854	-0,136	-0,674	0,276	-0,329	-0,251	0,521	0,238	1	0,322	0,224	0,999	0,429	0,550	0,775	0,448	0,461	0,438	-0,251	0,448
DOIH	0,151	0,277	-0,063	0,842	0,678	0,425	0,337	-0,105	0,726	0,469	0,322	1	0,762	0,345	0,321	-0,125	0,198	0,503	0,429	0,498	-0,105	0,503
NOO	0,597	0,581	-0,049	0,968	0,960	0,771	0,745	0,455	0,832	0,858	0,224	0,762	1	0,213	0,585	0,029	0,419	0,709	0,665	0,720	0,455	0,709
MTMOOP	-0,017	0,264	0,843	-0,153	-0,685	0,253	-0,350	-0,288	0,515	0,213	0,999	0,345	0,213	1	0,409	0,532	0,753	0,433	0,444	0,423	-0,288	0,433
TNofCalled	0,709	0,928	0,527	0,629	0,124	0,909	0,479	0,121	0,836	0,807	0,429	0,321	0,585	0,409	1	0,795	0,807	0,973	0,991	0,972	0,121	0,973
TNofNewData	0,340	0,643	0,820	0,006	-0,629	0,517	-0,048	-0,170	0,438	0,344	0,550	-0,125	0,029	0,532	0,795	1	0,758	0,668	0,732	0,661	-0,170	0,668
TNofAssert	0,515	0,778	0,786	0,268	-0,295	0,721	0,124	0,080	0,731	0,659	0,775	0,198	0,419	0,753	0,807	0,758	1	0,789	0,807	0,788	0,080	0,789
TNOO	0,727	0,936	0,436	0,740	0,318	0,943	0,526	0,123	0,935	0,872	0,448	0,503	0,709	0,433	0,973	0,668	0,789	1	0,993	1,000	0,123	1,000
TLOC	0,714	0,927	0,497	0,692	0,231	0,925	0,492	0,111	0,896	0,843	0,461	0,429	0,665	0,444	0,991	0,732	0,807	0,993	1	0,992	0,111	0,993
TRFC	0,740	0,939	0,424	0,750	0,341	0,951	0,545	0,152	0,935	0,884	0,438	0,498	0,720	0,423	0,972	0,661	0,788	1,000	0,992	1	0,152	1,000
TINVOK	0,546	0,238	-0,330	0,344	0,542	0,410	0,722	1,000	0,130	0,536	-0,251	-0,105	0,455	-0,288	0,121	-0,170	0,080	0,123	0,111	0,152	1	0,123
TWMPC1	0,727	0,936	0,436	0,740	0,318	0,943	0,526	0,123	0,935	0,872	0,448	0,503	0,709	0,433	0,973	0,668	0,789	1,000	0,993	1,000	0,123	1

Tableau 7.3 : Matrice de corrélation (Pearson) de Telestrada AspectJ

Variables	HDIIT	HEIT	FO	LCOM1	LCOM3	LOC	NOA	NOCC	RFC	WMPCI	CBO	DOIH	NOO	MTMOOP	TNofCalled	TNofNewData	TNofAsset	TNOO	TLOC	TRFC	TINVOK	TWMPCT
HDIIT	1	0,808	-0,213	0,956	0,648	0,623	0,455	0,448	0,629	0,758	-0,076	0,047	0,433	-0,041	0,461	0,213	0,428	0,515	0,496	0,515	0,448	0,515
HEIT	0,808	1	0,083	0,878	0,335	0,919	0,525	0,309	0,916	0,869	0,229	0,363	0,546	0,278	0,761	0,397	0,622	0,794	0,767	0,794	0,309	0,794
FO	-0,213	0,083	1	-0,454	-0,783	0,257	-0,360	-0,437	0,217	0,099	0,937	-0,046	0,184	0,923	0,310	0,441	0,432	0,289	0,288	0,289	-0,437	0,289
LCOM1	0,956	0,878	-0,454	1	0,918	0,756	0,751	0,423	0,900	0,951	-0,136	0,592	0,753	-0,171	0,561	0,122	0,454	0,683	0,683	0,683	0,423	0,683
LCOM3	0,648	0,335	-0,783	0,918	1	0,335	0,875	0,395	0,671	0,860	-0,783	0,791	0,894	-0,783	0,112	-0,447	-0,229	0,335	0,335	0,335	0,395	0,335
LOC	0,623	0,919	0,257	0,756	0,335	1	0,540	0,311	0,982	0,884	0,364	0,456	0,711	0,400	0,867	0,438	0,763	0,903	0,885	0,903	0,311	0,903
NOA	0,455	0,525	-0,360	0,751	0,875	0,540	1	0,473	0,460	0,346	-0,421	0,099	0,155	-0,369	0,431	0,175	0,172	0,362	0,374	0,362	0,473	0,362
NOCC	0,448	0,309	-0,437	0,423	0,395	0,311	0,473	1	0,249	0,407	-0,406	-0,105	0,382	-0,434	0,247	-0,248	0,280	0,311	0,310	0,311	1,000	0,311
RFC	0,629	0,916	0,217	0,900	0,671	0,982	0,460	0,249	1	0,916	0,375	0,548	0,724	0,416	0,808	0,366	0,743	0,872	0,843	0,872	0,249	0,872
WMPCI	0,758	0,869	0,099	0,951	0,860	0,884	0,346	0,407	0,916	1	0,298	0,482	0,819	0,322	0,659	0,161	0,771	0,777	0,740	0,777	0,407	0,777
CBO	-0,076	0,229	0,937	-0,136	-0,783	0,364	-0,421	-0,406	0,375	0,298	1	0,115	0,312	0,993	0,323	0,344	0,553	0,354	0,337	0,354	-0,406	0,354
DOIH	0,047	0,363	-0,046	0,592	0,791	0,456	0,099	-0,105	0,548	0,482	0,115	1	0,560	0,137	0,363	-0,045	0,411	0,456	0,410	0,456	-0,105	0,456
NOO	0,433	0,546	0,184	0,753	0,894	0,711	0,155	0,382	0,724	0,819	0,312	0,560	1	0,278	0,671	0,198	0,847	0,783	0,767	0,783	0,382	0,783
MTMOOP	-0,041	0,278	0,923	-0,171	-0,783	0,400	-0,369	-0,434	0,416	0,322	0,993	0,137	0,278	1	0,328	0,346	0,534	0,356	0,336	0,356	-0,434	0,356
TNofCalled	0,461	0,761	0,310	0,561	0,112	0,867	0,431	0,247	0,808	0,659	0,323	0,363	0,671	0,328	1	0,726	0,682	0,969	0,982	0,969	0,247	0,969
TNofNewData	0,213	0,397	0,441	0,122	-0,447	0,438	0,175	-0,248	0,366	0,161	0,344	-0,045	0,198	0,346	0,726	1	0,236	0,585	0,631	0,585	-0,248	0,585
TNofAsset	0,428	0,622	0,432	0,454	-0,229	0,763	0,172	0,280	0,743	0,771	0,553	0,411	0,847	0,534	0,682	0,236	1	0,755	0,740	0,755	0,280	0,755
TNOO	0,515	0,794	0,289	0,683	0,335	0,903	0,362	0,311	0,872	0,777	0,354	0,456	0,783	0,356	0,969	0,585	0,755	1	0,995	1,000	0,311	1,000
TLOC	0,496	0,767	0,288	0,683	0,335	0,885	0,374	0,310	0,843	0,740	0,337	0,410	0,767	0,336	0,982	0,631	0,740	0,995	1	0,995	0,310	0,995
TRFC	0,515	0,794	0,289	0,683	0,335	0,903	0,362	0,311	0,872	0,777	0,354	0,456	0,783	0,356	0,969	0,585	0,755	1,000	0,995	1	0,311	1,000
TINVOK	0,448	0,309	-0,437	0,423	0,395	0,311	0,473	1,000	0,249	0,407	-0,406	-0,105	0,382	-0,434	0,247	-0,248	0,280	0,311	0,310	0,311	1	0,311
TWMPCT	0,515	0,794	0,289	0,683	0,335	0,903	0,362	0,311	0,872	0,777	0,354	0,456	0,783	0,356	0,969	0,585	0,755	1,000	0,995	1,000	0,311	1

Tableau 7.4 : Matrice de corrélation (Spearman) de Telestrada AspectJ

Vu le nombre restreint des classes sélectionnées dans ce projet (15 classes aspectualisées uniquement), nous ne pouvons pas nous fier sur le coefficient de corrélation r_s . Cependant, et malgré tout, dans la version OA de Telestrada, il y a une corrélation significative entre le modèle MTMOOP et la métrique TNofAssert et entre le modèle MTMOOP et la métrique TNofCalled du côté de Pearson.

Remarque : les corrélations calculées sur les deux versions (OO et OA) du système Telestrada au complet sont très fortes (voir annexe).

Maintenant, nous allons passer aux courbes représentatives des données recueillies de chaque système. Du côté des classes logicielles, le modèle MTMOOP évolue de cette façon :

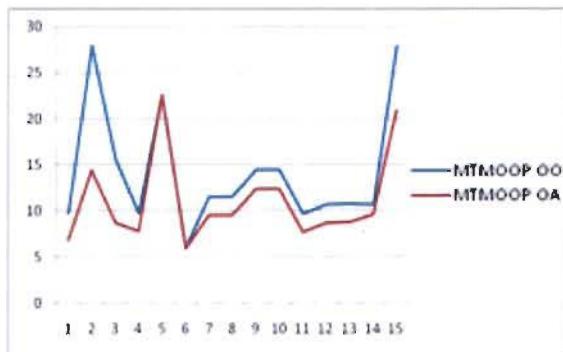


Figure 2.1 MTMOOP : Telestrada (OO vs OA)

Du côté des classes tests JUNIT, voici l'évolution de chaque métrique :

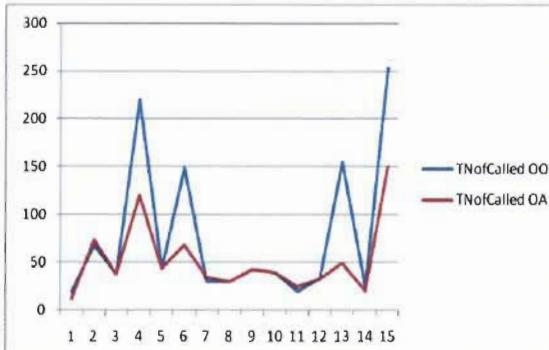


Figure 2.2 TNofCalled : Telestrada (OO vs OA)

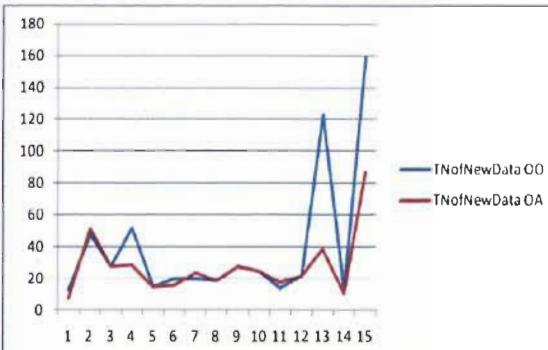


Figure 2.3 TNofNewData : Telestrada (OO vs OA)

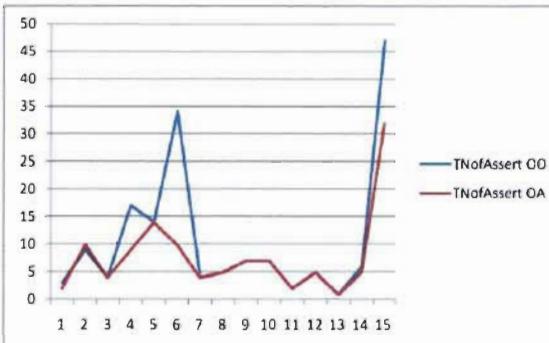


Figure 2.4 TNofAssert : Telestrada (OO vs OA)

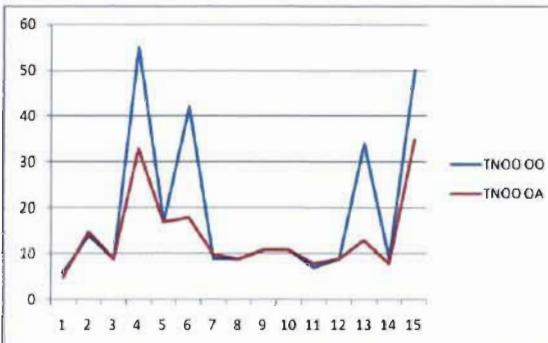


Figure 2.5 TNOO : Telestrada (OO vs OA)

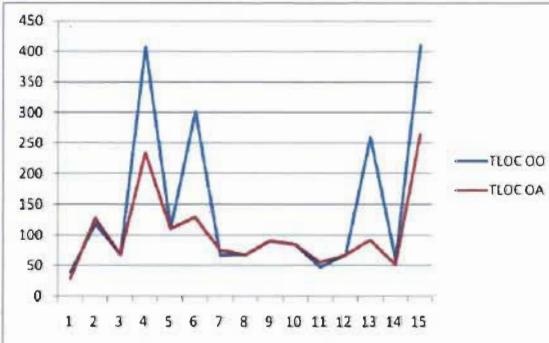


Figure 2.6 TLOC : Telestrada (OO vs OA)

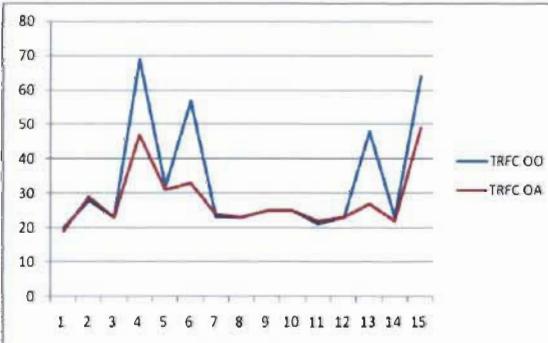


Figure 2.7 TRFC : Telestrada (OO vs OA)

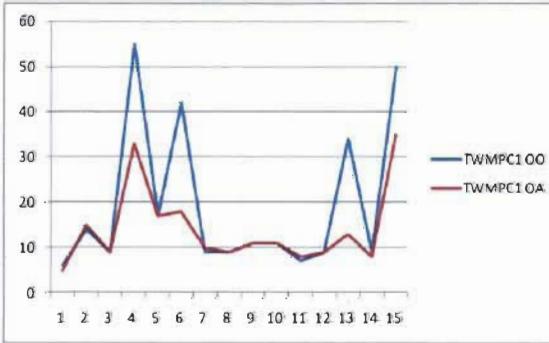


Figure 2.8 TWMPCT1: Telestrada (OO vs OA)

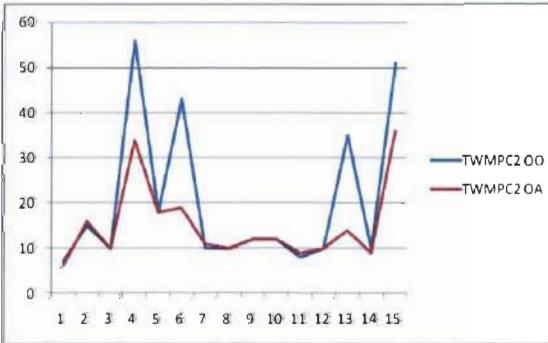


Figure 2.9 TWMPCT2 : Telestrada (OO vs OA)

Sur les 15 classes sélectionnées, le modèle MTMOOP indique bien la testabilité de 12 classes, comme on peut le voir facilement grâce aux courbes. Quand le modèle MTMOOP indique une baisse, la plupart des métriques suivent la même tendance (exp : classe 14) et quand il y a hausse (classes 2,15) elle est confirmée par les autres métriques. Par contre, nous obtenons le contraire de ce que le modèle MTMOOP prévoit pour les classes 4, 5, et 6.

Dans le contexte de notre comparaison entre les versions OO vs OA, la courbe des valeurs du modèle MTMOOP mesurée sur les classes aspectualisées, est bien en dessous de la courbe de l'équivalent de ces classes dans la version Java. Cela est fortement confirmé par les courbes de notre suite de métriques de testabilité calculées à partir des classes tests JUNIT correspondantes.

Par ailleurs, en effectuant notre test statistique t (voir les résultats dans le tableau 7), nous avons conclu globalement que les différences entre les valeurs des métriques respectives calculées sur les deux versions de Telestrada (OO et OA) sont :

- Positives pour toutes les métriques logicielles à part pour LCOM1 et WMPC.
- Significatives pour la plupart des métriques logicielles et le modèle MTMOOP.
- Positives, mais non significatives du côté des métriques de testabilité même si les résultats sont très proches du seuil de signification.

Variables	Valeur de T	Seuil de Student	#Classes
HDiff	3,16699396	2,602	15
HEff	1,87531095	2,602	15
LCOM1	-1,88120572	2,896	8
LCOM3	0	3,365	5
LOC	3,65619845	2,602	15
NOA	3,05505046	2,602	15
NOCC	0	2,602	15
RFC	4,20794289	2,602	15
WMPC1	-0,64168895	2,602	15
CBO	3,4335803	2,602	15
DOIH	0	2,602	15
NOO	0	2,602	15
MTMOOP	3,4335803	2,602	15
TNofCalled	2,19564633	2,602	15
TNofNewData	1,65859192	2,602	15
TNofAssert	1,73495775	2,602	15
TNOO	2,175722	2,602	15
TLOC	2,19711362	2,602	15
TRFC	2,211031	2,602	15
TINVOK	1	2,602	15
TWMPC	2,175722	2,602	15

Tableau 7.5 : Résultats du test statistique de Telestrada

- Projet 3 : Petstore

Variables	HDifF	HEff	FO	LCOMI	LCOM3	LOC	NOA	NOCC	RFC	WMPC1	CBO	DOIH	NOO	MTMOOP	TNofCalled	TNofNewData	TNofAsset	TNOO	TLOC	TRFC	TINVOK	TWMPG1
HDifF	1	0,864	0,742	0,114	-0,040	0,837	0,266	-0,209	0,695	0,541	0,684	-0,228	0,368	0,684	0,608	0,515	0,619	0,610	0,625	0,604	0,159	0,610
HEff	0,864	1	0,775	0,262	0,035	0,886	0,002	-0,145	0,632	0,625	0,681	-0,285	0,405	0,673	0,648	0,520	0,784	0,776	0,783	0,767	0,163	0,776
FO	0,742	0,775	1	0,452	0,422	0,846	0,233	-0,304	0,769	0,717	0,971	-0,181	0,661	0,964	0,748	0,729	0,725	0,763	0,768	0,765	0,484	0,763
LCOMI	0,114	0,262	0,452	1	0,177	0,547	-0,167	-0,068	0,676	0,793	0,493	-0,215	0,934	0,412	0,531	0,617	0,572	0,640	0,638	0,654	0,759	0,640
LCOM3	-0,040	0,035	0,422	0,177	1	0,180	0,126	-0,223	0,090	0,078	0,384	-0,200	0,247	0,376	0,227	0,233	0,180	0,182	0,175	0,186	0,210	0,182
LOC	0,837	0,886	0,846	0,547	0,180	1	0,096	-0,165	0,812	0,797	0,769	-0,268	0,657	0,741	0,655	0,609	0,798	0,800	0,796	0,799	0,413	0,800
NOA	0,266	0,002	0,233	-0,167	0,126	0,096	1	-0,050	0,312	-0,195	0,308	0,560	0,021	0,364	0,117	0,182	-0,157	-0,110	-0,071	-0,103	0,163	-0,110
NOCC	-0,209	-0,145	-0,304	-0,068	-0,223	-0,165	-0,050	1	-0,002	-0,100	-0,320	0,419	-0,067	-0,313	-0,175	-0,191	-0,177	-0,182	-0,174	-0,179	-0,034	-0,182
RFC	0,695	0,632	0,769	0,676	0,090	0,812	0,312	-0,002	1	0,785	0,774	0,038	0,830	0,746	0,620	0,638	0,586	0,642	0,656	0,648	0,520	0,642
WMPC1	0,541	0,625	0,717	0,793	0,078	0,797	-0,195	-0,100	0,785	1	0,682	-0,238	0,836	0,628	0,639	0,648	0,792	0,817	0,798	0,820	0,516	0,817
CBO	0,684	0,681	0,971	0,493	0,384	0,769	0,308	-0,320	0,774	0,682	1	-0,106	0,707	0,994	0,778	0,791	0,686	0,745	0,755	0,752	0,591	0,745
DOIH	-0,228	-0,285	-0,181	-0,215	-0,200	-0,268	0,560	0,419	0,038	-0,238	-0,106	1	-0,192	-0,033	-0,307	-0,252	-0,315	-0,284	-0,287	-0,282	-0,080	-0,284
NOO	0,368	0,405	0,661	0,934	0,247	0,657	0,021	-0,067	0,830	0,836	0,707	-0,192	1	0,640	0,683	0,753	0,614	0,694	0,700	0,708	0,787	0,694
MTMOOP	0,684	0,673	0,964	0,412	0,376	0,741	0,364	-0,313	0,746	0,628	0,994	-0,033	0,640	1	0,745	0,755	0,653	0,710	0,720	0,715	0,545	0,710
TNofCalled	0,608	0,648	0,748	0,531	0,227	0,655	0,117	-0,175	0,620	0,639	0,778	-0,307	0,683	0,745	1	0,963	0,816	0,855	0,892	0,860	0,607	0,855
TNofNewData	0,515	0,520	0,729	0,617	0,233	0,609	0,182	-0,191	0,638	0,648	0,791	-0,252	0,753	0,755	0,963	1	0,744	0,804	0,835	0,814	0,751	0,804
TNofAsset	0,619	0,784	0,725	0,572	0,180	0,798	-0,157	-0,177	0,586	0,792	0,686	-0,315	0,614	0,653	0,816	0,744	1	0,987	0,978	0,984	0,438	0,987
TNOO	0,610	0,776	0,763	0,640	0,182	0,800	-0,110	-0,182	0,642	0,817	0,745	-0,284	0,694	0,710	0,855	0,804	0,987	1	0,995	1,000	0,532	1,000
TLOC	0,625	0,783	0,768	0,638	0,175	0,796	-0,071	-0,174	0,656	0,798	0,755	-0,287	0,700	0,720	0,892	0,835	0,978	0,995	1	0,994	0,535	0,995
TRFC	0,604	0,767	0,765	0,654	0,186	0,799	-0,103	-0,179	0,648	0,820	0,752	-0,282	0,708	0,715	0,860	0,814	0,984	1,000	0,994	1	0,556	1,000
TINVOK	0,159	0,163	0,484	0,759	0,210	0,413	0,163	-0,034	0,520	0,516	0,591	-0,080	0,787	0,545	0,607	0,751	0,438	0,532	0,535	0,556	1	0,532
TWMPG1	0,610	0,776	0,763	0,640	0,182	0,800	-0,110	-0,182	0,642	0,817	0,745	-0,284	0,694	0,710	0,855	0,804	0,987	1,000	0,995	1,000	0,532	1

TABLEAU 8.1 : MATRICE DE CORRÉLATION (PEARSON) DE PETSTORE JAVA

Variables	HDiff	HEff	FO	LCOM1	LCOM3	LOC	NOA	NOCC	RFC	WMPCI	CBO	DOIH	NOO	MTMOOP	TNofCalled	TNofNewData	TNofAssert	TNOO	TLOC	TRFC	TINVOK	TWMPG1
HDiff	1	0,943	0,731	0,525	0,084	0,905	0,287	-0,077	0,771	0,797	0,661	-0,236	0,678	0,608	0,576	0,541	0,557	0,584	0,567	0,584	0,215	0,584
HEff	0,943	1	0,863	0,563	0,249	0,982	0,201	-0,113	0,776	0,856	0,756	-0,247	0,714	0,706	0,654	0,615	0,661	0,679	0,669	0,675	0,258	0,679
FO	0,731	0,863	1	0,537	0,445	0,868	0,238	-0,312	0,752	0,773	0,934	-0,205	0,747	0,904	0,732	0,711	0,733	0,734	0,739	0,732	0,360	0,734
LCOM1	0,525	0,563	0,537	1	0,221	0,585	-0,033	0,182	0,668	0,661	0,474	-0,257	0,884	0,423	0,412	0,374	0,360	0,378	0,371	0,387	0,414	0,378
LCOM3	0,084	0,249	0,445	0,221	1	0,232	0,079	-0,372	0,019	0,122	0,302	-0,363	0,177	0,258	0,270	0,179	0,344	0,317	0,314	0,313	0,039	0,317
LOC	0,905	0,982	0,868	0,585	0,232	1	0,189	-0,080	0,790	0,867	0,768	-0,192	0,721	0,730	0,685	0,649	0,688	0,700	0,698	0,697	0,291	0,700
NOA	0,287	0,201	0,238	-0,033	0,079	0,189	1	0,065	0,478	-0,038	0,381	0,517	0,166	0,368	0,049	0,202	-0,031	0,039	0,047	0,029	0,288	0,039
NOCC	-0,077	-0,113	-0,312	0,182	-0,372	-0,080	0,065	1	0,088	-0,031	-0,331	0,472	0,022	-0,330	-0,342	-0,311	-0,384	-0,342	-0,342	-0,342	-0,121	-0,342
RFC	0,771	0,776	0,752	0,668	0,019	0,790	0,478	0,088	1	0,669	0,798	0,136	0,828	0,779	0,529	0,579	0,490	0,529	0,531	0,527	0,401	0,529
WMPCI	0,797	0,856	0,773	0,661	0,122	0,867	-0,038	-0,031	0,669	1	0,655	-0,241	0,808	0,636	0,688	0,599	0,671	0,680	0,665	0,684	0,255	0,680
CBO	0,661	0,756	0,934	0,474	0,302	0,768	0,381	-0,331	0,798	0,655	1	-0,074	0,720	0,987	0,785	0,820	0,760	0,770	0,784	0,768	0,495	0,770
DOIH	-0,236	-0,247	-0,205	-0,257	-0,363	-0,192	0,517	0,472	0,136	-0,241	-0,074	1	-0,161	0,000	-0,222	-0,105	-0,253	-0,198	-0,185	-0,210	0,076	-0,198
NOO	0,678	0,714	0,747	0,884	0,177	0,721	0,166	0,022	0,828	0,808	0,720	-0,161	1	0,674	0,604	0,569	0,550	0,580	0,569	0,590	0,471	0,580
MTMOOP	0,608	0,706	0,904	0,423	0,258	0,730	0,368	-0,330	0,779	0,636	0,987	0,000	0,674	1	0,784	0,823	0,759	0,763	0,782	0,759	0,476	0,763
TNofCalled	0,576	0,654	0,732	0,412	0,270	0,685	0,049	-0,342	0,529	0,688	0,785	-0,222	0,604	0,784	1	0,957	0,974	0,980	0,985	0,983	0,611	0,980
TNofNewData	0,541	0,615	0,711	0,374	0,179	0,649	0,202	-0,311	0,579	0,599	0,820	-0,105	0,569	0,823	0,957	1	0,927	0,938	0,951	0,937	0,685	0,938
TNofAssert	0,557	0,661	0,733	0,360	0,344	0,688	-0,031	-0,384	0,490	0,671	0,760	-0,253	0,550	0,759	0,974	0,927	1	0,980	0,985	0,980	0,535	0,980
TNOO	0,584	0,679	0,734	0,378	0,317	0,700	0,039	-0,342	0,529	0,680	0,770	-0,198	0,580	0,763	0,980	0,938	0,980	1	0,996	0,999	0,589	1,000
TLOC	0,567	0,669	0,739	0,371	0,314	0,698	0,047	-0,342	0,531	0,665	0,784	-0,185	0,569	0,782	0,985	0,951	0,985	0,996	1	0,995	0,600	0,996
TRFC	0,584	0,675	0,732	0,387	0,313	0,697	0,029	-0,342	0,527	0,684	0,768	-0,210	0,590	0,759	0,983	0,937	0,980	0,999	0,995	1	0,596	0,999
TINVOK	0,215	0,258	0,360	0,414	0,039	0,291	0,288	-0,121	0,401	0,255	0,495	0,076	0,471	0,476	0,611	0,685	0,535	0,589	0,600	0,596	1	0,589
TWMPG1	0,584	0,679	0,734	0,378	0,317	0,700	0,039	-0,342	0,529	0,680	0,770	-0,198	0,580	0,763	0,980	0,938	0,980	1,000	0,996	0,999	0,589	1

TABLEAU 8.2 : MATRICE DE CORRÉLATION (SPEARMAN) DE PETSTORE JAVA

Variables	HDiff	HEst	FO	LCOM1	LCOM3	LOC	NOA	NOCC	RFC	WMPCI	CBO	DOIH	NOO	MTMOOP	TNofCalled	TNofNewData	TNofAsset	TNOO	TLOC	TRFC	TINVOK	TWMPCI
HDiff	1	0,872	0,741	0,108	0,196	0,848	0,342	-0,256	0,713	0,548	0,688	-0,221	0,364	0,686	0,276	0,385	0,301	0,289	0,309	0,309	0,326	0,289
HEst	0,872	1	0,750	0,240	0,018	0,875	0,074	-0,131	0,640	0,607	0,672	-0,289	0,378	0,663	0,569	0,455	0,578	0,576	0,572	0,570	0,215	0,576
FO	0,741	0,750	1	0,456	0,369	0,824	0,313	-0,246	0,768	0,681	0,982	-0,165	0,666	0,972	0,615	0,700	0,611	0,636	0,658	0,661	0,540	0,636
LCOM1	0,108	0,240	0,456	1	-0,165	0,521	-0,120	-0,061	0,651	0,801	0,464	-0,205	0,942	0,370	0,969	0,707	0,932	0,926	0,935	0,920	0,354	0,926
LCOM3	0,196	0,018	0,369	-0,165	1	0,140	0,348	-0,084	0,052	-0,060	0,345	-0,056	-0,015	0,374	-0,110	0,112	-0,046	-0,043	-0,041	-0,014	0,259	-0,042
LOC	0,848	0,875	0,824	0,521	0,140	1	0,206	-0,161	0,809	0,766	0,751	-0,276	0,633	0,719	0,639	0,516	0,662	0,664	0,652	0,659	0,264	0,664
NOA	0,342	0,074	0,313	-0,120	0,348	0,206	1	-0,101	0,384	-0,138	0,355	0,394	0,105	0,397	-0,019	0,224	-0,127	-0,105	-0,031	-0,055	0,416	-0,104
NOCC	-0,256	-0,131	-0,246	-0,061	-0,084	-0,161	-0,101	1	-0,097	-0,130	-0,254	0,127	-0,080	-0,244	-0,086	-0,120	-0,105	-0,116	-0,113	-0,108	0,023	-0,116
RFC	0,713	0,640	0,768	0,651	0,052	0,809	0,384	-0,097	1	0,769	0,750	0,055	0,805	0,718	0,644	0,565	0,601	0,602	0,627	0,612	0,378	0,602
WMPCI	0,548	0,607	0,681	0,801	-0,060	0,766	-0,138	-0,130	0,769	1	0,645	-0,243	0,847	0,579	0,835	0,643	0,878	0,879	0,858	0,863	0,261	0,879
CBO	0,688	0,672	0,982	0,464	0,345	0,751	0,355	-0,254	0,750	0,645	1	-0,099	0,685	0,993	0,612	0,736	0,601	0,630	0,659	0,664	0,612	0,631
DOIH	-0,221	-0,289	-0,165	-0,205	-0,056	-0,276	0,394	0,327	0,055	-0,243	-0,099	1	-0,170	-0,021	-0,224	-0,247	-0,248	-0,235	-0,238	-0,221	0,021	-0,235
NOO	0,364	0,378	0,666	0,942	-0,015	0,633	0,105	-0,080	0,805	0,847	0,685	-0,170	1	0,609	0,925	0,813	0,891	0,891	0,916	0,901	0,522	0,891
MTMOOP	0,686	0,663	0,972	0,370	0,374	0,719	0,397	-0,244	0,718	0,579	0,993	-0,021	0,609	1	0,529	0,678	0,519	0,553	0,581	0,589	0,601	0,553
TNofCalled	0,276	0,569	0,615	0,969	-0,110	0,639	-0,019	-0,086	0,644	0,835	0,612	-0,224	0,925	0,529	1	0,834	0,970	0,960	0,983	0,967	0,527	0,960
TNofNewData	0,385	0,455	0,700	0,707	0,112	0,516	0,224	-0,120	0,565	0,643	0,736	-0,247	0,813	0,678	0,834	1	0,806	0,798	0,867	0,842	0,792	0,799
TNofAsset	0,301	0,578	0,611	0,932	-0,046	0,662	-0,127	-0,105	0,601	0,878	0,601	-0,248	0,891	0,519	0,970	0,806	1	0,989	0,987	0,991	0,488	0,989
TNOO	0,389	0,576	0,636	0,926	-0,043	0,664	-0,105	-0,116	0,602	0,879	0,630	-0,235	0,891	0,553	0,960	0,798	0,989	1	0,988	0,995	0,433	1,000
TLOC	0,309	0,572	0,658	0,935	-0,041	0,652	-0,031	-0,113	0,627	0,858	0,659	-0,238	0,916	0,581	0,983	0,867	0,987	0,988	1	0,995	0,534	0,988
TRFC	0,309	0,570	0,661	0,920	-0,014	0,659	-0,055	-0,108	0,612	0,863	0,664	-0,221	0,901	0,589	0,967	0,842	0,991	0,995	0,995	1	0,518	0,996
TINVOK	0,326	0,215	0,540	0,354	0,259	0,264	0,416	0,023	0,378	0,261	0,612	0,021	0,522	0,601	0,527	0,792	0,488	0,433	0,534	0,518	1	0,434
TWMPCI	0,289	0,576	0,636	0,926	-0,042	0,664	-0,104	-0,116	0,602	0,879	0,631	-0,235	0,891	0,553	0,960	0,799	0,989	1,000	0,988	0,996	0,434	1

TABLEAU 8.3 : MATRICE DE CORRÉLATION (PEARSON) DE PETSTORE ASPECTJ

Variables	HDiff	HEff	FO	LCOM1	LCOM3	LOC	NOA	NOCC	RFC	WMPC1	CBO	DOIH	NOO	MTMOOP	TNofCalled	TNofNewData	TNofAssert	TNOO	TLOC	TRFC	TINVOK	TWMPG1
HDiff	1	0,948	0,730	0,554	0,320	0,903	0,342	-0,298	0,773	0,813	0,681	-0,242	0,722	0,630	0,485	0,481	0,469	0,460	0,460	0,456	0,275	0,460
HEff	0,948	1	0,819	0,562	0,360	0,976	0,270	-0,226	0,788	0,870	0,758	-0,247	0,753	0,709	0,551	0,549	0,511	0,549	0,532	0,538	0,288	0,549
FO	0,730	0,819	1	0,538	0,401	0,831	0,332	-0,269	0,838	0,702	0,981	-0,130	0,768	0,962	0,679	0,672	0,628	0,679	0,662	0,665	0,481	0,679
LCOM1	0,554	0,562	0,538	1	0,182	0,570	0,002	0,096	0,674	0,683	0,528	-0,207	0,887	0,497	0,529	0,500	0,497	0,454	0,474	0,474	0,399	0,454
LCOM3	0,320	0,360	0,401	0,182	1	0,383	0,383	-0,138	0,148	0,218	0,329	-0,193	0,231	0,272	0,199	0,167	0,226	0,236	0,206	0,230	0,062	0,236
LOC	0,903	0,976	0,831	0,570	0,383	1	0,263	-0,198	0,810	0,863	0,775	-0,185	0,749	0,728	0,601	0,591	0,564	0,590	0,580	0,588	0,335	0,590
NOA	0,342	0,270	0,332	0,002	0,383	0,263	1	-0,071	0,491	0,030	0,368	0,449	0,267	0,365	0,015	0,100	-0,051	-0,013	0,004	0,009	0,282	-0,013
NOCC	-0,298	-0,226	-0,269	0,096	-0,138	-0,198	-0,071	1	-0,142	-0,170	-0,269	0,327	-0,113	-0,255	-0,129	-0,096	-0,145	-0,161	-0,161	-0,129	0,227	-0,161
RFC	0,773	0,788	0,838	0,674	0,148	0,810	0,491	-0,142	1	0,685	0,835	0,148	0,843	0,824	0,487	0,477	0,429	0,447	0,455	0,457	0,463	0,447
WMPC1	0,813	0,870	0,702	0,683	0,218	0,863	0,030	-0,170	0,685	1	0,674	-0,235	0,827	0,650	0,660	0,638	0,623	0,641	0,636	0,651	0,311	0,641
CBO	0,681	0,758	0,981	0,528	0,329	0,775	0,368	-0,269	0,835	0,674	1	-0,068	0,782	0,993	0,718	0,715	0,648	0,701	0,696	0,699	0,547	0,701
DOIH	-0,242	-0,247	-0,130	-0,207	-0,193	-0,185	0,449	0,327	0,148	-0,235	-0,068	1	-0,093	0,012	-0,185	-0,143	-0,185	-0,192	-0,185	-0,150	0,187	-0,192
NOO	0,722	0,753	0,768	0,887	0,231	0,749	0,267	-0,113	0,843	0,827	0,782	-0,093	1	0,761	0,641	0,614	0,563	0,594	0,600	0,608	0,432	0,594
MTMOOP	0,630	0,709	0,962	0,497	0,272	0,728	0,365	-0,255	0,824	0,650	0,993	0,012	0,761	1	0,718	0,713	0,650	0,702	0,697	0,703	0,565	0,702
TNofCalled	0,485	0,551	0,679	0,529	0,199	0,601	0,015	-0,129	0,487	0,660	0,718	-0,185	0,641	0,718	1	0,971	0,962	0,971	0,987	0,985	0,723	0,971
TNofNewData	0,481	0,549	0,672	0,500	0,167	0,591	0,100	-0,096	0,477	0,638	0,715	-0,143	0,614	0,713	0,971	1	0,930	0,949	0,966	0,961	0,727	0,949
TNofAssert	0,469	0,511	0,628	0,497	0,226	0,564	-0,051	-0,145	0,429	0,623	0,648	-0,185	0,563	0,650	0,962	0,930	1	0,948	0,969	0,961	0,731	0,948
TNOO	0,460	0,549	0,679	0,454	0,236	0,590	-0,013	-0,161	0,447	0,641	0,701	-0,192	0,594	0,702	0,971	0,949	0,948	1	0,990	0,992	0,655	1,000
TLOC	0,460	0,532	0,662	0,474	0,206	0,580	0,004	-0,161	0,455	0,636	0,696	-0,185	0,600	0,697	0,987	0,966	0,969	0,990	1	0,995	0,699	0,990
TRFC	0,456	0,538	0,665	0,474	0,230	0,588	0,009	-0,129	0,457	0,651	0,699	-0,150	0,608	0,703	0,985	0,961	0,961	0,992	0,995	1	0,710	0,992
TINVOK	0,275	0,288	0,481	0,399	0,062	0,335	0,282	0,227	0,463	0,311	0,547	0,187	0,432	0,565	0,723	0,727	0,731	0,655	0,699	0,710	1	0,655
TWMPG1	0,460	0,549	0,679	0,454	0,236	0,590	-0,013	-0,161	0,447	0,641	0,701	-0,192	0,594	0,702	0,971	0,949	0,948	1,000	0,990	0,992	0,655	1

TABLEAU 8.4 : MATRICE DE CORRÉLATION (SPEARMAN) DE PETSTORE_ASPECTJ

Dans les deux versions de Petstore (OO et OA), le modèle MTMOOP corrèle fortement avec notre suite de métriques de testabilité et la majorité des métriques logicielles sélectionnées. En plus, les métriques de cas de test sont fortement corrélées entre elles.

Maintenant, nous allons passer aux courbes représentatives des données recueillies de chaque système. Du côté des classes logicielles, le modèle MTMOOP évolue de cette façon :

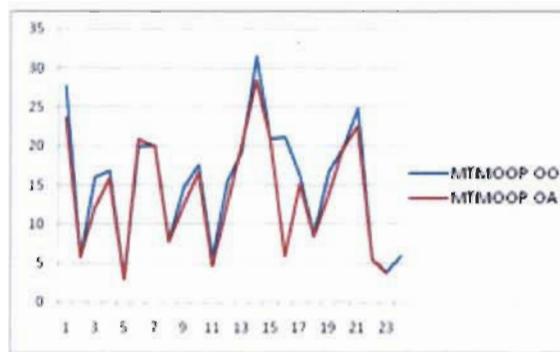


Figure 3.1 MTMOOP : Petstore (OO vs OA)

Du coté des classes tests JUNIT voici l'évolution de chaque métrique :

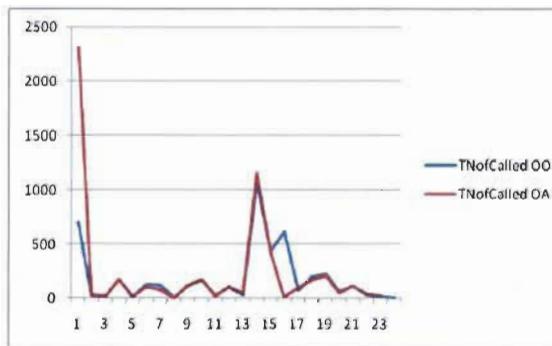


Figure 3.2 TNofCalled : Petstore (OO vs OA)

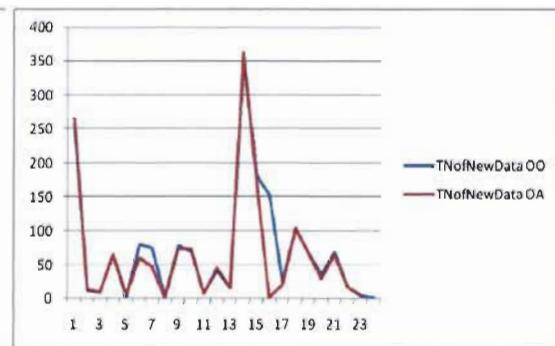


Figure 3.3 TNofNewData : Petstore (OO vs OA)

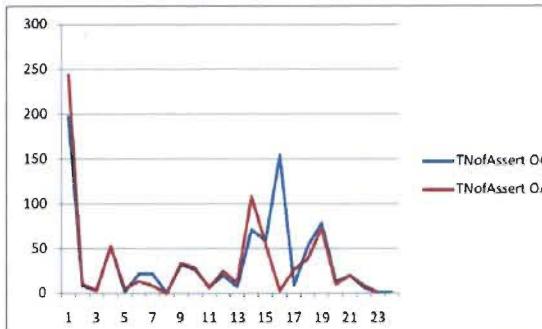


Figure 3.4 TNofAssert: Petstore (OO vs OA)

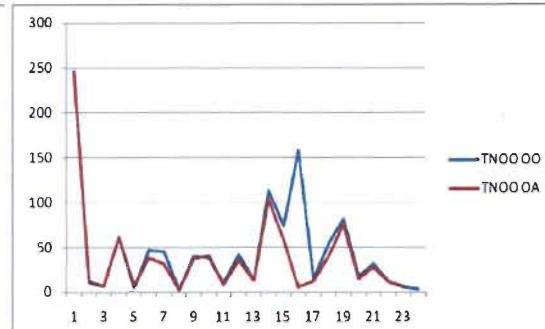


Figure 3.5 TNOO : Petstore (OO vs OA)

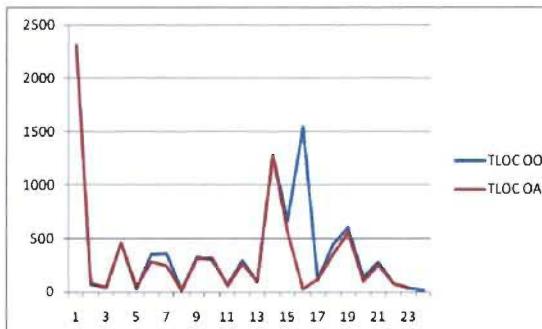


Figure 3.6 TLOC : Petstore (OO vs OA)

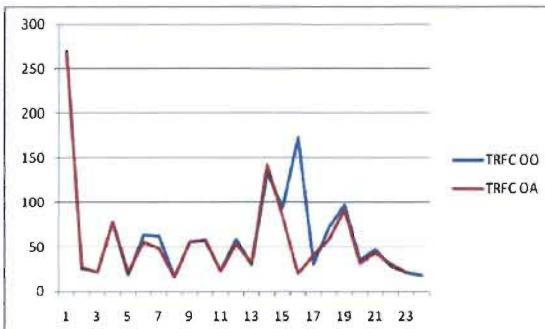


Figure 3.7 TRFC : Petstore (OO vs OA)

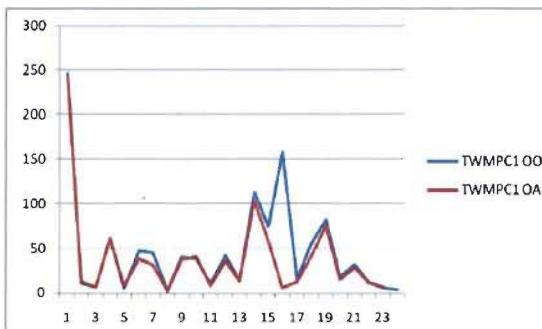


Figure 3.8 TWMP1C1 : Petstore (OO vs OA)

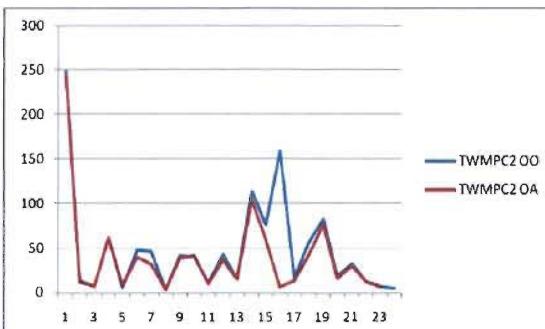


Figure 3.9 TWMP1C2 : Petstore (OO vs OA)

La forte corrélation entre le modèle MTMOOP et notre suite de métriques de testabilité se traduit par ces courbes. On remarque que le modèle MTMOOP prévoit assez bien la testabilité grâce à la ressemblance remarquable entre la courbe du MTMOOP et celles des autres métriques.

Si on observe les extréums de la courbe du modèle MTMOOP (Figure 3.1) et on les vérifie dans les autres figures, on remarquera comme maximum les classes 1, 4, 6, 10 et 21 pour toutes les métriques et comme minimum les classes 2, 5, 8, 11 et 16.

Pour Petstore, toutes les courbes ont la même allure ou presque. On peut également remarquer une légère amélioration du point de vue de la testabilité du côté de la version OA de PetStore. Si on prend par exemple le cas de la classe 16, le modèle MTMOOP a prévu que la testabilité de cette classe après aspectualisation serait améliorée comparée à celle de son équivalent avant aspectualisation. Cela est d'ailleurs confirmé par toutes les mesures obtenues sur les classes tests JUNIT de ces deux classes équivalentes.

Par ailleurs, en effectuant notre test statistique t (voir les résultats dans le tableau 8), nous avons conclu globalement que les différences entre les valeurs des métriques respectives calculées sur les deux versions de Petstore (OO et OA) sont :

- Positives et très significatives au niveau des métriques : HDiff, HEff, LOC, RFC, CBO et MTMOOP.
- Positives et significatives au niveau des métriques de testabilité TNOO et WMPC.
- Négatives et significatives pour LCOM1.
- Les autres métriques n'affichent pas des résultats significatifs.

Variables	Valeur de T	Seuil de Student	#Classes
HDiff	3,75642772	2,485	25
HEff	3,75313479	2,485	25
LCOM1	-3,54854601	2,5	23
LCOM3	0,74851358	2,508	22
LOC	4,69226232	2,485	25
NOA	1,44463024	2,485	25
NOCC	1	2,485	25
RFC	4,75885828	2,485	25
WMPC1	-1,12693318	2,485	25
CBO	3,44037461	2,485	25
DOIH	1	2,485	25
NOO	-4,34248119	2,485	25
MTMOOP	3,59221113	2,485	25
TNofCalled	-1,0293868	2,5	23
TNofNewData	1,09966965	2,5	23
TNofAssert	-1,1650406	2,5	23
TNOO	2,88500339	2,5	23
TLOC	2,25034705	2,5	23
TRFC	1,04514916	2,5	23
TINVOK	-2,13990316	2,5	23
TWMPC	2,92701585	2,5	23

Tableau 8.5 : Résultats du test statistique de PestStore

- Projet 4 : HSQLBD

Variables	HDiff	HEff	FO	LCOM1	LCOM3	LOC	NOA	NOCC	RFC	WMPC1	CBO	DOIH	NOO	MTMOOP	TNofCalled	TNofNewData	TNofAssert	TNOO	TLOC	TRFC	TINVOK	TWMP1
HDiff	1	0,824	0,467	0,378	0,290	0,739	0,457	-0,308	0,455	0,834	0,523	0,058	0,619	0,457	0,379	0,304	0,523	0,678	0,608	0,712	0,431	0,678
HEff	0,824	1	0,469	0,420	0,309	0,893	0,324	-0,061	0,504	0,834	0,516	-0,069	0,554	0,448	0,458	0,385	0,568	0,826	0,639	0,789	0,333	0,826
FO	0,467	0,469	1	0,424	0,213	0,521	0,633	-0,028	0,761	0,592	0,904	0,201	0,489	0,902	0,565	0,572	0,059	0,400	0,568	0,503	0,457	0,400
LCOM1	0,378	0,420	0,424	1	0,481	0,597	0,396	-0,049	0,344	0,679	0,593	-0,149	0,891	0,460	0,474	0,407	0,268	0,503	0,629	0,496	0,209	0,503
LCOM3	0,290	0,309	0,213	0,481	1	0,521	0,444	-0,204	0,227	0,475	0,254	-0,302	0,501	0,145	0,153	0,101	0,282	0,266	0,317	0,291	0,173	0,266
LOC	0,739	0,893	0,521	0,597	0,521	1	0,448	-0,120	0,556	0,861	0,562	-0,064	0,713	0,470	0,533	0,447	0,592	0,847	0,715	0,810	0,343	0,847
NOA	0,457	0,324	0,633	0,396	0,444	0,448	1	-0,009	0,580	0,587	0,598	0,071	0,598	0,543	0,528	0,512	0,096	0,312	0,627	0,466	0,539	0,312
NOCC	-0,308	-0,061	-0,028	-0,049	-0,204	-0,120	-0,009	1	0,194	-0,111	-0,114	0,043	-0,126	-0,095	-0,081	-0,067	-0,205	-0,128	-0,148	-0,182	-0,197	-0,128
RFC	0,455	0,504	0,761	0,344	0,227	0,556	0,580	0,194	1	0,628	0,591	0,276	0,421	0,589	0,343	0,323	0,141	0,358	0,400	0,388	0,256	0,358
WMPC1	0,834	0,834	0,592	0,679	0,475	0,861	0,587	-0,111	0,628	1	0,642	-0,048	0,800	0,542	0,466	0,391	0,487	0,754	0,712	0,800	0,501	0,754
CBO	0,523	0,516	0,904	0,593	0,254	0,562	0,598	-0,114	0,591	0,642	1	0,204	0,616	0,983	0,617	0,590	0,147	0,518	0,682	0,607	0,484	0,518
DOIH	0,058	-0,069	0,201	-0,149	-0,302	-0,064	0,071	0,043	0,276	-0,048	0,204	1	0,006	0,326	0,012	0,023	-0,032	0,046	0,007	0,083	0,114	0,046
NOO	0,619	0,554	0,489	0,891	0,501	0,713	0,598	-0,126	0,421	0,800	0,616	0,006	1	0,486	0,618	0,523	0,512	0,715	0,809	0,745	0,440	0,715
MTMOOP	0,457	0,448	0,902	0,460	0,145	0,470	0,543	-0,095	0,589	0,542	0,983	0,326	0,486	1	0,554	0,542	0,066	0,435	0,589	0,529	0,453	0,435
TNofCalled	0,379	0,458	0,565	0,474	0,153	0,533	0,528	-0,081	0,343	0,466	0,617	0,012	0,618	0,554	1	0,984	0,354	0,650	0,907	0,645	0,322	0,650
TNofNewData	0,304	0,385	0,572	0,407	0,101	0,447	0,512	-0,067	0,323	0,391	0,590	0,023	0,523	0,542	0,984	1	0,215	0,554	0,843	0,561	0,301	0,554
TNofAssert	0,523	0,568	0,059	0,268	0,282	0,592	0,096	-0,205	0,141	0,487	0,147	-0,012	0,512	0,066	0,354	0,215	1	0,748	0,531	0,672	0,198	0,748
TNOO	0,678	0,826	0,400	0,503	0,266	0,847	0,312	-0,128	0,358	0,754	0,518	0,046	0,715	0,435	0,650	0,554	0,748	1	0,792	0,922	0,323	1,000
TLOC	0,608	0,639	0,568	0,629	0,317	0,715	0,627	-0,148	0,400	0,712	0,682	0,007	0,809	0,589	0,907	0,843	0,531	0,792	1	0,849	0,546	0,792
TRFC	0,712	0,789	0,503	0,496	0,291	0,810	0,466	-0,182	0,388	0,800	0,607	0,083	0,745	0,529	0,645	0,561	0,672	0,922	0,849	1	0,664	0,922
TINVOK	0,431	0,333	0,457	0,209	0,173	0,343	0,539	-0,197	0,256	0,501	0,484	0,114	0,440	0,453	0,322	0,301	0,198	0,323	0,546	0,664	1	0,323
TWMP1	0,678	0,826	0,400	0,503	0,266	0,847	0,312	-0,128	0,358	0,754	0,518	0,046	0,715	0,435	0,650	0,554	0,748	1,000	0,792	0,922	0,323	1

- TABLEAU 9.1 : MATRICE DE CORRÉLATION (PEARSON) DE HSQLBD JAVA

Variables	HDifT	HEffT	FO	LCOM1	LCOM3	LOC	NOA	NOCC	RFC	WMPC1	CBO	DOIH	NOO	MTMOOP	TNofCalled	TNofNewData	TNofAssert	TNOO	TLOC	TRFC	TINVOK	TWMPCL
HDifT	1	0,920	0,561	0,348	0,265	0,806	0,500	-0,278	0,681	0,902	0,588	0,168	0,701	0,527	0,684	0,552	0,785	0,793	0,759	0,757	0,520	0,793
HEffT	0,920	1	0,615	0,357	0,488	0,952	0,508	-0,297	0,690	0,902	0,629	0,078	0,706	0,558	0,664	0,511	0,775	0,806	0,758	0,769	0,522	0,806
FO	0,561	0,615	1	0,347	0,275	0,611	0,558	-0,040	0,761	0,662	0,960	0,297	0,507	0,939	0,622	0,706	0,404	0,567	0,614	0,618	0,610	0,567
LCOM1	0,348	0,357	0,347	1	0,610	0,455	0,486	0,137	0,553	0,581	0,400	-0,157	0,704	0,350	0,253	0,172	0,374	0,405	0,391	0,361	0,325	0,405
LCOM3	0,265	0,488	0,275	0,610	1	0,616	0,392	-0,198	0,373	0,430	0,300	-0,333	0,468	0,227	0,140	-0,027	0,360	0,322	0,279	0,275	0,191	0,322
LOC	0,806	0,952	0,611	0,455	0,616	1	0,588	-0,182	0,719	0,865	0,633	0,100	0,755	0,569	0,629	0,481	0,717	0,793	0,750	0,769	0,552	0,793
NOA	0,500	0,508	0,558	0,486	0,392	0,588	1	0,033	0,640	0,598	0,608	0,262	0,658	0,590	0,591	0,614	0,432	0,599	0,625	0,630	0,636	0,599
NOCC	-0,278	-0,297	-0,040	0,137	-0,198	-0,182	0,033	1	0,035	-0,148	-0,049	0,112	-0,075	0,003	-0,238	-0,135	-0,292	-0,146	-0,166	-0,152	-0,142	-0,146
RFC	0,681	0,690	0,761	0,553	0,373	0,719	0,640	0,035	1	0,801	0,810	0,471	0,707	0,800	0,599	0,562	0,567	0,690	0,709	0,736	0,703	0,690
WMPC1	0,902	0,902	0,662	0,581	0,430	0,865	0,598	-0,148	0,801	1	0,693	0,110	0,873	0,619	0,741	0,613	0,785	0,866	0,836	0,841	0,657	0,866
CBO	0,588	0,629	0,960	0,400	0,300	0,633	0,608	-0,049	0,810	0,693	1	0,320	0,566	0,983	0,662	0,727	0,451	0,627	0,681	0,686	0,666	0,627
DOIH	0,168	0,078	0,297	-0,157	-0,333	0,100	0,262	0,112	0,471	0,110	0,320	1	0,111	0,416	0,214	0,277	0,160	0,211	0,241	0,263	0,320	0,211
NOO	0,701	0,706	0,507	0,704	0,468	0,755	0,658	-0,075	0,707	0,873	0,566	0,111	1	0,480	0,768	0,629	0,762	0,856	0,844	0,851	0,737	0,856
MTMOOP	0,527	0,558	0,939	0,350	0,227	0,569	0,590	0,003	0,800	0,619	0,983	0,416	0,480	1	0,588	0,680	0,376	0,545	0,608	0,608	0,609	0,545
TNofCalled	0,684	0,664	0,622	0,253	0,140	0,629	0,591	-0,238	0,599	0,741	0,662	0,214	0,768	0,588	1	0,920	0,789	0,860	0,935	0,889	0,805	0,860
TNofNewData	0,552	0,511	0,706	0,172	-0,027	0,481	0,614	-0,135	0,562	0,613	0,727	0,277	0,629	0,680	0,920	1	0,545	0,676	0,784	0,737	0,758	0,676
TNofAssert	0,785	0,775	0,404	0,374	0,360	0,717	0,432	-0,292	0,567	0,785	0,451	0,160	0,762	0,376	0,789	0,545	1	0,881	0,861	0,833	0,602	0,881
TNOO	0,793	0,806	0,567	0,405	0,322	0,793	0,599	-0,146	0,690	0,866	0,627	0,211	0,856	0,545	0,860	0,676	0,881	1	0,951	0,961	0,743	1,000
TLOC	0,759	0,758	0,614	0,391	0,279	0,750	0,625	-0,166	0,709	0,836	0,681	0,241	0,844	0,608	0,935	0,784	0,861	0,951	1	0,972	0,822	0,951
TRFC	0,757	0,769	0,618	0,361	0,275	0,769	0,630	-0,152	0,736	0,841	0,686	0,263	0,851	0,608	0,889	0,737	0,833	0,961	0,972	1	0,870	0,961
TINVOK	0,520	0,522	0,610	0,325	0,191	0,552	0,636	-0,142	0,703	0,657	0,666	0,320	0,737	0,609	0,805	0,758	0,602	0,743	0,822	0,870	1	0,743
TWMPCL	0,793	0,806	0,567	0,405	0,322	0,793	0,599	-0,146	0,690	0,866	0,627	0,211	0,856	0,545	0,860	0,676	0,881	1,000	0,951	0,961	0,743	1

TABLEAU 9.2 : MATRICE DE CORRÉLATION (SPEARMAN) DE HSQLBD JAVA

Variables	HDiff	HEff	FO	LCOM1	LCOM3	LOC	NOA	NOCC	RFC	WMPC1	CBO	DOIH	NOO	MTMOOP	NofCalled	NofNewData	NofAsseri	TNOO	TLOC	TRFC	TINVOK	TWMPCL
HDiff	1	0,862	0,405	0,428	0,242	0,655	0,504	-0,306	0,445	0,843	0,484	0,052	0,652	0,399	0,253	0,191	0,473	0,569	0,545	0,573	0,427	0,569
HEff	0,862	1	0,377	0,565	0,366	0,754	0,399	-0,132	0,510	0,893	0,454	-0,110	0,637	0,348	0,181	0,110	0,527	0,598	0,512	0,604	0,528	0,598
FO	0,405	0,377	1	0,435	0,211	0,440	0,667	0,021	0,710	0,530	0,859	0,152	0,432	0,855	0,359	0,345	0,125	0,252	0,432	0,256	0,301	0,252
LCOM1	0,428	0,565	0,435	1	0,413	0,592	0,362	-0,056	0,396	0,711	0,634	-0,098	0,893	0,491	0,372	0,308	0,324	0,454	0,596	0,465	0,654	0,454
LCOM3	0,242	0,366	0,211	0,413	1	0,615	0,386	-0,088	0,217	0,399	0,275	-0,183	0,419	0,185	0,339	0,302	0,429	0,438	0,440	0,442	0,324	0,438
LOC	0,655	0,754	0,440	0,592	0,615	1	0,415	-0,124	0,456	0,757	0,493	-0,007	0,629	0,405	0,547	0,474	0,790	0,846	0,741	0,849	0,506	0,846
NOA	0,504	0,399	0,667	0,362	0,386	0,415	1	0,010	0,611	0,589	0,625	0,116	0,602	0,567	0,468	0,452	0,119	0,257	0,578	0,261	0,282	0,257
NOCC	-0,306	-0,132	0,021	-0,056	-0,088	-0,124	0,010	1	0,232	-0,116	-0,091	0,090	-0,130	-0,061	-0,061	-0,059	-0,147	-0,133	-0,125	-0,134	-0,090	-0,133
RFC	0,445	0,510	0,710	0,396	0,217	0,456	0,611	0,232	1	0,634	0,513	0,217	0,454	0,489	0,124	0,093	0,119	0,193	0,251	0,199	0,364	0,193
WMPC1	0,843	0,893	0,530	0,711	0,399	0,757	0,589	-0,116	0,634	1	0,595	-0,029	0,813	0,476	0,296	0,229	0,532	0,615	0,624	0,622	0,592	0,615
CBO	0,484	0,454	0,859	0,634	0,275	0,493	0,625	-0,091	0,513	0,595	1	0,178	0,593	0,979	0,427	0,379	0,202	0,384	0,596	0,388	0,323	0,384
DOIH	0,052	-0,110	0,152	-0,098	-0,183	-0,007	0,116	0,090	0,217	-0,029	0,178	1	0,053	0,307	0,012	0,009	0,054	0,087	0,006	0,092	0,248	0,087
NOO	0,652	0,637	0,432	0,893	0,419	0,629	0,602	-0,130	0,454	0,813	0,593	0,053	1	0,448	0,421	0,346	0,428	0,572	0,698	0,581	0,681	0,572
MTMOOP	0,399	0,348	0,855	0,491	0,185	0,405	0,567	-0,061	0,489	0,476	0,979	0,307	0,448	1	0,377	0,340	0,140	0,312	0,504	0,314	0,245	0,312
NofCalled	0,253	0,181	0,359	0,372	0,339	0,547	0,468	-0,061	0,124	0,296	0,427	0,012	0,421	0,377	1	0,992	0,517	0,617	0,880	0,615	0,206	0,617
NofNewData	0,191	0,110	0,345	0,308	0,302	0,474	0,452	-0,059	0,093	0,229	0,379	0,009	0,346	0,340	0,992	1	0,455	0,539	0,826	0,537	0,181	0,539
NofAsseri	0,473	0,527	0,125	0,324	0,429	0,790	0,119	-0,147	0,119	0,532	0,202	0,054	0,428	0,140	0,517	0,455	1	0,956	0,634	0,956	0,455	0,956
TNOO	0,569	0,598	0,252	0,454	0,438	0,846	0,257	-0,133	0,193	0,615	0,384	0,087	0,572	0,312	0,617	0,539	0,956	1	0,775	1,000	0,457	1,000
TLOC	0,545	0,512	0,432	0,596	0,440	0,741	0,578	-0,125	0,251	0,624	0,596	0,006	0,698	0,504	0,880	0,826	0,634	0,775	1	0,774	0,351	0,775
TRFC	0,573	0,604	0,256	0,465	0,442	0,849	0,261	-0,134	0,199	0,622	0,388	0,092	0,581	0,314	0,615	0,537	0,956	1,000	0,774	1	0,476	1,000
TINVOK	0,427	0,528	0,301	0,654	0,324	0,506	0,282	-0,090	0,364	0,592	0,323	0,248	0,681	0,245	0,206	0,181	0,455	0,457	0,351	0,476	1	0,457
TWMPCL	0,569	0,598	0,252	0,454	0,438	0,846	0,257	-0,133	0,193	0,615	0,384	0,087	0,572	0,312	0,617	0,539	0,956	1,000	0,775	1,000	0,457	1

TABLEAU 9.3 : MATRICE DE CORRÉLATION (PEARSON) DE HSQLBD ASPECTJ

Variables	HDifF	HEifF	FO	LCOM1	LCOM3	LOC	NOA	NOCC	RFC	WMPC1	CBO	DOIH	NOO	MTMOOP	NofCalled	NofNewData	NofAssert	TNOO	TLOC	TRFC	TINVOK	TWMPC
HDifF	1	0,917	0,482	0,495	0,235	0,749	0,545	-0,327	0,703	0,922	0,542	0,128	0,730	0,456	0,619	0,473	0,772	0,746	0,716	0,739	0,464	0,746
HEifF	0,917	1	0,512	0,510	0,488	0,896	0,580	-0,308	0,699	0,891	0,572	0,065	0,726	0,479	0,572	0,401	0,745	0,751	0,707	0,746	0,530	0,751
FO	0,482	0,512	1	0,466	0,347	0,562	0,597	-0,011	0,635	0,584	0,961	0,236	0,461	0,944	0,601	0,633	0,412	0,536	0,620	0,529	0,370	0,536
LCOM1	0,495	0,510	0,466	1	0,483	0,455	0,472	0,119	0,720	0,691	0,542	-0,085	0,753	0,475	0,233	0,168	0,329	0,349	0,399	0,354	0,538	0,349
LCOM3	0,235	0,488	0,347	0,483	1	0,703	0,365	-0,102	0,293	0,356	0,383	-0,212	0,371	0,321	0,284	0,131	0,448	0,410	0,404	0,413	0,416	0,410
LOC	0,749	0,896	0,562	0,455	0,703	1	0,585	-0,168	0,627	0,762	0,594	0,135	0,660	0,515	0,635	0,482	0,761	0,790	0,769	0,790	0,545	0,790
NOA	0,545	0,580	0,597	0,472	0,365	0,585	1	0,033	0,671	0,627	0,630	0,252	0,667	0,608	0,609	0,593	0,478	0,574	0,655	0,581	0,489	0,574
NOCC	-0,327	-0,308	-0,011	0,119	-0,102	-0,168	0,033	1	0,017	-0,175	-0,043	0,154	-0,105	0,008	-0,198	-0,133	-0,289	-0,161	-0,136	-0,158	-0,036	-0,161
RFC	0,703	0,699	0,635	0,720	0,293	0,627	0,671	0,017	1	0,821	0,718	0,438	0,773	0,686	0,465	0,381	0,525	0,629	0,621	0,630	0,602	0,629
WMPC1	0,922	0,891	0,584	0,691	0,356	0,762	0,627	-0,175	0,821	1	0,641	0,117	0,888	0,551	0,629	0,498	0,756	0,766	0,748	0,763	0,612	0,766
CBO	0,542	0,572	0,961	0,542	0,383	0,594	0,630	-0,043	0,718	0,641	1	0,250	0,521	0,982	0,617	0,629	0,454	0,591	0,661	0,585	0,393	0,591
DOIH	0,128	0,065	0,236	-0,085	-0,212	0,135	0,252	0,154	0,438	0,117	0,250	1	0,144	0,340	0,164	0,199	0,165	0,236	0,223	0,244	0,298	0,236
NOO	0,730	0,726	0,461	0,753	0,371	0,660	0,667	-0,105	0,773	0,888	0,521	0,144	1	0,426	0,655	0,528	0,727	0,763	0,758	0,770	0,750	0,763
MTMOOP	0,456	0,479	0,944	0,475	0,321	0,515	0,608	0,008	0,686	0,551	0,982	0,340	0,426	1	0,534	0,578	0,362	0,499	0,578	0,493	0,345	0,499
NofCalled	0,619	0,572	0,601	0,233	0,284	0,635	0,609	-0,198	0,465	0,629	0,617	0,164	0,655	0,534	1	0,939	0,803	0,864	0,936	0,863	0,353	0,864
NofNewData	0,473	0,401	0,633	0,168	0,131	0,482	0,593	-0,133	0,381	0,498	0,629	0,199	0,528	0,578	0,939	1	0,615	0,686	0,808	0,686	0,265	0,686
NofAssert	0,772	0,745	0,412	0,329	0,448	0,761	0,478	-0,289	0,525	0,756	0,454	0,165	0,727	0,362	0,803	0,615	1	0,925	0,878	0,923	0,505	0,925
TNOO	0,746	0,751	0,536	0,349	0,410	0,790	0,574	-0,161	0,629	0,766	0,591	0,236	0,763	0,499	0,864	0,686	0,925	1	0,953	0,999	0,543	1,000
TLOC	0,716	0,707	0,620	0,399	0,404	0,769	0,655	-0,136	0,621	0,748	0,661	0,223	0,758	0,578	0,936	0,808	0,878	0,953	1	0,954	0,488	0,953
TRFC	0,739	0,746	0,529	0,354	0,413	0,790	0,581	-0,158	0,630	0,763	0,585	0,244	0,770	0,493	0,863	0,686	0,923	0,999	0,954	1	0,555	0,999
TINVOK	0,464	0,530	0,370	0,538	0,416	0,545	0,489	-0,036	0,602	0,612	0,393	0,298	0,750	0,345	0,353	0,265	0,505	0,543	0,488	0,555	1	0,543
TWMPC	0,746	0,751	0,536	0,349	0,410	0,790	0,574	-0,161	0,629	0,766	0,591	0,236	0,763	0,499	0,864	0,686	0,925	1,000	0,953	0,999	0,543	1

TABLEAU 9.4 : MATRICE DE CORRÉLATION (SPEARMAN) DE HSQLBD ASPECTJ

On observe une forte corrélation entre le modèle MTMOOP et les métriques TNofCalled, TNofNewData, TNOO, TLOC, TRFC, TINVOK et TWMPC. La métrique NofAssert ne corrèle pas avec le modèle MTMOOP dans le cas de Pearson, mais corrèle assez bien en utilisant le coefficient de corrélation de Spearman, ce qui prouve qu'il y a une relation non linéaire entre ces deux métriques.

Pour la version AspectJ de HSQLBD, on observe aussi une forte corrélation entre le modèle MTMOOP et les métriques TNofCalled, TNofNewData, TNOO, TLOC, TRFC, TINVOK et TWMPC. Comme pour la version Java de HSQLBD, la métrique NofAssert ne corrèle pas avec le modèle MTMOOP dans le cas de Pearson, mais corrèle bien en utilisant le coefficient de corrélation de Spearman, ce qui prouve qu'il y a une relation non linéaire entre ces deux métriques.

Maintenant, nous allons passer aux courbes représentatives des données recueillies de chaque système. Du côté des classes logicielles, le modèle MTMOOP évolue de cette façon :

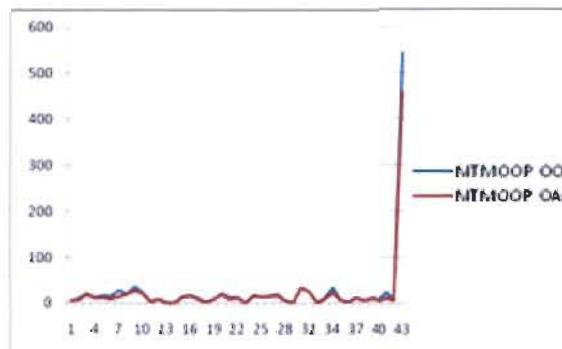


Figure 4.1 MTMOOP : HSQLBD OO vs HSQLBD OA

Du côté des classes tests JUNIT, voici l'évolution de chaque métrique :

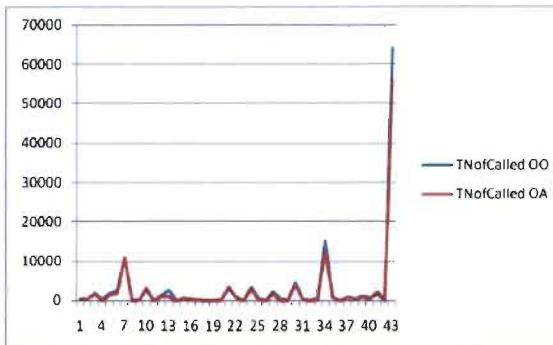


Figure 4.2 TNofCalled : HSQLBD OO vs HSQLBD OA

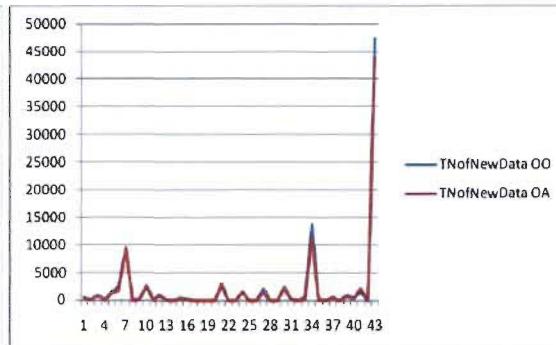


Figure 4.3 TNofNewData : HSQLBD OO vs HSQLBD OA

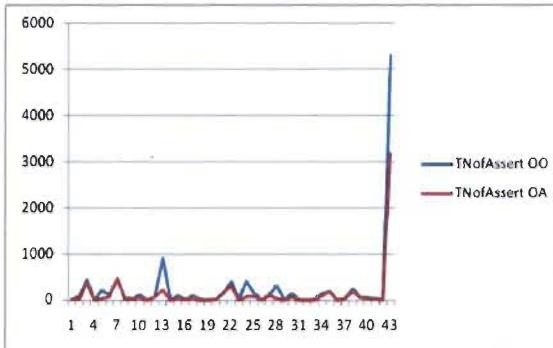


Figure 4.4 TNofAssert : HSQLBD OO vs HSQLBD OA

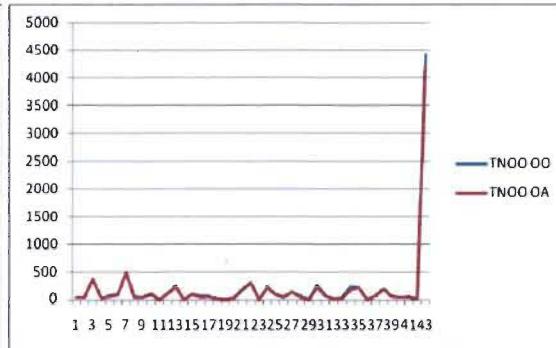


Figure 4.5 TNOO : HSQLBD OO vs HSQLBD OA

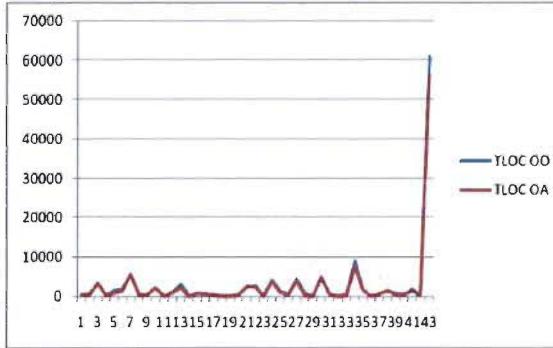


Figure 4.6 TLOC : HSQLBD OO vs HSQLBD OA

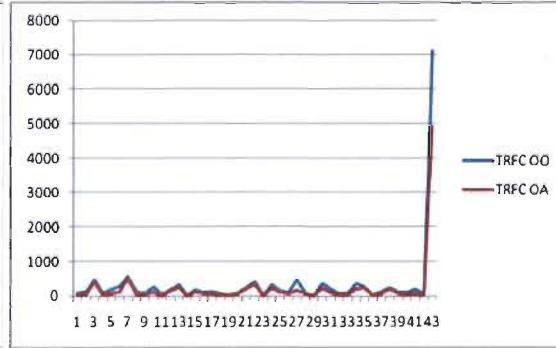


Figure 4.7 TRFC : HSQLBD OO vs HSQLBD OA

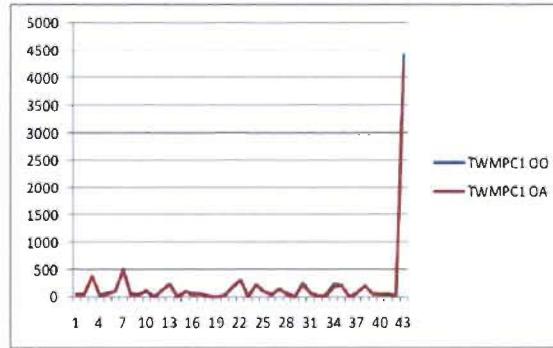


Figure 4.8 TWMPG1 : HSQLBD OO vs HSQLBD OA

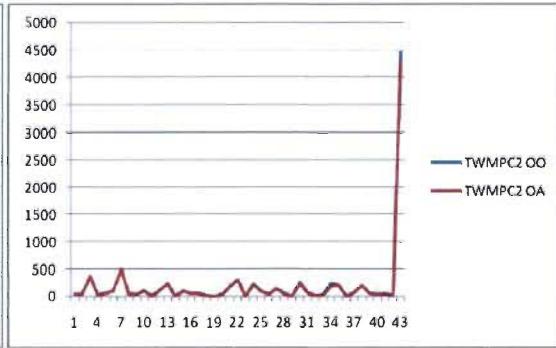


Figure 4.9 TWMPG2 : HSQLBD OO vs HSQLBD OA

Comme pour Petstore, nous avons obtenu avec HSQLBD de bons résultats. Les courbes ont la même allure. Par ailleurs, on peut facilement identifier les classes les plus complexes grâce aux courbes de la figure 4.1. Les classes tests les plus complexes (identifiées grâce aux figures des métriques de testabilité 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8 et 4.9) correspondent aux classes logicielles identifiées dans la figure 4.1.

En ce qui concerne la comparaison entre la version OO vs OA, le modèle MTMOOP indique juste quelques améliorations après aspectualisation, surtout au niveau des classes 7, 13, 34, 41 et 43. Ces prédictions sont confirmées par les valeurs obtenues par notre suite de métrique de testabilité, surtout au niveau des métriques TnofAssert et TRFC. D'après les figures précédentes, nous pouvons remarquer que les courbes appartenant à la version AspectJ de HSQLBD sont, tout le temps, égales ou légèrement au-dessous des courbes modélisant la version Java de ce programme.

Par ailleurs, en effectuant notre test statistique t (voir les résultats dans le tableau 9), nous avons conclu globalement que les différences entre les valeurs des métriques respectives calculées sur les deux versions de HSQLBD (OO et OA) sont :

- Positives pour toutes les métriques (logicielles et de testabilité).
- Très significatives pour la plupart des métriques logicielles.
- Très significatives pour la plupart des métriques de testabilité, à part les métriques : TNofNewData, TNOO et WMPC, qui ne sont pas loin du seuil de signification.

Variables	Valeur de T	Seuil de Student	#Classes
HDiff	2,60015851	2,423	42
HEff	1,84668358	2,423	42
LCOM1	1,77601776	2,438	36
LCOM3	1,01233153	2,438	34
LOC	4,94513068	2,423	42
NOA	2,44429666	2,423	42
NOCC	0	2,423	42
RFC	2,92954022	2,423	42
WMPC1	3,25589453	2,423	42
CBO	4,35701293	2,423	42
DOIH	1	2,423	42
NOO	3,57839043	2,423	42
MTMOOP	4,25059463	2,423	42
TNofCalled	2,70536419	2,423	42
TNofNewData	1,47929566	2,423	42
TNofAssert	2,69071516	2,423	42
TNOO	2,3671879	2,423	42
TLOC	2,75584236	2,423	42
TRFC	5,72229653	2,423	42
TINVOK	5,4747651	2,423	42
TWMPC	2,3671879	2,423	42

Tableau 9 : Résultats du test statistique de HSQLBD

- Projet 5 : JHotdraw

Variables	HDifT	HEffT	FO	LCOM1	LCOM3	LOC	NOA	NOCC	RFC	WMPC1	CBO	DOIH	NOO	MTMOOP	TNofCalled	TNofNewData	TNofAssert	TNOO	TLOC	TRFC	TINVOK	TWMPCL
HDifT	1	0,836	0,440	0,218	0,254	0,576	0,443	-0,154	0,293	0,475	0,406	0,347	0,337	0,421	0,587	0,553	0,533	0,634	0,625	0,538	0,398	0,634
HEffT	0,836	1	0,674	0,558	0,361	0,812	0,623	-0,081	0,424	0,734	0,653	0,325	0,607	0,640	0,777	0,701	0,731	0,774	0,810	0,679	0,526	0,774
FO	0,440	0,674	1	0,865	0,500	0,877	0,823	0,071	0,655	0,877	0,911	0,398	0,894	0,881	0,817	0,754	0,709	0,800	0,824	0,806	0,730	0,800
LCOM1	0,218	0,558	0,865	1	0,436	0,875	0,775	0,136	0,720	0,895	0,863	0,403	0,934	0,829	0,859	0,828	0,719	0,906	0,886	0,872	0,729	0,906
LCOM3	0,254	0,361	0,500	0,436	1	0,502	0,603	0,209	0,308	0,550	0,535	0,147	0,549	0,507	0,484	0,521	0,340	0,493	0,479	0,560	0,540	0,493
LOC	0,576	0,812	0,877	0,875	0,502	1	0,861	-0,021	0,640	0,931	0,888	0,436	0,836	0,869	0,912	0,887	0,785	0,939	0,949	0,903	0,780	0,939
NOA	0,443	0,623	0,823	0,775	0,603	0,861	1	-0,005	0,628	0,815	0,824	0,430	0,761	0,811	0,781	0,784	0,554	0,704	0,749	0,768	0,747	0,704
NOCC	-0,154	-0,081	0,071	0,136	0,209	-0,021	-0,005	1	-0,121	0,101	0,037	-0,376	0,187	-0,031	-0,032	-0,018	-0,025	0,106	0,011	0,126	0,131	0,106
RFC	0,293	0,424	0,655	0,720	0,308	0,640	0,628	-0,121	1	0,634	0,670	0,848	0,622	0,722	0,846	0,821	0,699	0,831	0,850	0,819	0,725	0,831
WMPC1	0,475	0,734	0,877	0,895	0,550	0,931	0,815	0,101	0,634	1	0,832	0,312	0,954	0,783	0,870	0,892	0,703	0,961	0,931	0,893	0,742	0,961
CBO	0,406	0,653	0,911	0,863	0,535	0,888	0,824	0,037	0,670	0,832	1	0,479	0,826	0,990	0,849	0,778	0,771	0,830	0,849	0,849	0,780	0,830
DOIH	0,347	0,325	0,398	0,403	0,147	0,436	0,430	-0,376	0,848	0,312	0,479	1	0,249	0,589	0,369	0,312	0,352	0,271	0,338	0,276	0,252	0,271
NOO	0,337	0,607	0,894	0,934	0,549	0,836	0,761	0,187	0,622	0,954	0,826	0,249	1	0,763	0,848	0,867	0,676	0,935	0,884	0,909	0,793	0,935
MTMOOP	0,421	0,640	0,881	0,829	0,507	0,869	0,811	-0,031	0,722	0,783	0,990	0,589	0,763	1	0,823	0,738	0,762	0,778	0,814	0,804	0,744	0,778
TNofCalled	0,587	0,777	0,817	0,859	0,484	0,912	0,781	-0,032	0,846	0,870	0,849	0,369	0,848	0,823	1	0,950	0,866	0,910	0,980	0,924	0,841	0,910
TNofNewData	0,553	0,701	0,754	0,828	0,521	0,887	0,784	-0,018	0,821	0,892	0,778	0,312	0,867	0,738	0,950	1	0,688	0,882	0,942	0,890	0,807	0,882
TNofAssert	0,533	0,731	0,709	0,719	0,340	0,785	0,554	-0,025	0,699	0,703	0,771	0,352	0,676	0,762	0,866	0,688	1	0,831	0,856	0,834	0,751	0,831
TNOO	0,634	0,774	0,800	0,906	0,493	0,939	0,704	0,106	0,831	0,961	0,830	0,271	0,935	0,778	0,910	0,882	0,831	1	0,957	0,938	0,788	1,000
TLOC	0,625	0,810	0,824	0,886	0,479	0,949	0,749	0,011	0,850	0,931	0,849	0,338	0,884	0,814	0,980	0,942	0,856	0,957	1	0,923	0,800	0,957
TRFC	0,538	0,679	0,806	0,872	0,560	0,903	0,768	0,126	0,819	0,893	0,849	0,276	0,909	0,804	0,924	0,890	0,834	0,938	0,923	1	0,952	0,938
TINVOK	0,398	0,526	0,730	0,729	0,540	0,780	0,747	0,131	0,725	0,742	0,780	0,252	0,793	0,744	0,841	0,807	0,751	0,788	0,800	0,952	1	0,788
TWMPCL	0,634	0,774	0,800	0,906	0,493	0,939	0,704	0,106	0,831	0,961	0,830	0,271	0,935	0,778	0,910	0,882	0,831	1,000	0,957	0,938	0,788	1

TABLEAU 10.1 : MATRICE DE CORRÉLATION (PEARSON) DE JHOTDRAW JAVA

Variables	HDiff	HEff	FO	LCOM1	LCOM3	LOC	NOA	NOCC	RFC	WMPC1	CBO	DOIH	NOO	MTMOOP	TNofCalled	TNofNewData	TNofAssert	TNOO	TLOC	TRFC	TINVOK	TWMPC1
HDiff	1	0,962	0,544	0,372	0,410	0,808	0,668	0,143	0,696	0,536	0,632	0,524	0,419	0,683	0,683	0,654	0,646	0,713	0,717	0,668	0,547	0,713
HEff	0,962	1	0,673	0,495	0,445	0,910	0,730	0,202	0,756	0,616	0,719	0,538	0,529	0,763	0,782	0,733	0,777	0,787	0,806	0,756	0,641	0,787
FO	0,544	0,673	1	0,761	0,540	0,787	0,643	0,379	0,744	0,784	0,904	0,403	0,803	0,839	0,757	0,693	0,773	0,745	0,780	0,705	0,623	0,745
LCOM1	0,372	0,495	0,761	1	0,700	0,739	0,604	0,507	0,609	0,945	0,746	0,260	0,969	0,647	0,858	0,851	0,743	0,901	0,867	0,913	0,825	0,901
LCOM3	0,410	0,445	0,540	0,700	1	0,590	0,875	0,208	0,445	0,707	0,689	0,287	0,704	0,649	0,597	0,648	0,444	0,560	0,589	0,645	0,687	0,560
LOC	0,808	0,910	0,787	0,739	0,590	1	0,770	0,249	0,787	0,752	0,782	0,471	0,710	0,778	0,835	0,791	0,816	0,827	0,838	0,841	0,755	0,827
NOA	0,668	0,730	0,643	0,604	0,875	0,770	1	0,260	0,676	0,692	0,750	0,367	0,679	0,726	0,805	0,803	0,707	0,784	0,808	0,797	0,759	0,784
NOCC	0,143	0,202	0,379	0,507	0,208	0,249	0,260	1	0,241	0,565	0,341	-0,118	0,579	0,218	0,537	0,511	0,524	0,557	0,548	0,571	0,540	0,557
RFC	0,696	0,756	0,744	0,609	0,445	0,787	0,676	0,241	1	0,659	0,780	0,770	0,638	0,865	0,748	0,693	0,738	0,719	0,760	0,731	0,682	0,719
WMPC1	0,536	0,616	0,784	0,945	0,707	0,752	0,692	0,565	0,659	1	0,827	0,242	0,971	0,710	0,948	0,950	0,848	0,976	0,963	0,946	0,855	0,976
CBO	0,632	0,719	0,904	0,746	0,689	0,782	0,750	0,341	0,780	0,827	1	0,484	0,800	0,954	0,819	0,786	0,782	0,805	0,836	0,751	0,679	0,805
DOIH	0,524	0,538	0,103	0,260	0,287	0,471	0,367	-0,118	0,770	0,242	0,484	1	0,202	0,681	0,259	0,230	0,310	0,223	0,261	0,241	0,230	0,223
NOO	0,419	0,529	0,803	0,969	0,704	0,710	0,679	0,579	0,638	0,971	0,800	0,202	1	0,673	0,942	0,925	0,878	0,937	0,938	0,953	0,891	0,937
MTMOOP	0,683	0,763	0,839	0,647	0,649	0,778	0,726	0,218	0,865	0,710	0,954	0,681	0,673	1	0,758	0,716	0,753	0,729	0,772	0,685	0,624	0,729
TNofCalled	0,683	0,782	0,757	0,858	0,597	0,835	0,805	0,537	0,748	0,948	0,819	0,259	0,942	0,758	1	0,984	0,928	0,952	0,976	0,959	0,892	0,952
TNofNewData	0,654	0,733	0,693	0,851	0,648	0,791	0,803	0,511	0,693	0,950	0,786	0,230	0,925	0,716	0,984	1	0,877	0,958	0,971	0,946	0,878	0,958
TNofAssert	0,646	0,777	0,773	0,743	0,444	0,816	0,707	0,524	0,738	0,848	0,782	0,310	0,878	0,753	0,928	0,877	1	0,861	0,892	0,893	0,830	0,861
TNOO	0,713	0,787	0,745	0,901	0,560	0,827	0,784	0,557	0,719	0,976	0,805	0,223	0,937	0,729	0,952	0,958	0,861	1	0,982	0,936	0,821	1,000
TLOC	0,717	0,806	0,780	0,867	0,589	0,838	0,808	0,548	0,760	0,963	0,836	0,261	0,938	0,772	0,976	0,971	0,892	0,982	1	0,935	0,838	0,982
TRFC	0,668	0,756	0,705	0,913	0,645	0,841	0,797	0,571	0,731	0,946	0,751	0,241	0,953	0,685	0,959	0,946	0,893	0,936	0,935	1	0,957	0,936
TINVOK	0,547	0,641	0,623	0,825	0,687	0,755	0,759	0,540	0,682	0,855	0,679	0,230	0,891	0,624	0,892	0,878	0,830	0,821	0,838	0,957	1	0,821
TWMPC1	0,713	0,787	0,745	0,901	0,560	0,827	0,784	0,557	0,719	0,976	0,805	0,223	0,937	0,729	0,952	0,958	0,861	1,000	0,982	0,936	0,821	1

TABLEAU 10.2 : MATRICE DE CORRÉLATION (SPEARMAN) DE JHOTDRAW JAVA

Variables	HDiff	HEff	FO	LCOM1	LCOM3	LOC	NOA	NOCC	RFC	WMPC1	CBO	DOIH	NOO	MTMOOP	TNofCalled	TNofNewData	TNofAssert	TNOO	TLOC	TRFC	TINVOK	TWMPCT
HDiff	1	0,845	0,452	0,223	0,219	0,598	0,456	-0,170	0,319	0,507	0,423	0,341	0,376	0,435	0,765	0,783	0,660	0,655	0,773	0,441	0,159	0,655
HEff	0,845	1	0,665	0,542	0,344	0,807	0,610	-0,094	0,414	0,732	0,644	0,311	0,606	0,631	0,855	0,896	0,796	0,793	0,909	0,605	0,329	0,793
FO	0,452	0,665	1	0,872	0,472	0,883	0,821	0,022	0,662	0,886	0,920	0,384	0,908	0,890	0,657	0,639	0,763	0,806	0,681	0,880	0,846	0,806
LCOM1	0,223	0,542	0,872	1	0,427	0,877	0,766	0,112	0,711	0,895	0,871	0,394	0,934	0,838	0,768	0,731	0,803	0,874	0,771	0,942	0,986	0,874
LCOM3	0,219	0,344	0,472	0,427	1	0,505	0,601	0,171	0,284	0,532	0,514	0,098	0,524	0,485	0,758	0,731	0,620	0,712	0,689	0,735	0,739	0,712
LOC	0,598	0,807	0,883	0,877	0,505	1	0,861	-0,041	0,651	0,942	0,901	0,434	0,852	0,883	0,935	0,951	0,879	0,892	0,959	0,728	0,463	0,892
NOA	0,456	0,610	0,821	0,766	0,601	0,861	1	-0,036	0,636	0,811	0,832	0,441	0,757	0,824	0,856	0,806	0,485	0,586	0,665	0,480	0,307	0,586
NOCC	-0,170	-0,094	0,022	0,112	0,171	-0,041	-0,036	1	-0,131	0,070	0,007	-0,399	0,153	-0,060	0,100	0,065	0,310	0,284	0,171	0,271	0,224	0,284
RFC	0,319	0,414	0,662	0,711	0,284	0,651	0,636	-0,131	1	0,634	0,678	0,839	0,619	0,732	0,876	0,867	0,728	0,767	0,826	0,627	0,400	0,767
WMPC1	0,507	0,732	0,886	0,895	0,532	0,942	0,811	0,070	0,634	1	0,843	0,306	0,955	0,796	0,897	0,924	0,967	0,988	0,978	0,930	0,751	0,988
CBO	0,423	0,644	0,920	0,871	0,514	0,901	0,832	0,007	0,678	0,843	1	0,464	0,839	0,990	0,732	0,707	0,771	0,802	0,726	0,784	0,665	0,802
DOIH	0,341	0,311	0,384	0,394	0,098	0,434	0,441	-0,399	0,839	0,306	0,464	1	0,243	0,574	0,025	0,016	-0,249	-0,246	-0,096	-0,398	-0,508	-0,246
NOO	0,376	0,606	0,908	0,934	0,524	0,852	0,757	0,153	0,619	0,955	0,839	0,243	1	0,779	0,774	0,775	0,872	0,922	0,824	0,979	0,915	0,922
MTMOOP	0,435	0,631	0,890	0,838	0,485	0,883	0,824	-0,060	0,732	0,796	0,990	0,574	0,779	1	0,669	0,638	0,635	0,659	0,625	0,593	0,448	0,659
TNofCalled	0,765	0,855	0,657	0,768	0,758	0,935	0,856	0,100	0,876	0,897	0,732	0,025	0,774	0,669	1	0,991	0,830	0,897	0,940	0,803	0,602	0,897
TNofNewData	0,783	0,896	0,639	0,731	0,731	0,951	0,806	0,065	0,867	0,924	0,707	0,016	0,775	0,638	0,991	1	0,861	0,914	0,966	0,819	0,615	0,914
TNofAssert	0,660	0,796	0,763	0,803	0,620	0,879	0,485	0,310	0,728	0,967	0,771	-0,249	0,872	0,635	0,830	0,861	1	0,984	0,955	0,901	0,700	0,984
TNOO	0,655	0,793	0,806	0,874	0,712	0,892	0,586	0,284	0,767	0,988	0,802	-0,246	0,922	0,659	0,897	0,914	0,984	1	0,971	0,942	0,760	1,000
TLOC	0,773	0,909	0,681	0,771	0,689	0,959	0,665	0,171	0,826	0,978	0,726	-0,096	0,824	0,625	0,940	0,966	0,955	0,971	1	0,862	0,638	0,971
TRFC	0,441	0,605	0,880	0,942	0,735	0,728	0,480	0,271	0,627	0,930	0,784	-0,398	0,979	0,593	0,803	0,819	0,901	0,942	0,862	1	0,935	0,942
TINVOK	0,159	0,329	0,846	0,986	0,739	0,463	0,307	0,224	0,400	0,751	0,665	-0,508	0,915	0,448	0,602	0,615	0,700	0,760	0,638	0,935	1	0,760
TWMPCT	0,655	0,793	0,806	0,874	0,712	0,892	0,586	0,284	0,767	0,988	0,802	-0,246	0,922	0,659	0,897	0,914	0,984	1,000	0,971	0,942	0,760	1

TABLEAU 10.3 : MATRICE DE CORRÉLATION (PEARSON) DE AJHOTDRAW ASPECTJ

Variables	HDiff	HEff	FO	LCOM1	LCOM3	LOC	NOA	NOCC	RFC	WMPC1	CBO	DOIH	NOO	MTMOOP	TNofCalled	TNofNewData	TNofAssert	TNOO	TLOC	TRFC	TINVOK	TWMPC1
HDiff	1	0,970	0,581	0,441	0,357	0,837	0,737	0,131	0,744	0,641	0,679	0,526	0,528	0,739	0,350	0,478	0,283	0,334	0,382	0,242	0,143	0,334
HEff	0,970	1	0,679	0,546	0,396	0,916	0,785	0,166	0,773	0,698	0,753	0,524	0,612	0,799	0,375	0,500	0,310	0,357	0,406	0,268	0,170	0,357
FO	0,581	0,679	1	0,827	0,527	0,823	0,646	0,329	0,782	0,849	0,932	0,364	0,863	0,855	0,847	0,824	0,804	0,862	0,879	0,861	0,811	0,862
LCOM1	0,441	0,546	0,827	1	0,680	0,799	0,614	0,506	0,669	0,939	0,828	0,309	0,968	0,728	0,903	0,770	0,891	0,891	0,855	0,891	0,900	0,891
LCOM3	0,357	0,396	0,527	0,680	1	0,588	0,860	0,160	0,410	0,684	0,675	0,270	0,682	0,648	0,955	0,883	0,811	0,811	0,883	0,811	0,855	0,811
LOC	0,837	0,916	0,823	0,799	0,588	1	0,812	0,293	0,819	0,858	0,850	0,443	0,814	0,823	0,865	0,922	0,794	0,834	0,861	0,803	0,734	0,834
NOA	0,737	0,785	0,646	0,614	0,860	0,812	1	0,212	0,682	0,706	0,757	0,403	0,689	0,757	0,563	0,648	0,501	0,494	0,567	0,448	0,392	0,494
NOCC	0,131	0,166	0,329	0,506	0,160	0,293	0,242	1	0,222	0,527	0,295	-0,158	0,583	0,168	0,514	0,355	0,627	0,568	0,481	0,614	0,680	0,568
RFC	0,744	0,773	0,782	0,669	0,410	0,819	0,682	0,222	1	0,748	0,816	0,706	0,721	0,874	0,785	0,824	0,737	0,765	0,759	0,715	0,624	0,765
WMPC1	0,641	0,698	0,849	0,939	0,684	0,858	0,706	0,527	0,748	1	0,877	0,262	0,968	0,763	0,964	0,885	0,944	0,983	0,964	0,981	0,960	0,983
CBO	0,679	0,753	0,932	0,828	0,675	0,850	0,757	0,295	0,816	0,877	1	0,421	0,858	0,948	0,804	0,849	0,726	0,798	0,827	0,784	0,710	0,798
DOIH	0,526	0,524	0,364	0,309	0,270	0,443	0,403	-0,158	0,706	0,262	0,421	1	0,228	0,637	-0,318	-0,195	-0,353	-0,346	-0,307	-0,437	-0,506	-0,346
NOO	0,528	0,612	0,863	0,968	0,682	0,814	0,689	0,583	0,721	0,968	0,858	0,228	1	0,734	0,912	0,803	0,957	0,937	0,890	0,966	0,968	0,937
MTMOOP	0,739	0,799	0,855	0,728	0,648	0,823	0,757	0,168	0,874	0,763	0,948	0,637	0,734	1	0,577	0,673	0,518	0,579	0,607	0,544	0,453	0,579
TNofCalled	0,350	0,375	0,847	0,903	0,955	0,865	0,563	0,514	0,785	0,964	0,804	-0,318	0,912	0,577	1	0,944	0,943	0,974	0,966	0,969	0,943	0,974
TNofNewData	0,478	0,500	0,824	0,770	0,883	0,922	0,648	0,355	0,824	0,885	0,849	-0,195	0,803	0,673	0,944	1	0,857	0,900	0,931	0,888	0,833	0,900
TNofAssert	0,283	0,310	0,804	0,891	0,811	0,794	0,501	0,627	0,737	0,944	0,726	-0,353	0,957	0,518	0,943	0,857	1	0,966	0,933	0,954	0,937	0,966
TNOO	0,334	0,357	0,862	0,891	0,811	0,834	0,494	0,568	0,765	0,983	0,798	-0,346	0,937	0,579	0,974	0,900	0,966	1	0,980	0,983	0,950	1,000
TLOC	0,382	0,406	0,879	0,855	0,883	0,861	0,567	0,481	0,759	0,964	0,827	-0,307	0,890	0,607	0,966	0,931	0,933	0,980	1	0,960	0,920	0,980
TRFC	0,242	0,268	0,861	0,891	0,811	0,803	0,448	0,614	0,715	0,981	0,784	-0,437	0,966	0,544	0,969	0,888	0,954	0,983	0,960	1	0,979	0,983
TINVOK	0,143	0,170	0,811	0,900	0,855	0,734	0,392	0,680	0,624	0,960	0,710	-0,506	0,968	0,453	0,943	0,833	0,937	0,950	0,920	0,979	1	0,950
TWMPC1	0,334	0,357	0,862	0,891	0,811	0,834	0,494	0,568	0,765	0,983	0,798	-0,346	0,937	0,579	0,974	0,900	0,966	1,000	0,980	0,983	0,950	1

TABLEAU 10.4 : MATRICE DE CORRÉLATION (SPEARMAN) DE AJHOTDRAW ASPECTJ

On remarque une forte corrélation (Person et Spearman) entre toutes les métriques sélectionnées avec le modèle MTMOOP (métriques logicielles et métriques de testabilité) dans les deux versions du système (OO et OA). Il y a juste la métrique NOCC qui ne corrèle pas avec le modèle MTMOOP.

On remarque aussi que les métriques de cas de tests sont très fortement corrélées entre elles.

Maintenant, nous allons passer aux courbes représentatives des données recueillies de chaque système. Du côté des classes logicielles, le modèle MTMOOP évolue de cette façon :

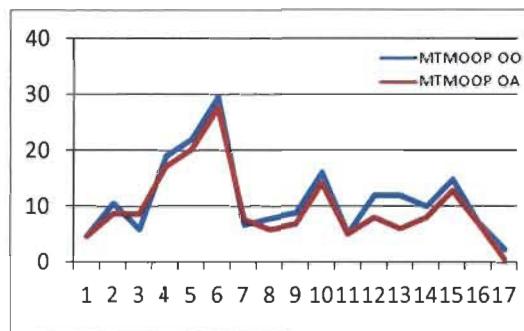


Figure 5.1 MTMOOP : JHotDraw vs AJHotDraw

Du côté des classes tests JUNIT, voici l'évolution de chaque métrique :

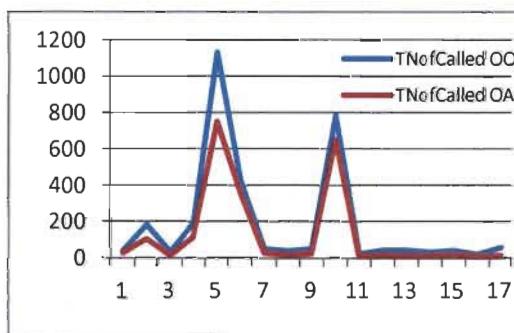


Figure 5.2 TNofCalled : JHotDraw vs AJHotDraw

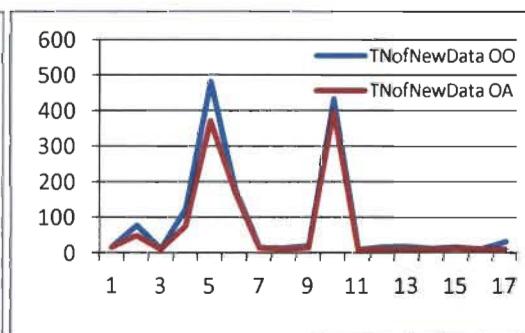


Figure 5.3 TNofNewData: JHotDraw vs AJHotDraw

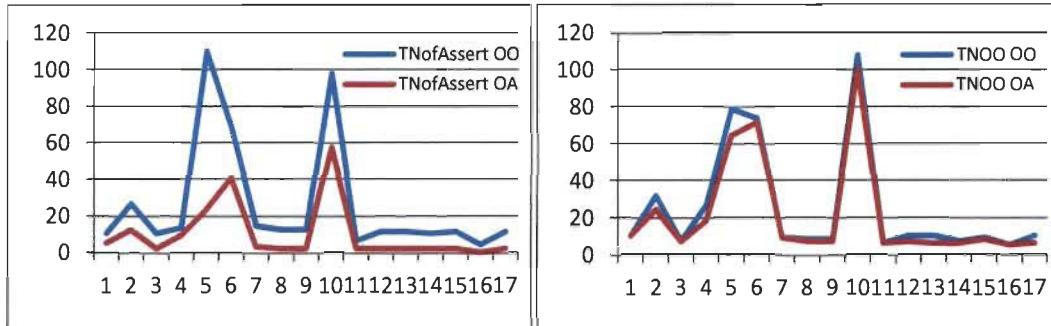


Figure 5.4 TNofAssert : JHotDraw vs AJHotDraw

Figure 5.5 TNOO : JHotDraw vs AJHotDraw

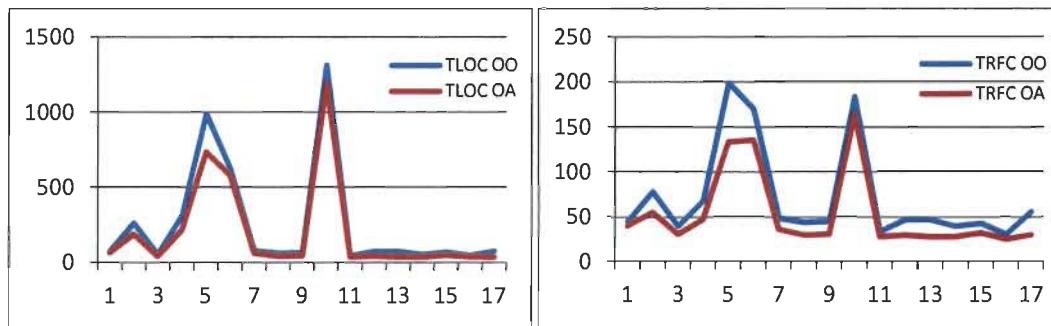


Figure 5.6 TLOC: JHotDraw vs AJHotDraw

Figure 5.7 TRFC : JHotDraw vs AJHotDraw

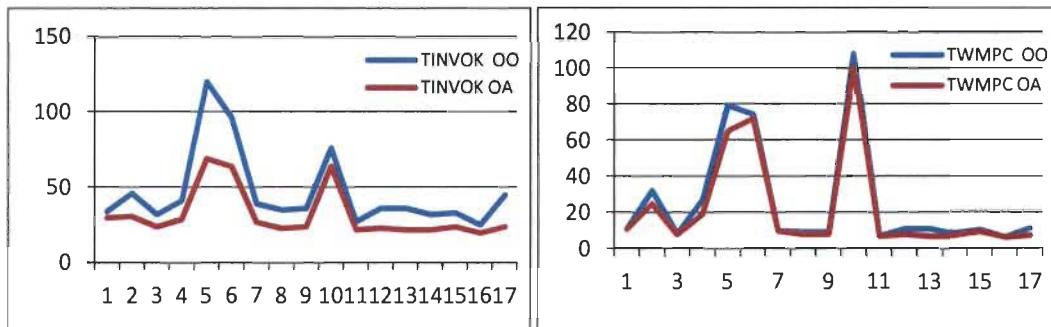


Figure 5.8 TINVOK : JHotDraw vs AJHotDraw

Figure 5.9 TWMPC : JHotDraw vs AJHotDraw

Comme pour Petstore et HSQLBD, nous avons obtenu de très bons résultats avec ce système. Les courbes ont la même allure. Les courbes de la Figure 5.1 prévoient que les classes 5, 6 et 10 sont celles qui demanderont plus d'efforts de test. Les métriques des cas de test confirment les prédictions du MTMOOP.

En ce qui concerne la comparaison entre la version OO vs OA, le modèle MTMOOP indique une amélioration visible après aspectualisation surtout au niveau des classes 12, 13, 14 et 15. Ces prédictions sont confirmées par les valeurs obtenues par notre suite de métriques de testabilité. D'après les figures précédentes, on remarque que les courbes relatives à la version AspectJ de AJHotDraw sont, tout le temps, égales ou au-dessous des courbes modélisant la version Java de ce programme.

Par ailleurs, en effectuant notre test statistique t (voir les résultats dans le tableau 10), nous avons conclu globalement que les différences entre les valeurs des métriques respectives calculées sur les deux versions du système (OO et OA) sont :

- Positives pour toutes les métriques (logicielles et de testabilité) sauf LCOM3.
- Très significatives pour la plupart des métriques logicielles.
- Très significatives pour les métriques de testabilité, à part la métrique NofNewData qui n'est pas loin du seuil de signification.

Variables	Valeur de T	Seuil de Student	#Classes
HDiff	2,750444402	2,403	56
HEff	3,550267939	2,403	55
LCOM1	3,354216371	2,429	38
LCOM3	-0,110909932	2,438	35
LOC	4,085914192	2,403	55
NOA	1,351690671	2,403	55
NOCC	1	2,403	55
RFC	8,357147779	2,403	55
WMPC1	4,687598502	2,403	55
CBO	4,371696866	2,403	55
DOIH	0	2,403	56
NOO	5,364738715	2,403	55
MTMOOP	4,112915497	2,403	55
TNofCalled	2,651727	2,567	17
TNofNewData	2,354334	2,567	17
TNofAssert	3,194828	2,567	17
TNOO	3,306853	2,567	17
TLOC	2,950237	2,567	17
TRFC	5,013597	2,567	17
TINVOK	5,231249	2,567	17
TWMPC	3,306853	2,567	17

Tableau 10.5 : Résultats du test statistique de (J, AJ) Hotdraw

4.2.2.3 Discussion

En ce qui concerne Healthwather, nous ne pouvons pas trancher s'il y a une amélioration ou une détérioration de sa testabilité après l'aspectualisation du code des préoccupations transverses. L'étude des corrélations et le test statistique que nous avons effectués n'ont pas abouti à des résultats concluants.

Pour Telestrada, en effectuant notre test statistique t (voir les résultats dans le tableau 7), nous avons conclu globalement que les différences entre les valeurs des métriques respectives calculées sur les deux versions de Telestrada (OO et OA) sont positives pour toutes les métriques logicielles à part pour LCOM1 et WMPC, significatives pour la plupart des métriques logicielles et pour le modèle MTMOOP, et positives, mais non significatives du côté des métriques de testabilité même si les résultats sont très proches du seuil de signification. On peut alors, conclure que la testabilité des classes aspectualisées de la version OA de Telestrada est légèrement (amélioration, mais non significative) meilleure que dans le cas de la version OO. Mais, reste qu'il n'y a pas assez de classe aspectualisées pour pouvoir trancher.

Concernant Petstore, nous avons obtenu de très bonnes corrélations entre le modèle MTMOOP et les métriques de cas de test. En plus, notre test statistique a révélé une amélioration très significative du modèle MTMOOP et des métriques TNOO et TWMPC. En plus, il n'y a pas de détérioration significative du côté des classes tests. On peut alors, conclure qu'il existe une amélioration de la testabilité des classes OO en aspectualisant le code des préoccupations transverses.

Finalement, concernant les deux derniers systèmes (HSQLBD et JHotdraw), les corrélations obtenues sont excellentes. Le nombre des classes testées est satisfaisant et les résultats des tests statistiques effectués sont positifs et en majorité significatifs. Pour ces deux systèmes, on ne peut douter de l'amélioration significative de la testabilité des classes aspectualisées.

Nous pouvons raisonnablement rejeter l'hypothèse H_0 et accepter l'hypothèse H_1 selon laquelle la testabilité des classes OO est améliorée après aspectualisation du code des préoccupations transverses.

4.2.3 Conclusion

Cette section du chapitre a examiné, dans un premier temps, la relation entre le modèle de testabilité MTMOOP et notre suite de métriques de testabilité. Les corrélations entre le modèle MTMOOP et notre suite de métriques (TNofCalled, TNofNewData, TNofAssert, TNOO, TLOC, TRFC, TWMP1, TWMP2) sont fortes et significatives.

Puis, nous avons tracé les courbes de valeurs de chaque métrique afin de comparer leurs allures (évolution) et essayer de collecter quelques informations utiles pour notre étude. Nous avons remarqué que les courbes avaient la même tendance ou presque, pour chaque projet respectivement. C'est-à-dire, quand la valeur du modèle MTMOOP d'une classe est plus importante que celles des autres classes du même système, les valeurs obtenues par le calcul de notre suite de métriques sur la classe test correspondante à cette classe sont plus important que les valeurs obtenues pour les autres classes. Cette théorie a été vérifiée dans la majorité des cas. Mais pourquoi ne dit-on pas dans tout les cas? C'est parce que le modèle MTMOOP tient compte juste de l'encapsulation, de l'héritage et du couplage, mais ne tient pas compte de la cohésion et de la complexité des classes, etc. Dans le cas où ces deux facteurs sont influents dans un logiciel, cela peut fausser les prédictions du modèle MTMOOP si on veut comparer la testabilité des mêmes classes, dans de différentes versions de ce programme.

Enfin, nous avons fait un test statistique pour voir si les différences entre les mesures calculées sur les versions respectives de chaque programme sont significatives ou pas. Les résultats obtenus nous ont permis d'accepter l'hypothèse H_1 et de rejeter l'hypothèse H_0 . Alors, nous avons conclu que les classes OO sont améliorées en aspectualisant le code des préoccupations transverses.

Notre étude a impliqué seulement des systèmes logiciels écrits en Java. Tandis qu'il n'y a aucune raison de soupçonner que les résultats seraient différents avec des systèmes écrits dans d'autres langages orientés objet (comme C ++), il serait intéressant d'étudier des systèmes écrits dans d'autres langages. Nous espérons, cependant, que cette étude aidera à une meilleure compréhension de la contribution du paradigme orienté aspect à la testabilité des classes comportant des préoccupations transverses.

4.3 Une analyse empirique de la testabilité des programmes OO : tests générés automatiquement vs tests développés.

Dans cette section, nous allons présenter les résultats de l'étude empirique comparative réalisée sur des programmes OO dont les classes tests ont été développées par des développeurs versus ces mêmes programmes avec des classes tests générées automatiquement.

4.3.1 Cadre expérimental

Le but de cette étude est d'explorer empiriquement les différences (étude empirique comparative) au niveau de la testabilité des programmes OO en utilisant dans un premier temps les classes tests développées par des développeurs, puis dans un second temps des classes tests générées automatiquement.

Les classes tests développées manuellement sont disponibles en « *Open source* ». Ces classes ont été développées en utilisant JUnit [JUnit].

Les classes tests générées automatiquement ont été développées automatiquement grâce à CodePro [CodePro]. Nous avons essayé de générer le maximum de classes tests pour chaque système.

4.3.1.1 Collecte des données

Nous avons utilisé notre sélection de métriques logicielles pour quantifier les classes pour lesquelles des cas de test JUnit [JUnit] ont été développés, et les métriques de cas test pour quantifier les cas de test JUnit [JUnit].

Les métriques relatives au code source des programmes et les métriques de test ont été calculées en utilisant l'outil Borland Together [Together].

4.3.1.2 Hypothèses et analyse statistique

Nous présentons, dans cette section, la méthodologie de l'étude empirique que nous avons mené afin de comparer la qualité des classes tests qui ont été développées manuellement avec la qualité les classes tests qui ont été générées automatiquement.

Nous avons effectué des tests statistiques à l'aide des corrélations. Les hypothèses nulles et alternatives que nous avons vérifiées lors de nos expériences sont :

- H_0 : Les classes tests qui ont été développées par des développeurs reflètent plus la réalité de leurs classes logicielles respectives.
- H_1 : Les classes tests qui ont été générées automatiquement reflètent plus la réalité de leurs classes logicielles respectives.

Dans cette expérimentation, rejeter l'hypothèse nulle montre qu'il y a une plus forte relation statistiquement significative entre les classes logicielles et classes tests qui ont été générées automatiquement (niveau de signification choisi $\alpha = 0,05$). Pour l'analyse des données recueillies, nous avons préféré une mesure non paramétrique de la corrélation. Nous avons utilisé le coefficient de corrélation de Spearman. Cette technique, basée sur les rangs des observations, est largement utilisée pour mesurer le degré de linéarité des relations entre deux variables (deux ensembles de données classées). Elle mesure le degré de rapprochement des grappes de données classées, autour d'une ligne droite.

4.3.2 Étude de Cas

4.3.2.1 Systèmes sélectionnés

Les systèmes sélectionnés sont les suivants :

- ANT (www.apache.org) : Apache Ant est une bibliothèque Java et un outil de ligne de commande dont la mission est de conduire les processus décrits dans la construction des fichiers comme des cibles et des points d'extension qui dépendent les uns des autres. L'utilisation principale connue d'Ant est la construction d'applications Java. Ant fournit un certain nombre de tâches intégrées permettant de compiler, assembler, tester et exécuter des applications Java.
- DBUnit (<http://www.dbunit.org/>) : C'est une extension de JUnit (aussi utilisable avec Ant) destinée aux projets qui exploitent des bases de données. Cet outil permet de résoudre les problèmes qui peuvent survenir lorsqu'un test corrompt la base de données et entraîne ensuite un échec des tests unitaires suivant.
- JFreeChart (<http://www.jfree.org/jfreechart>) : C'est une bibliothèque de diagrammes gratuite pour la plate-forme Java.
- POI (<http://poi.apache.org/>) : (Poor Obfuscation Implementation) est un projet de l'*Apache Software Foundation* permettant de manipuler avec la technologie Java divers types de fichiers créés par Microsoft Office.

SYSTÈMES			CLASSES LOGICIELLES — CLASSES TESTS		
	#Classes	MLOC	#TClasses développées manuellement	#TClasses développées automatiquement	#TClasses intersection (manu \cap auto)
ANT	713	89.85	115	268	32
DBU	238	76	84	168	70
JFC	496	137.73	230	430	201
POI	1540	95	372	946	250

TABLEAU 11 : QUELSQUES CARACTÉRISTIQUE DES SYSTÈMES ET LEURS SUITES DE TESTS UTILISÉS.

Le tableau 1 récapitule quelques caractéristiques des systèmes analysés : Le nombre des classes logicielles de chaque système, la moyenne des lignes de code des classes logicielles, le nombre de classes test JUNIT développées manuellement pour chaque système, le nombre de classes test JUNIT qui ont été développées automatiquement pour chaque système, le nombre de classes test JUNIT développées automatiquement dont leurs équivalentes développées manuellement existe.

Les premières observations que nous pouvons déjà faire sont :

- Uniquement un sous-ensemble de classes logicielles a été testé avec JUNIT,
- Le pourcentage des classes évaluées varie d'un système à un autre, et
- Les classes logicielles pour lesquelles des scénarios de test JUNIT ont été développés sont, en général, de grandes classes complexes.

4.3.2.2 Résultats

Les tableaux 12, 13, 14 et 15 résument les résultats de l'analyse de corrélation. Les tableaux montrent, pour chacun des systèmes sélectionnés et entre chaque paire de mesures distinctes (MTMOOP, métrique (logicielles et de cas test)), les valeurs obtenues par le coefficient de corrélation de Spearman r_s . Les coefficients de corrélation de Spearman obtenus sont significatifs (à $\alpha = 0,05$) et ils sont mis en gras dans les tableaux. Cela signifie que pour les paires correspondantes des mesures, il existe une corrélation d'un niveau de confiance de 95 %.

Variables	Jeux de tests Manuels	Intersection jeux de tests Manuels	Intersection jeux de tests Auto	jeux de tests Auto
	MTMOOP	MTMOOP	MTMOOP	MTMOOP
HDiff	0,644	0,738	0,738	0,498
HEff	0,787	0,740	0,740	0,519
CBO	0,966	0,816	0,816	0,796
FO	0,921	0,776	0,776	0,703
LCOM1	0,527	0,640	0,640	0,182
LCOM3	0,509	-0,284	-0,284	-0,021
DIT	0,437	0,810	0,810	0,793
NOC	-0,145	0,588	0,588	0,234
LOC	0,815	0,444	0,444	0,345
NOA	0,635	0,520	0,520	0,285
NOO	0,655	0,703	0,703	0,446
RFC	0,875	0,857	0,857	0,782
WMPC1	0,758	0,817	0,817	0,548
TNofCalled	0,256	0,283	0,518	0,402
TNofNewData	0,144	0,453	0,590	0,429
TNofAssert	0,030	-0,053	0,364	0,125
TLOC	0,325	0,404	0,465	0,402
TNOO	0,403	-0,156	0,471	0,352
TRFC	0,507	0,308	0,502	0,435
TINVOK	0,470	0,451	0,463	0,513
TWMPG1	0,357	-0,250	0,471	0,360

TABLEAU 12 : VALLEURS DE CORRÉLATION ENTRE LE MTMOOP ET LES MÉTRIQUES SÉLECTIONNÉES POUR LE SYSTÈME ANT.

Variables	Jeux de tests Manuels	Intersection jeux de tests Manuels	Intersection Jeux de tests Auto	jeux de tests auto
	MTMOOP	MTMOOP	MTMOOP	MTMOOP
HDiff	0,489	0,551	0,551	0,662
HEff	0,637	0,674	0,674	0,720
CBO	0,959	0,952	0,952	0,904
FO	0,911	0,906	0,906	0,823
LCOM1	0,432	0,414	0,414	0,324
LCOM3	0,237	0,176	0,176	0,253
DOIH	0,367	0,368	0,368	0,421
NOCC	0,028	0,021	0,021	-0,272
NOCON	0,064	0,148	0,148	0,221
LOC	0,667	0,697	0,697	0,713
NOA	0,023	0,017	0,017	0,173
NOO	0,226	0,229	0,229	0,277
RFC	0,775	0,755	0,755	0,798
WMPC	0,542	0,564	0,564	0,569
TNofCalled	0,396	0,432	0,684	0,683
TNofNewData	0,341	0,427	0,716	0,621
TNofAssert	0,172	0,145	0,134	0,419
TLOC	0,353	0,394	0,714	0,667
TNOO	0,236	0,233	0,730	0,651
TRFC	0,450	0,471	0,719	0,685
TINVOK	0,462	0,501	0,318	0,431
TWMPC	0,249	0,249	0,730	0,651

TABLEAU 13 : VALLEURS DE CORRÉLATION ENTRE LE MTMOOP ET LES MÉTRIQUES SÉLECTIONNÉES POUR LE SYSTÈME DBUNIT.

Variables	jeux de tests Manuels	Intersection jeux de tests Manuels	Intersection jeux de tests Auto	jeux de tests Auto
	MTMOOP	MTMOOP	MTMOOP	MTMOOP
HDiff	0,675	0,676	0,676	0,781
HEff	0,757	0,758	0,758	0,815
CBO	0,989	0,989	0,989	0,947
FO	0,974	0,972	0,972	0,928
LCOM1	0,618	0,592	0,592	0,573
LCOM3	0,384	0,376	0,376	0,243
DOIH	0,635	0,619	0,619	0,681
NOCC	0,233	0,222	0,222	-0,268
NOCON	0,163	0,119	0,119	0,517
LOC	0,753	0,752	0,752	0,821
NOA	0,460	0,439	0,439	0,539
NOO	0,513	0,503	0,503	0,585
RFC	0,863	0,860	0,860	0,883
WMPC1	0,651	0,647	0,647	0,735
TNofCalled	0,250	0,246	0,805	0,825
TNofNewData	0,087	0,050	0,802	0,811
TNofAssert	0,236	0,262	0,630	0,721
TLOC	0,283	0,271	0,775	0,812
TNOO	0,259	0,269	0,689	0,775
TRFC	0,335	0,310	0,694	0,781
TINVOK	0,345	0,314	0,494	0,626
TWMPC1	0,265	0,277	0,689	0,775

TABLEAU 14 : VALEURS DE CORRÉLATION ENTRE LE MTMOOP ET LES MÉTRIQUES SÉLECTIONNÉES POUR LE SYSTÈME JFREECHART.

Variables	Jeux de tests Manuels	Intersection jeux de tests Manuels	Intersection jeux de tests Auto	jeux de tests Auto
	MTMOOP	MTMOOP	MTMOOP	MTMOOP
HDiff	0,395	0,314	0,314	0,578
HEff	0,479	0,361	0,361	0,581
CBO	0,924	0,879	0,879	0,881
FO	0,850	0,771	0,771	0,825
LCOM1	0,146	-0,006	-0,006	0,155
LCOM3	0,107	0,159	0,159	0,005
DOIH	0,155	0,259	0,259	0,374
NOCC	-0,071	-0,100	-0,100	-0,268
NOCON	0,237	0,225	0,225	0,307
LOC	0,476	0,385	0,385	0,526
NOA	0,280	0,277	0,277	0,221
NOO	0,246	0,141	0,141	0,327
RFC	0,609	0,506	0,506	0,635
WMPC1	0,401	0,268	0,268	0,479
TNofCalled	0,289	0,143	0,350	0,501
TNofNewData	0,263	0,184	0,502	0,532
TNofAssert	0,215	0,106	0,273	0,382
TLOC	0,215	0,114	-0,014	0,191
TNOO	0,183	0,084	0,309	0,489
TRFC	0,324	0,189	0,242	0,456
TINVOK	0,339	0,211	-0,025	0,192
TWMPC1	0,151	0,074	0,309	0,489

TABLEAU 15 : VALLEURS DE CORRÉLATION ENTRE LE MTMOOP ET LES MÉTRIQUES SÉLECTIONNÉES POUR LE SYSTÈME POI.

4.3.2.3 Discussion

La première observation globale que nous pouvons faire est que les résultats obtenus soutiennent l'idée que les classes tests qui ont été générées automatiquement reflètent plus la réalité de leurs classes logicielles respectives puisqu'il y a une relation statistiquement significative entre le modèle MTMOOP et les métriques sélectionnées (Métriques logicielles et métriques de cas de test) plus importantes du côté des classes tests développées automatiquement.

Les coefficients de corrélation de Spearman obtenus entre le modèle MTMOOP et les métriques de cas de test du côté des jeux de tests développés automatiquement sont globalement significatifs ($\alpha = 0,05$) pour les quatre systèmes sélectionnés (sauf pour le système DBUnit entre le modèle MTMOOP et la métrique TNofAssert et pour le système POI entre le modèle MTMOOP et les métriques TLOC et TINVOK). En plus, les coefficients de corrélation de Spearman obtenus entre le modèle MTMOOP et les métriques de cas de test du côté des jeux de tests développés automatiquement sont beaucoup plus élevés que les coefficients de corrélation de Spearman obtenus entre le modèle MTMOOP et les métriques de cas de test du côté des jeux de tests développés manuellement. Pour le système DBUnit, les corrélations entre le modèle MTMOOP et TNofAssert a légèrement baissé. Ainsi pour le système POI entre le modèle MTMOOP et les métriques TLOC et TINVOK. Malgré cela, nous pouvons raisonnablement rejeter l'hypothèse H0 et accepter l'hypothèse H1.

Les résultats obtenus sont basés sur le jeu de données que nous avons rassemblées des systèmes analysés. Comme mentionnés précédemment, nous avons utilisé seulement un sous-ensemble de classes et leurs scénarios de test JUnit correspondant (voir la Tableau 11) de chacun des systèmes analysés. Nous pensons que le nombre de classes analysées est assez important pour généraliser.

Cette étude prouve que les classes tests générées automatiquement sont plus complètes et reflète plus la réalité que les classes tests développées. CodePro [CodePro] génère les tests unitaires de toutes les méthodes d'une classe et pour tous les chemins possibles, par contre, un humain va faire parfois des choix concernant les chemins à tester, etc.

Cette étude montre aussi que les classes tests générées automatiquement sont des meilleurs cas d'étude, mais pas nécessairement meilleurs dans la vie réelle, puisque des classes tests développées pourront être aussi fonctionnelles que celles générées automatiquement, dépendamment de l'expérience du testeur évidemment. Il est aussi possible que le style de développement utilisé par les développeurs pour écrire les scénarios de test puisse affecter les résultats obtenus ou produire des résultats différents pour des applications spécifiques.

Il serait intéressant de reproduire cette étude en utilisant des systèmes pour lesquels des scénarios de test JUnit [JUnit] ont été développés pour un maximum de classes logicielles. L'étude devrait être reproduite sur un plus grand nombre de systèmes orientés objet pour augmenter la généralité des résultats.

4.3.3 Conclusion

Dans cette section, nous avons présenté les résultats d'une étude empirique comparative réalisée sur des programmes OO dont les classes tests ont été développées manuellement versus ces mêmes programmes où les classes tests ont été générées automatiquement.

La testabilité a été examinée du point de vue des tests unitaires tout en évaluant l'effort de test. Nous avons conçu une étude empirique en utilisant des données rassemblées de quatre systèmes logiciels Java [Java] « *Open-Source* » pour lesquels des scénarios de test JUnit existent déjà. Pour capturer la testabilité des classes, nous avons utilisé diverses métriques pour évaluer quantitativement les différentes caractéristiques des cas de test JUnit correspondants.

Pour évaluer la différence entre les classes tests développées et les classes tests générées automatiquement, nous avons comparé les tests statistiques utilisant la corrélation. Les résultats obtenus soutiennent l'idée que les classes tests générées automatiquement reflètent plus la réalité de leurs classes logicielles respectives que les classes tests écrites par des développeurs.

Cependant, l'étude réalisée devrait être reproduite en utilisant plusieurs autres systèmes où il y a plus de tests unitaires développés pour tirer des conclusions plus générales.

La contribution de cette section devrait être vue comme exploratoire et indicative plutôt que conclusive. De plus, sachant que la testabilité logicielle est affectée par différents facteurs, il serait intéressant d'étendre la suite de métriques de cas de test utilisées pour mieux refléter l'effort de test.

CHAPITRE 5

CONCLUSION

La testabilité est un attribut qualité très important qui nous permet d'évaluer l'effort de test à fournir pour valider un logiciel. Dans ce mémoire, nous avons essayé de prédire la testabilité en utilisant le modèle MTMOOP. Ces prédictions nous permettent non seulement de savoir si notre logiciel est facile à tester ou non (en termes d'effort), mais nous renseignent également, de façon globale, sur la qualité de notre système. Ceci nous permet d'agir, le cas échéant, pour l'améliorer. Ces prédictions, nous permettent également de réduire les coûts (en termes de ressources humaines et de temps).

Dans ce mémoire, nous avons réussi à valider trois hypothèses. La première hypothèse que nous avons réussi à valider est qu'il existe une corrélation significative entre le modèle MTMOOP et la testabilité. Cela signifie que le modèle MTMOOP prédit bien la testabilité d'un logiciel à partir des classes OO.

La deuxième hypothèse que nous avons pu valider est que la testabilité des classes OO est améliorée après aspectualisation du code des préoccupations transverses. Cette conclusion est logique puisque les classes aspectualisées contiennent moins de code et sont plus cohésives en apparence. Cette étude a permis de valider cela empiriquement.

La troisième hypothèse validée dans ce mémoire est que les classes tests générées automatiquement reflètent plus la réalité de leurs classes logicielles respectives. Cette hypothèse a été validée grâce à une étude empirique comparative réalisée sur des programmes OO dont les classes tests ont été développées versus ces mêmes programmes où les classes tests ont été générées automatiquement.

Nous comptons reprendre ces études et les étendre en intégrant les aspects dans notre processus d'évaluation. Ceci nous permettra de comparer la testabilité globale des systèmes OO versus les mêmes systèmes version OA.

ANNEXES

Variables	HDIT	HEIT	FO	LCOM1	LCOM3	LOC	NOA	NOCC	RFC	WMPC1	CBO	DOIH	NOO	MTMOOP	TNofCalled	TNofNewData	TNofAsset	TNOO	TLOC	TRFC	TINVOK	TWMPG1
HDIT	1	0,649	0,506	-0.259	-0.163	0,677	0,451	-0.110	0,664	0,804	0.127	-0.042	0,548	0,089	0,700	0,630	0,662	0,627	0,657	0,645	0,542	0,635
HEIT	0,649	1	0,507	-0.074	0,147	0,870	0,271	-0,056	0,683	0,788	0,285	-0,047	0,448	0,256	0,640	0,489	0,636	0,582	0,694	0,646	0,621	0,643
FO	0,506	0,507	1	-0.176	-0.067	0,659	0,312	-0.014	0,766	0,664	0,473	-0.118	0,626	0,427	0,608	0,499	0,504	0,710	0,687	0,736	0,627	0,726
LCOM1	-0.259	-0.074	-0.176	1	0,472	0,073	0,605	0,697	0,021	0,217	-0.142	-0.074	0,652	-0,191	-0,074	-0,115	0,032	-0,030	-0,059	-0,056	-0,088	-0,056
LCOM3	-0.163	0,147	-0.067	0,472	1	0,247	0,650	0,217	0,203	0,326	-0,044	0,185	0,528	-0,057	0,112	-0,049	0,238	0,076	0,094	0,078	0,070	0,076
LOC	0,677	0,870	0,659	0,073	0,247	1	0,447	-0,094	0,757	0,838	0,541	-0,077	0,601	0,505	0,745	0,592	0,677	0,784	0,817	0,796	0,649	0,797
NOA	0,451	0,271	0,312	0,605	0,650	0,447	1	-0,023	0,545	0,639	-0,016	-0,005	0,706	-0,062	0,490	0,465	0,470	0,427	0,354	0,371	0,196	0,358
NOCC	-0,110	-0,056	-0,014	0,697	0,217	-0,094	-0,023	1	-0,147	-0,066	-0,046	-0,264	-0,001	-0,088	-0,135	-0,113	-0,127	-0,125	-0,104	-0,098	-0,032	-0,096
RFC	0,664	0,683	0,766	0,021	0,203	0,757	0,545	-0,147	1	0,837	0,161	0,277	0,695	0,166	0,733	0,608	0,704	0,716	0,682	0,713	0,557	0,692
WMPC1	0,804	0,788	0,664	0,217	0,326	0,838	0,639	-0,066	0,837	1	0,167	-0,083	0,861	0,104	0,709	0,582	0,714	0,669	0,695	0,695	0,593	0,680
CBO	0,127	0,285	0,473	-0,142	-0,044	0,541	-0,016	-0,046	0,164	0,167	1	-0,224	0,082	0,986	0,368	0,289	0,124	0,574	0,508	0,493	0,248	0,522
DOIH	-0,042	-0,047	-0,118	-0,074	0,185	-0,077	-0,005	-0,264	0,277	-0,083	-0,224	1	-0,157	-0,068	0,066	-0,028	0,165	0,032	-0,048	-0,030	-0,131	-0,048
NOO	0,548	0,448	0,626	0,652	0,528	0,601	0,706	-0,001	0,695	0,861	0,082	-0,157	1	-0,003	0,561	0,446	0,601	0,550	0,525	0,549	0,432	0,532
MTMOOP	0,089	0,256	0,427	-0,191	-0,057	0,505	-0,063	-0,088	0,166	0,104	0,986	-0,068	-0,003	1	0,352	0,263	0,114	0,558	0,480	0,466	0,207	0,494
TNofCalled	0,700	0,640	0,608	-0,074	0,112	0,745	0,490	-0,135	0,733	0,709	0,368	0,066	0,561	0,352	1	0,875	0,904	0,896	0,914	0,899	0,715	0,892
TNofNewData	0,630	0,489	0,499	-0,115	-0,049	0,592	0,465	-0,113	0,608	0,582	0,289	-0,028	0,446	0,263	0,875	1	0,700	0,689	0,689	0,680	0,520	0,671
TNofAsset	0,662	0,636	0,504	0,032	0,238	0,677	0,470	-0,127	0,704	0,714	0,124	0,165	0,601	0,114	0,904	0,700	1	0,740	0,821	0,794	0,721	0,780
TNOO	0,627	0,582	0,710	-0,030	0,076	0,784	0,427	-0,125	0,716	0,669	0,574	0,032	0,550	0,558	0,896	0,689	0,740	1	0,944	0,957	0,681	0,959
TLOC	0,657	0,694	0,687	-0,059	0,094	0,817	0,354	-0,104	0,682	0,695	0,508	-0,048	0,525	0,480	0,914	0,689	0,821	0,944	1	0,990	0,863	0,992
TRFC	0,645	0,646	0,736	-0,056	0,078	0,796	0,371	-0,098	0,713	0,695	0,493	-0,030	0,549	0,466	0,899	0,680	0,794	0,957	0,990	1	0,864	0,999
TINVOK	0,542	0,621	0,627	-0,088	0,070	0,649	0,196	-0,032	0,557	0,593	0,248	-0,131	0,432	0,207	0,715	0,520	0,721	0,681	0,863	0,864	1	0,857
TWMPG1	0,635	0,643	0,726	-0,056	0,076	0,797	0,358	-0,096	0,692	0,680	0,522	-0,048	0,532	0,494	0,892	0,671	0,780	0,959	0,992	0,999	0,857	1

Tableau 16.1 : Matrice de corrélation (Pearson) de Healthwatcher Java (Toutes les classes)

Variables	HDiff	HEff	FO	LCOM1	LCOM3	LOC	NOA	NOCC	RFC	WMPC1	CBO	DOIH	NOO	MTMOOP	TNofCalled	TNofNewData	TNofAssert	TNOO	TLOC	TRFC	TINVOK	TWMPG1
HDiff	1	0,985	0,508	-0,254	-0,381	0,550	0,783	-0,405	0,822	0,738	-0,048	0,271	0,615	-0,056	0,680	0,638	0,849	0,660	0,527	0,591	0,334	0,543
HEff	0,985	1	0,512	-0,045	-0,208	0,574	0,796	-0,396	0,849	0,754	-0,042	0,277	0,638	-0,058	0,688	0,631	0,873	0,670	0,537	0,599	0,354	0,551
FO	0,508	0,512	1	-0,298	-0,318	0,387	0,503	0,289	0,623	0,738	0,322	-0,314	0,752	0,246	0,355	0,339	0,419	0,325	0,504	0,570	0,641	0,535
LCOM1	-0,254	-0,045	-0,298	1	0,593	0,229	0,438	0,397	0,249	0,291	-0,249	0,099	0,458	-0,336	0,074	-0,070	0,271	0,081	0,059	0,069	0,102	0,075
LCOM3	-0,381	-0,208	-0,318	0,593	1	0,109	0,779	0,266	0,103	0,278	-0,294	0,321	0,476	-0,320	0,013	-0,111	0,203	-0,078	-0,079	-0,082	-0,035	-0,085
LOC	0,550	0,574	0,387	0,229	0,109	1	0,581	-0,332	0,390	0,456	0,641	-0,064	0,508	0,531	0,784	0,697	0,450	0,861	0,868	0,851	0,045	0,872
NOA	0,783	0,796	0,503	0,438	0,779	0,581	1	-0,238	0,681	0,709	0,058	0,102	0,711	-0,017	0,695	0,678	0,738	0,625	0,579	0,612	0,262	0,574
NOCC	-0,405	-0,396	0,289	0,297	0,266	-0,332	-0,238	1	-0,204	0,099	-0,066	-0,680	0,221	-0,204	-0,532	-0,499	-0,453	-0,513	-0,233	-0,190	0,358	-0,196
RFC	0,822	0,849	0,623	0,249	0,103	0,390	0,681	-0,204	1	0,855	-0,164	0,346	0,722	-0,147	0,554	0,468	0,885	0,465	0,423	0,510	0,569	0,429
WMPC1	0,738	0,754	0,738	0,291	0,278	0,456	0,709	0,099	0,855	1	-0,096	-0,074	0,945	-0,205	0,459	0,373	0,718	0,415	0,506	0,603	0,671	0,531
CBO	-0,048	-0,042	0,322	-0,249	-0,294	0,641	0,058	-0,066	-0,164	-0,096	1	-0,300	0,035	0,946	0,406	0,389	-0,158	0,513	0,618	0,545	-0,139	0,621
DOIH	0,271	0,277	-0,314	0,099	0,321	-0,064	0,102	-0,680	0,346	-0,074	-0,300	1	-0,269	-0,060	0,287	0,233	0,488	0,160	-0,094	-0,091	-0,146	-0,148
NOO	0,615	0,635	0,752	0,458	0,476	0,508	0,711	0,221	0,722	0,945	0,035	-0,269	1	-0,133	0,416	0,351	0,572	0,397	0,546	0,624	0,596	0,576
MTMOOP	-0,056	-0,058	0,246	-0,336	-0,320	0,531	-0,017	-0,204	-0,147	-0,205	0,946	-0,060	-0,133	1	0,393	0,380	-0,116	0,461	0,515	0,439	-0,167	0,502
TNofCalled	0,680	0,688	0,355	0,074	0,011	0,784	0,695	-0,532	0,554	0,459	0,406	0,287	0,416	0,393	1	0,945	0,725	0,943	0,876	0,855	0,101	0,839
TNofNewData	0,638	0,631	0,339	-0,070	-0,111	0,697	0,678	-0,499	0,468	0,373	0,389	0,233	0,351	0,380	0,945	1	0,642	0,875	0,805	0,770	0,043	0,765
TNofAssert	0,849	0,873	0,419	0,271	0,203	0,450	0,738	-0,453	0,885	0,718	-0,158	0,488	0,572	-0,116	0,725	0,642	1	0,601	0,487	0,546	0,454	0,460
TNOO	0,660	0,670	0,325	0,081	-0,078	0,861	0,625	-0,513	0,465	0,415	0,513	0,160	0,397	0,461	0,943	0,875	0,601	1	0,912	0,889	0,012	0,906
TLOC	0,527	0,537	0,504	0,059	-0,079	0,868	0,579	-0,233	0,423	0,506	0,618	-0,094	0,546	0,515	0,876	0,805	0,487	0,912	1	0,982	0,214	0,991
TRFC	0,591	0,599	0,570	0,069	-0,082	0,851	0,612	-0,190	0,510	0,603	0,545	-0,091	0,624	0,439	0,855	0,770	0,546	0,889	0,982	1	0,311	0,985
TINVOK	0,334	0,354	0,641	0,102	-0,035	0,045	0,262	0,358	0,569	0,671	-0,139	-0,146	0,596	-0,167	0,101	0,043	0,454	0,012	0,214	0,311	1	0,216
TWMPG1	0,543	0,551	0,535	0,075	-0,085	0,872	0,574	-0,196	0,429	0,531	0,621	-0,148	0,576	0,502	0,839	0,765	0,460	0,906	0,991	0,985	0,216	1

Tableau 16.2 : Matrice de corrélation (Spearman) de Healthwatcher Java (Toutes les classes)

Variables	HDiff	HET	FO	LCOM1	LCOM3	LOC	NOA	NOCC	RFC	WMPC1	CBO	DOIH	NOO	MTMOOP	TNoFCalled	TNoFNewData	TNoFAssert	TNOO	TLOC	TRFC	TINVOK	TWMPG1
HDiff	1	0,675	0,396	-0,249	-0,101	0,779	0,332	-0,164	0,472	0,760	0,412	-0,091	0,388	0,323	0,636	0,661	0,382	0,658	0,654	0,658	0,080	0,658
HET	0,675	1	0,429	-0,063	0,168	0,938	0,254	-0,066	0,497	0,776	0,391	-0,112	0,409	0,298	0,631	0,618	0,341	0,665	0,696	0,652	-0,019	0,665
FO	0,396	0,429	1	-0,216	-0,144	0,530	0,010	-0,083	0,735	0,322	0,962	0,131	0,199	0,878	0,381	0,474	-0,100	0,697	0,698	0,676	-0,073	0,697
LCOM1	-0,249	-0,063	-0,216	1	0,518	0,046	0,574	0,578	-0,006	0,267	-0,241	-0,150	0,735	-0,299	-0,093	-0,173	0,137	-0,093	-0,138	-0,093	0,050	-0,093
LCOM3	-0,101	0,168	-0,144	0,518	1	0,236	0,680	0,216	0,121	0,373	-0,184	-0,092	0,582	-0,233	0,053	-0,015	0,277	-0,043	-0,065	-0,041	0,082	-0,043
LOC	0,779	0,938	0,530	0,046	0,236	1	0,432	-0,115	0,575	0,867	0,494	-0,142	0,548	0,374	0,744	0,738	0,354	0,812	0,823	0,800	0,009	0,812
NOA	0,332	0,254	0,010	0,574	0,680	0,432	1	-0,025	0,118	0,590	-0,035	-0,293	0,667	-0,142	0,313	0,371	0,175	0,320	0,268	0,303	-0,093	0,320
NOCC	-0,164	-0,066	-0,083	0,578	0,216	-0,115	-0,025	1	-0,088	-0,096	-0,105	-0,070	-0,019	-0,109	-0,121	-0,133	-0,075	-0,177	-0,190	-0,185	-0,079	-0,177
RFC	0,472	0,497	0,735	-0,006	0,121	0,575	0,118	-0,088	1	0,319	0,826	0,575	0,050	0,872	0,338	0,452	-0,005	0,707	0,716	0,694	-0,013	0,707
WMPC1	0,760	0,776	0,322	0,267	0,373	0,867	0,590	-0,096	0,319	1	0,242	-0,410	0,816	0,068	0,671	0,595	0,474	0,672	0,668	0,678	0,133	0,672
CBO	0,412	0,391	0,962	-0,241	-0,184	0,494	-0,035	-0,105	0,826	0,242	1	0,308	0,064	0,963	0,327	0,447	-0,129	0,755	0,755	0,733	-0,078	0,755
DOIH	-0,091	-0,112	0,131	-0,150	-0,092	-0,142	-0,293	-0,070	0,575	-0,410	0,308	1	-0,585	0,550	-0,205	-0,098	-0,160	-0,182	-0,160	-0,181	-0,011	-0,182
NOO	0,388	0,409	0,199	0,735	0,582	0,548	0,667	-0,019	0,050	0,816	0,064	-0,585	1	-0,144	0,429	0,391	0,313	0,441	0,398	0,431	-0,025	0,441
MTMOOP	0,323	0,298	0,878	-0,299	-0,233	0,374	-0,142	-0,109	0,872	0,068	0,963	0,550	-0,144	1	0,219	0,355	-0,174	0,739	0,749	0,717	-0,086	0,739
TNoFCalled	0,636	0,631	0,381	-0,093	0,053	0,744	0,313	-0,121	0,338	0,671	0,327	-0,205	0,429	0,219	1	0,833	0,713	0,888	0,919	0,926	0,417	0,888
TNoFNewData	0,661	0,618	0,474	-0,173	-0,015	0,738	0,371	-0,133	0,452	0,595	0,447	-0,098	0,391	0,355	0,833	1	0,363	0,842	0,836	0,824	-0,033	0,842
TNoFAssert	0,382	0,341	-0,100	0,137	0,277	0,354	0,175	-0,075	-0,005	0,474	-0,129	-0,160	0,313	-0,174	0,713	0,363	1	0,355	0,404	0,442	0,739	0,355
TNOO	0,658	0,665	0,697	-0,093	-0,043	0,812	0,320	-0,177	0,707	0,672	0,755	-0,182	0,441	0,739	0,888	0,842	0,355	1	0,990	0,992	0,066	1,000
TLOC	0,654	0,696	0,698	-0,138	-0,065	0,823	0,268	-0,190	0,716	0,668	0,755	-0,160	0,398	0,749	0,919	0,836	0,404	0,990	1	0,992	0,151	0,990
TRFC	0,658	0,652	0,676	-0,093	-0,041	0,800	0,303	-0,185	0,694	0,678	0,733	-0,181	0,431	0,717	0,926	0,824	0,442	0,992	0,992	1	0,190	0,992
TINVOK	0,080	-0,019	-0,073	0,050	0,082	0,009	-0,093	-0,079	-0,013	0,133	-0,078	-0,011	-0,025	-0,086	0,417	-0,033	0,739	0,066	0,151	0,190	1	0,066
TWMPG1	0,658	0,665	0,697	-0,093	-0,043	0,812	0,320	-0,177	0,707	0,672	0,755	-0,182	0,441	0,739	0,888	0,842	0,355	1,000	0,990	0,992	0,066	1

Tableau 16.3 : Matrice de corrélation (Pearson) de Healthwatcher AspectJ (Toutes les classes)

Variables	HDif	HEff	FO	L.COM1	L.COM2	LOC	NOA	NOCC	RFC	WMPC1	CBO	DOIH	NOO	MTMOOP	TNoCalled	TNoNewData	TNoAsset	TNOO	TLOC	TRFC	TINVOK	TWMPC1
HDif	1	0,966	0,434	-0,267	-0,344	0,829	0,465	-0,571	0,684	0,496	0,521	0,250	0,306	0,473	0,670	0,624	0,173	0,755	0,742	0,758	0,032	0,755
HEff	0,966	1	0,507	-0,079	-0,219	0,894	0,436	-0,563	0,764	0,482	0,594	0,294	0,305	0,535	0,700	0,640	0,132	0,786	0,773	0,791	0,032	0,786
FO	0,434	0,507	1	-0,243	-0,374	0,553	0,010	-0,022	0,651	0,229	0,977	0,142	0,243	0,907	0,450	0,513	-0,363	0,644	0,651	0,623	-0,201	0,644
L.COM1	-0,267	-0,079	-0,243	1	0,640	0,247	0,426	0,284	0,164	0,361	-0,233	0,019	0,515	-0,318	0,089	-0,121	0,405	0,055	-0,024	0,068	0,180	0,055
L.COM2	-0,344	-0,219	-0,374	0,640	1	0,134	0,798	0,264	-0,001	0,366	-0,372	0,073	0,550	-0,406	-0,010	-0,139	0,498	-0,186	-0,216	-0,175	0,052	-0,186
LOC	0,829	0,894	0,553	0,247	0,134	1	0,562	-0,371	0,649	0,735	0,589	0,018	0,617	0,445	0,810	0,695	0,266	0,917	0,900	0,916	-0,040	0,917
NOA	0,465	0,436	0,010	0,426	0,798	0,562	1	-0,229	0,149	0,627	0,021	-0,250	0,658	-0,079	0,624	0,541	0,600	0,671	0,662	0,667	-0,047	0,671
NOCC	-0,571	-0,563	-0,022	0,284	0,264	-0,371	-0,229	1	-0,512	0,008	-0,090	-0,558	0,187	-0,212	-0,294	-0,230	-0,187	-0,146	-0,159	-0,166	-0,182	-0,146
RFC	0,684	0,764	0,651	0,164	-0,001	0,649	0,149	-0,512	1	0,120	0,730	0,662	0,017	0,765	0,491	0,518	-0,205	0,679	0,686	0,685	-0,025	0,679
WMPC1	0,496	0,482	0,229	0,361	0,366	0,735	0,627	0,008	0,120	1	0,207	-0,512	0,902	0,002	0,674	0,457	0,552	0,888	0,876	0,887	-0,063	0,888
CBO	0,521	0,594	0,977	-0,233	-0,372	0,589	0,021	-0,090	0,730	0,207	1	0,241	0,196	0,937	0,456	0,531	-0,391	0,700	0,700	0,681	-0,204	0,700
DOIH	0,250	0,294	0,142	0,019	0,073	0,018	0,250	-0,558	0,662	-0,512	0,241	1	0,640	0,461	-0,036	0,075	-0,331	-0,249	-0,216	-0,221	0,201	-0,249
NOO	0,306	0,305	0,243	0,515	0,550	0,617	0,658	0,187	0,017	0,902	0,196	0,640	1	-0,045	0,595	0,434	0,464	0,758	0,742	0,745	-0,139	0,758
MTMOOP	0,473	0,535	0,907	-0,318	-0,406	0,445	-0,079	-0,212	0,765	0,002	0,937	0,461	-0,045	1	0,328	0,422	-0,423	0,450	0,480	0,442	-0,164	0,450
TNoCalled	0,670	0,700	0,450	0,089	-0,010	0,810	0,624	-0,294	0,491	0,674	0,456	-0,036	0,595	0,328	1	0,880	0,491	0,936	0,957	0,941	0,087	0,936
TNoNewData	0,624	0,640	0,513	-0,121	-0,139	0,695	0,541	-0,230	0,518	0,457	0,531	0,075	0,434	0,422	0,880	1	0,259	0,788	0,820	0,789	0,041	0,788
TNoAsset	0,173	0,132	-0,363	0,405	0,498	0,266	0,600	-0,187	-0,205	0,552	-0,391	-0,331	0,464	-0,423	0,491	0,259	1	0,497	0,538	0,518	0,262	0,497
TNOO	0,755	0,786	0,644	0,055	-0,186	0,917	0,671	-0,146	0,679	0,888	0,700	-0,249	0,758	0,450	0,936	0,788	0,497	1	0,988	0,996	-0,020	1,000
TLOC	0,742	0,773	0,651	-0,024	-0,216	0,900	0,662	-0,159	0,686	0,876	0,700	-0,216	0,742	0,480	0,957	0,820	0,538	0,988	1	0,986	-0,007	0,988
TRFC	0,758	0,791	0,623	0,068	-0,175	0,916	0,667	-0,166	0,685	0,887	0,681	-0,221	0,745	0,442	0,941	0,789	0,518	0,996	0,986	1	0,052	0,996
TINVOK	0,032	0,032	-0,201	0,180	0,052	-0,040	-0,047	-0,182	-0,025	-0,003	-0,204	0,201	-0,139	-0,164	0,087	0,041	0,262	-0,020	-0,007	0,052	1	-0,020
TWMPC1	0,755	0,786	0,644	0,055	-0,186	0,917	0,671	-0,146	0,679	0,888	0,700	-0,249	0,758	0,450	0,936	0,788	0,497	1,000	0,988	0,996	-0,020	1

Tableau 16.4 : Matrice de corrélation (Spearman) de Healthwatcher AspectJ (Toutes les classes)

Variables	HDiff	HEff	FO	LCOM1	LCOM3	LOC	NOA	NOCC	RFC	WMPC1	CBO	DOIH	NOO	MTMOOP	TnofCalled	TnofNewData	TnofAsset	TNOO	TLOC	TRFC	TINVOK	TWMPC1
HDiff	1	0,753	0,272	-0,029	0,268	0,790	0,785	-0,244	0,466	0,874	0,350	-0,148	0,584	0,273	0,715	0,576	0,854	0,770	0,786	0,797	0,715	0,770
HEff	0,753	1	0,329	0,143	-0,125	0,902	0,593	-0,148	0,514	0,754	0,414	-0,078	0,528	0,371	0,871	0,764	0,860	0,880	0,886	0,868	0,431	0,880
FO	0,272	0,329	1	-0,035	-0,759	0,492	0,105	-0,002	0,239	0,311	0,977	-0,280	0,339	0,884	0,508	0,618	0,342	0,427	0,454	0,430	0,286	0,427
LCOM1	-0,029	0,143	-0,035	1	0,420	0,369	0,487	0,208	0,427	0,430	-0,060	0,410	0,789	-0,075	0,415	0,003	0,196	0,443	0,419	0,438	0,069	0,443
LCOM3	0,268	-0,125	-0,759	0,420	1	-0,090	0,802	0,221	-0,122	0,527	-0,738	0,253	0,546	-0,745	-0,091	-0,457	0,145	0,031	0,013	0,071	0,507	0,031
LOC	0,790	0,902	0,492	0,369	-0,090	1	0,765	-0,181	0,602	0,877	0,577	-0,135	0,763	0,505	0,935	0,774	0,914	0,970	0,971	0,971	0,600	0,970
NOA	0,785	0,593	0,105	0,487	0,802	0,765	1	-0,172	0,435	0,889	0,156	-0,110	0,783	0,078	0,670	0,387	0,823	0,769	0,763	0,797	0,722	0,769
NOCC	-0,244	-0,148	-0,002	0,208	0,221	-0,181	-0,172	1	-0,170	-0,103	-0,064	-0,073	0,016	-0,096	-0,158	-0,149	-0,187	-0,167	-0,172	-0,178	-0,199	-0,167
RFC	0,466	0,514	0,239	0,427	-0,122	0,602	0,435	-0,170	1	0,475	0,328	0,642	0,399	0,576	0,566	0,474	0,538	0,579	0,580	0,575	0,324	0,579
WMPC1	0,874	0,754	0,311	0,430	0,527	0,877	0,889	-0,103	0,475	1	0,379	-0,214	0,886	0,262	0,790	0,586	0,917	0,856	0,862	0,880	0,750	0,856
CIO	0,350	0,414	0,977	-0,060	-0,738	0,577	0,156	-0,064	0,328	0,379	1	-0,250	0,386	0,917	0,574	0,669	0,427	0,505	0,527	0,508	0,336	0,505
DOIH	-0,148	-0,078	-0,280	0,410	0,253	-0,135	-0,110	-0,073	0,642	-0,214	-0,250	1	-0,251	0,153	-0,119	-0,125	-0,133	-0,121	-0,128	-0,139	-0,240	-0,121
NOO	0,584	0,528	0,339	0,789	0,546	0,763	0,783	0,016	0,399	0,886	0,386	-0,251	1	0,249	0,664	0,424	0,730	0,734	0,728	0,755	0,644	0,734
MTMOOP	0,273	0,371	0,884	-0,075	-0,745	0,505	0,078	-0,096	0,576	0,262	0,917	0,153	0,249	1	0,513	0,623	0,352	0,437	0,458	0,432	0,219	0,437
TnofCalled	0,715	0,871	0,508	0,415	-0,091	0,935	0,670	-0,158	0,566	0,790	0,574	-0,119	0,664	0,513	1	0,890	0,840	0,967	0,979	0,957	0,504	0,967
TnofNewData	0,576	0,764	0,618	0,003	-0,457	0,774	0,387	-0,149	0,474	0,586	0,669	-0,125	0,424	0,623	0,890	1	0,667	0,790	0,823	0,776	0,364	0,790
TnofAsset	0,854	0,860	0,342	0,196	0,145	0,914	0,823	-0,187	0,538	0,917	0,427	-0,133	0,730	0,352	0,840	0,667	1	0,882	0,896	0,899	0,700	0,882
TNOO	0,770	0,880	0,427	0,443	0,031	0,970	0,769	-0,167	0,579	0,856	0,505	-0,121	0,734	0,437	0,967	0,790	0,882	1	0,996	0,996	0,578	1,000
TLOC	0,786	0,886	0,454	0,419	0,013	0,971	0,763	-0,172	0,580	0,862	0,527	-0,128	0,728	0,458	0,979	0,823	0,896	0,996	1	0,993	0,586	0,996
TRFC	0,797	0,868	0,430	0,438	0,071	0,971	0,797	-0,178	0,575	0,880	0,508	-0,139	0,755	0,432	0,957	0,776	0,899	0,996	0,993	1	0,649	0,996
TINVOK	0,715	0,431	0,286	0,069	0,507	0,600	0,722	-0,199	0,324	0,750	0,336	-0,240	0,644	0,219	0,504	0,364	0,700	0,578	0,586	0,649	1	0,578
TWMPC1	0,770	0,880	0,427	0,443	0,031	0,970	0,769	-0,167	0,579	0,856	0,505	-0,121	0,734	0,437	0,967	0,790	0,882	1,000	0,996	0,996	0,578	1

Tableau 17.1 : Matrice de corrélation (Pearson) de Telestrada Java (Toutes les classes)

Variables	HDiff	HExt	FO	LCOM1	LCOM3	LOC	NOA	NOCC	RFC	WMPC1	CBO	DOIH	NOO	MTMOOP	TnoCalled	TnoNewData	TnoAssert	TNOO	TLOC	TRFC	TINVOK	TWMPC1
HDiff	1	0,997	0,607	-0,187	-0,147	0,861	0,813	-0,377	0,457	0,810	0,686	-0,015	0,749	0,309	0,912	0,823	0,926	0,923	0,914	0,927	0,876	0,923
HExt	0,997	1	0,616	-0,224	-0,097	0,867	0,812	-0,375	0,472	0,814	0,692	-0,013	0,756	0,326	0,920	0,835	0,931	0,929	0,921	0,933	0,876	0,929
FO	0,607	0,616	1	0,006	-0,337	0,710	0,394	0,028	0,092	0,683	0,983	-0,312	0,666	0,575	0,705	0,673	0,665	0,693	0,697	0,688	0,629	0,693
LCOM1	-0,187	-0,224	0,006	1	0,886	0,031	0,216	0,369	0,203	0,224	-0,076	0,161	0,450	-0,082	-0,023	-0,138	0,021	0,033	0,016	0,075	0,135	0,033
LCOM3	-0,147	-0,097	-0,337	0,886	1	0,110	0,815	0,313	0,164	0,256	-0,410	0,418	0,802	-0,443	-0,092	-0,436	0,091	0,177	0,124	0,314	0,435	0,177
LOC	0,861	0,867	0,710	0,031	0,110	1	0,739	-0,135	0,345	0,968	0,768	-0,205	0,942	0,288	0,894	0,835	0,887	0,896	0,894	0,897	0,847	0,896
NOA	0,813	0,812	0,394	0,216	0,815	0,739	1	-0,324	0,449	0,722	0,466	0,016	0,655	0,187	0,788	0,812	0,796	0,796	0,793	0,797	0,703	0,796
NOCC	-0,377	-0,375	0,028	0,369	0,313	-0,135	-0,324	1	-0,124	-0,038	-0,010	-0,192	0,019	-0,044	-0,236	-0,184	-0,263	-0,249	-0,237	-0,256	-0,223	-0,249
RFC	0,457	0,472	0,092	0,203	0,164	0,345	0,449	-0,124	1	0,289	0,144	0,749	0,244	0,548	0,389	0,397	0,394	0,390	0,388	0,395	0,320	0,390
WMPC1	0,810	0,814	0,683	0,224	0,256	0,968	0,722	-0,038	0,289	1	0,739	0,298	0,985	0,183	0,862	0,814	0,856	0,866	0,864	0,866	0,820	0,866
CBO	0,686	0,692	0,983	-0,076	-0,410	0,768	0,466	-0,010	0,144	0,739	1	-0,301	0,714	0,563	0,767	0,730	0,735	0,758	0,761	0,752	0,693	0,758
DOIH	-0,015	-0,013	-0,312	0,161	0,418	-0,205	0,016	-0,192	0,749	-0,298	-0,301	1	-0,331	0,471	-0,127	-0,134	-0,103	-0,121	-0,127	-0,114	-0,129	-0,121
NOO	0,749	0,756	0,666	0,450	0,802	0,942	0,655	0,019	0,244	0,985	0,714	-0,331	1	0,141	0,817	0,760	0,811	0,822	0,820	0,821	0,789	0,822
MTMOOP	0,309	0,326	0,575	-0,082	-0,443	0,288	0,187	-0,044	0,548	0,183	0,563	0,471	0,141	1	0,349	0,388	0,322	0,330	0,338	0,328	0,248	0,330
TnoCalled	0,912	0,920	0,705	-0,023	-0,092	0,894	0,788	-0,236	0,389	0,862	0,767	-0,127	0,817	0,349	1	0,947	0,981	0,998	0,999	0,997	0,944	0,998
TnoNewData	0,823	0,835	0,673	-0,138	-0,436	0,835	0,812	-0,184	0,397	0,814	0,730	-0,134	0,760	0,388	0,947	1	0,904	0,936	0,946	0,928	0,820	0,936
TnoAssert	0,926	0,931	0,665	0,021	0,091	0,887	0,796	-0,263	0,394	0,856	0,735	-0,103	0,811	0,322	0,981	0,904	1	0,982	0,982	0,987	0,969	0,982
TNOO	0,923	0,929	0,693	0,033	0,177	0,896	0,796	-0,249	0,390	0,866	0,758	-0,121	0,821	0,330	0,998	0,936	0,982	1	0,999	0,998	0,947	1,000
TLOC	0,914	0,921	0,697	0,016	0,124	0,894	0,793	-0,237	0,388	0,864	0,761	-0,127	0,820	0,339	0,999	0,946	0,982	0,999	1	0,997	0,946	0,999
TRFC	0,927	0,933	0,688	0,075	0,314	0,897	0,797	-0,256	0,395	0,866	0,752	-0,114	0,821	0,328	0,997	0,928	0,987	0,998	0,997	1	0,957	0,998
TINVOK	0,876	0,876	0,629	0,135	0,435	0,847	0,703	-0,223	0,320	0,820	0,693	-0,129	0,789	0,248	0,944	0,820	0,969	0,947	0,946	0,957	1	0,947
TWMPC1	0,923	0,929	0,693	0,033	0,177	0,896	0,796	-0,249	0,390	0,866	0,758	-0,121	0,822	0,330	0,998	0,936	0,982	1,000	0,999	0,998	0,947	1

Tableau 17.2 : Matrice de corrélation (Spearman) de Telestrada Java (Toutes les classes)

Variables	HDifT	HEiT	FO	I.COM1	I.COM3	LOC	NOA	NOCC	RFC	WMPC1	CBO	DOIH	NOO	MTMOOP	TNofCalled	TNofNewData	TNofAsset	TNOO	TLOC	TRFC	TINVOK	TWMPC1
HDifT	1	0,845	0,203	0,018	0,356	0,834	0,764	-0,233	0,424	0,865	0,287	-0,143	0,570	0,190	0,784	0,640	0,877	0,843	0,847	0,861	0,750	0,843
HEiT	0,845	1	0,232	0,357	0,135	0,884	0,679	-0,156	0,467	0,825	0,282	-0,084	0,548	0,216	0,648	0,703	0,839	0,872	0,863	0,853	0,819	0,872
FO	0,203	0,232	1	0,123	-0,754	0,400	0,011	0,041	0,156	0,308	0,972	-0,296	0,534	0,838	0,488	0,650	0,274	0,392	0,405	0,373	0,156	0,392
I.COM1	0,018	0,357	0,123	1	0,282	0,634	0,456	0,135	0,688	0,535	0,153	0,454	0,791	0,139	0,471	0,132	0,038	0,473	0,413	0,435	-0,062	0,473
I.COM3	0,356	0,135	-0,754	0,282	1	0,281	0,887	0,215	0,094	0,595	-0,715	0,248	0,645	-0,726	-0,090	-0,583	0,294	0,209	0,168	0,300	0,566	0,209
LOC	0,834	0,884	0,400	0,634	0,281	1	0,823	-0,178	0,553	0,944	0,468	-0,147	0,807	0,360	0,959	0,805	0,894	0,975	0,967	0,968	0,679	0,975
NOA	0,764	0,679	0,011	0,456	0,887	0,823	1	-0,150	0,398	0,875	0,065	-0,097	0,760	-0,029	0,727	0,470	0,795	0,796	0,785	0,818	0,738	0,796
NOCC	-0,233	-0,156	0,041	0,135	0,215	-0,178	-0,150	1	-0,157	-0,097	-0,018	-0,076	0,025	-0,058	-0,235	-0,231	-0,236	-0,238	-0,243	-0,255	-0,293	-0,238
RFC	0,424	0,467	0,156	0,688	0,094	0,553	0,398	-0,157	1	0,452	0,235	0,680	0,369	0,559	0,568	0,498	0,489	0,564	0,558	0,559	0,381	0,564
WMPC1	0,865	0,815	0,308	0,535	0,595	0,944	0,875	-0,097	0,452	1	0,279	-0,211	0,885	0,232	0,866	0,676	0,893	0,914	0,905	0,922	0,736	0,914
CBO	0,287	0,282	0,972	0,153	-0,715	0,468	0,065	-0,018	0,235	0,379	1	-0,279	0,400	0,871	0,538	0,665	0,349	0,465	0,471	0,449	0,237	0,465
DOIH	-0,143	-0,084	-0,296	0,454	0,248	-0,147	-0,097	-0,076	0,680	-0,211	-0,279	1	-0,253	0,225	-0,122	-0,126	-0,124	-0,124	-0,128	-0,133	-0,153	-0,124
NOO	0,570	0,548	0,334	0,791	0,645	0,807	0,760	0,025	0,369	0,885	0,400	-0,253	1	0,226	0,712	0,516	0,682	0,753	0,739	0,758	0,598	0,753
MTMOOP	0,190	0,216	0,838	0,139	-0,726	0,360	-0,029	-0,058	0,559	0,232	0,871	0,225	0,226	1	0,452	0,493	0,257	0,372	0,377	0,351	0,132	0,372
TNofCalled	0,784	0,848	0,488	0,471	-0,090	0,959	0,727	-0,235	0,568	0,866	0,538	-0,122	0,712	0,452	1	0,923	0,891	0,983	0,986	0,972	0,654	0,983
TNofNewData	0,640	0,703	0,650	0,132	-0,583	0,805	0,470	-0,231	0,498	0,676	0,665	-0,126	0,516	0,593	0,923	1	0,763	0,857	0,875	0,839	0,510	0,857
TNofAsset	0,877	0,839	0,274	0,038	0,294	0,894	0,795	-0,236	0,489	0,893	0,349	-0,124	0,682	0,257	0,891	0,763	1	0,930	0,938	0,942	0,772	0,930
TNOO	0,843	0,872	0,392	0,473	0,209	0,975	0,796	-0,238	0,564	0,914	0,465	-0,124	0,758	0,351	0,972	0,839	0,942	0,995	0,993	1	0,776	0,995
TLOC	0,847	0,863	0,405	0,413	0,168	0,967	0,785	-0,243	0,558	0,905	0,471	-0,128	0,739	0,377	0,986	0,875	0,938	0,997	1	0,993	0,712	0,997
TRFC	0,861	0,853	0,373	0,435	0,360	0,968	0,818	-0,255	0,559	0,922	0,449	-0,133	0,758	0,351	0,972	0,839	0,942	0,995	0,993	1	0,776	0,995
TINVOK	0,750	0,519	0,156	-0,062	0,566	0,679	0,738	-0,293	0,381	0,736	0,237	-0,153	0,598	0,132	0,654	0,510	0,772	0,707	0,712	0,776	1	0,707
TWMPC1	0,843	0,872	0,392	0,473	0,209	0,975	0,796	-0,238	0,564	0,914	0,465	-0,124	0,753	0,372	0,983	0,857	0,930	1,000	0,997	0,995	0,707	1

Tableau 17.3 : Matrice de corrélation (Pearson) de Telestrada AspectJ (Toutes les classes)

Variables	HDiff	HEIT	FO	LCOM1	LCOM3	LOC	NOA	NOCC	RFC	WMPC1	CBO	DOIH	NOO	MTMOOP	TNofCalled	TNofNewData	TNofAssert	TNOO	TLOC	TRFC	TINVOK	TWMPG1
HDiff	1	0,997	0,588	-0,162	-0,171	0,855	0,707	-0,369	0,424	0,805	0,674	-0,016	0,748	0,285	0,963	0,878	0,958	0,965	0,965	0,964	0,940	0,965
HEIT	0,997	1	0,591	-0,032	-0,052	0,861	0,717	-0,367	0,438	0,811	0,674	-0,013	0,756	0,294	0,970	0,889	0,962	0,971	0,971	0,969	0,939	0,971
FO	0,588	0,591	1	0,127	-0,268	0,693	0,234	0,054	0,033	0,677	0,981	-0,325	0,664	0,554	0,530	0,493	0,499	0,520	0,520	0,517	0,465	0,520
LCOM1	-0,162	-0,032	0,127	1	0,750	0,230	0,306	0,385	0,395	0,286	0,093	0,185	0,525	0,070	0,029	-0,174	-0,033	0,085	0,058	0,168	0,067	0,085
LCOM3	-0,171	-0,052	-0,268	0,750	1	0,320	0,907	0,314	0,365	0,322	-0,355	0,424	0,899	-0,391	-0,074	-0,447	0,060	0,173	0,144	0,324	0,480	0,173
LOC	0,855	0,861	0,693	0,230	0,320	1	0,661	-0,121	0,306	0,969	0,754	-0,210	0,947	0,254	0,841	0,789	0,835	0,842	0,842	0,841	0,809	0,842
NOA	0,707	0,717	0,234	0,306	0,907	0,661	1	-0,268	0,415	0,653	0,315	0,017	0,602	0,025	0,752	0,783	0,759	0,758	0,754	0,758	0,723	0,758
NOCC	-0,369	-0,367	0,054	0,385	0,314	-0,121	-0,268	1	-0,104	-0,026	0,011	-0,194	0,027	-0,028	-0,415	-0,370	-0,413	-0,415	-0,416	-0,416	-0,411	-0,415
RFC	0,424	0,438	0,033	0,395	0,365	0,306	0,415	-0,104	1	0,249	0,090	0,770	0,207	0,521	0,468	0,488	0,461	0,466	0,466	0,465	0,418	0,466
WMPC1	0,805	0,811	0,677	0,286	0,322	0,969	0,653	-0,026	0,249	1	0,737	-0,303	0,986	0,160	0,796	0,758	0,793	0,799	0,798	0,798	0,764	0,799
CBO	0,674	0,674	0,981	0,093	-0,355	0,754	0,315	0,011	0,090	0,737	1	-0,310	0,714	0,543	0,610	0,569	0,585	0,603	0,603	0,598	0,546	0,603
DOIH	-0,016	-0,013	-0,325	0,185	0,424	-0,210	0,017	-0,194	0,770	-0,303	-0,310	1	-0,334	0,480	0,028	0,026	0,029	0,027	0,028	0,028	0,028	0,027
NOO	0,748	0,756	0,664	0,525	0,899	0,947	0,602	0,027	0,207	0,986	0,714	-0,334	1	0,122	0,737	0,686	0,734	0,742	0,740	0,741	0,711	0,742
MTMOOP	0,285	0,294	0,554	0,070	-0,391	0,254	0,025	-0,028	0,521	0,160	0,543	0,480	0,122	1	0,247	0,298	0,223	0,234	0,235	0,230	0,166	0,234
TNofCalled	0,963	0,970	0,530	0,029	-0,074	0,841	0,752	-0,415	0,468	0,796	0,610	0,028	0,737	0,247	1	0,922	0,989	0,999	0,999	0,998	0,966	0,999
TNofNewData	0,878	0,889	0,493	-0,174	-0,447	0,789	0,783	-0,370	0,488	0,758	0,569	0,026	0,686	0,298	0,922	1	0,901	0,919	0,920	0,917	0,827	0,919
TNofAssert	0,958	0,962	0,499	-0,033	0,060	0,835	0,759	-0,413	0,461	0,793	0,585	0,029	0,734	0,223	0,989	0,901	1	0,989	0,990	0,991	0,980	0,989
TNOO	0,965	0,971	0,520	0,085	0,173	0,842	0,758	-0,415	0,466	0,799	0,603	0,027	0,742	0,234	0,999	0,919	0,989	1	1,000	0,999	0,967	1,000
TLOC	0,965	0,971	0,520	0,058	0,144	0,842	0,754	-0,416	0,466	0,798	0,603	0,028	0,740	0,235	0,999	0,920	0,990	1,000	1	0,999	0,968	1,000
TRFC	0,964	0,969	0,517	0,168	0,324	0,841	0,758	-0,416	0,465	0,798	0,598	0,028	0,741	0,230	0,998	0,917	0,991	0,999	0,999	1	0,972	0,999
TINVOK	0,940	0,939	0,465	0,067	0,480	0,809	0,723	-0,411	0,418	0,764	0,546	0,028	0,711	0,166	0,966	0,827	0,980	0,967	0,968	0,972	1	0,967
TWMPG1	0,965	0,971	0,520	0,085	0,173	0,842	0,758	-0,415	0,466	0,799	0,603	0,027	0,742	0,234	0,999	0,919	0,989	1,000	1,000	0,999	0,967	1

Tableau 17.4 : Matrice de corrélation (Spearman) de Telestrada AspectJ (Toutes les classes)

Variables	HDift	HEIT	FO	LCOM1	LCOM3	LOC	NOA	NOCC	RFC	WMPC1	CBO	DOIH	NOO	MTMOOP	TNofCalled	TNofNewData	TNofAssert	TNOO	TLOC	TRFC	TINVOK	TWMPC1
HDift	1	0,747	0,773	0,260	0,224	0,822	0,464	-0,189	0,799	0,738	0,759	-0,035	0,613	0,753	0,646	0,642	0,650	0,668	0,666	0,674	0,514	0,668
HEIT	0,747	1	0,773	0,372	0,246	0,898	0,243	-0,089	0,691	0,733	0,725	-0,065	0,556	0,717	0,742	0,653	0,825	0,827	0,835	0,819	0,335	0,827
FO	0,773	0,773	1	0,489	0,442	0,878	0,417	-0,099	0,785	0,785	0,987	-0,149	0,717	0,970	0,772	0,774	0,753	0,790	0,780	0,795	0,589	0,790
LCOM1	0,260	0,372	0,489	1	0,187	0,589	0,011	-0,037	0,582	0,754	0,504	-0,097	0,870	0,458	0,585	0,626	0,620	0,672	0,667	0,682	0,604	0,672
LCOM3	0,224	0,246	0,442	0,187	1	0,372	0,490	-0,272	0,248	0,298	0,449	-0,319	0,355	0,416	0,276	0,288	0,274	0,277	0,269	0,280	0,239	0,277
LOC	0,822	0,898	0,878	0,589	0,372	1	0,417	-0,124	0,841	0,879	0,846	-0,095	0,771	0,824	0,757	0,739	0,832	0,847	0,840	0,848	0,537	0,847
NOA	0,464	0,243	0,417	0,011	0,490	0,417	1	-0,158	0,490	0,294	0,447	-0,002	0,434	0,441	0,315	0,381	0,176	0,213	0,217	0,227	0,424	0,213
NOCC	-0,189	-0,089	-0,099	-0,037	-0,272	-0,124	-0,158	1	-0,156	-0,110	-0,123	-0,257	-0,059	-0,166	-0,119	-0,136	-0,130	-0,134	-0,122	-0,142	-0,261	-0,134
RFC	0,799	0,691	0,785	0,582	0,248	0,841	0,490	-0,156	1	0,848	0,785	0,180	0,796	0,800	0,679	0,724	0,632	0,683	0,699	0,692	0,594	0,683
WMPC1	0,738	0,733	0,785	0,754	0,298	0,879	0,294	-0,110	0,848	1	0,767	-0,053	0,882	0,737	0,735	0,751	0,817	0,846	0,833	0,849	0,577	0,846
CBO	0,759	0,725	0,987	0,504	0,449	0,846	0,447	-0,123	0,785	0,767	1	-0,130	0,730	0,986	0,783	0,800	0,729	0,778	0,771	0,787	0,640	0,778
DOIH	-0,035	-0,065	-0,149	-0,097	-0,319	-0,095	-0,002	-0,257	0,180	-0,053	-0,130	1	-0,136	0,028	-0,066	-0,018	-0,096	-0,086	-0,064	-0,081	0,071	-0,086
NOO	0,613	0,556	0,717	0,870	0,155	0,771	0,434	-0,059	0,796	0,882	0,730	-0,136	1	0,675	0,728	0,766	0,684	0,744	0,738	0,755	0,686	0,744
MTMOOP	0,753	0,717	0,970	0,458	0,416	0,824	0,441	-0,166	0,800	0,737	0,986	0,028	0,675	1	0,765	0,787	0,706	0,755	0,752	0,764	0,637	0,755
TNofCalled	0,646	0,742	0,772	0,585	0,276	0,757	0,315	-0,119	0,679	0,735	0,783	-0,066	0,728	0,765	1	0,959	0,859	0,895	0,923	0,899	0,610	0,895
TNofNewData	0,642	0,653	0,774	0,626	0,288	0,739	0,381	-0,136	0,724	0,751	0,800	-0,018	0,766	0,787	0,959	1	0,784	0,843	0,873	0,853	0,718	0,843
TNofAssert	0,650	0,825	0,753	0,620	0,274	0,832	0,176	-0,130	0,632	0,817	0,729	-0,096	0,684	0,706	0,859	0,784	1	0,988	0,973	0,984	0,511	0,988
TNOO	0,668	0,827	0,790	0,672	0,277	0,847	0,213	-0,134	0,683	0,846	0,778	-0,086	0,744	0,755	0,899	0,853	0,984	0,999	0,991	1	0,613	0,999
TLOC	0,666	0,835	0,780	0,667	0,269	0,840	0,217	-0,122	0,699	0,833	0,771	-0,064	0,738	0,752	0,923	0,873	0,973	0,992	1	0,991	0,576	0,992
TRFC	0,674	0,819	0,795	0,682	0,280	0,848	0,227	-0,142	0,692	0,849	0,787	-0,081	0,755	0,764	0,899	0,853	0,984	0,999	0,991	1	0,613	0,999
TINVOK	0,514	0,335	0,589	0,604	0,239	0,537	0,424	-0,261	0,594	0,577	0,640	0,071	0,686	0,637	0,610	0,718	0,511	0,582	0,576	0,613	1	0,582
TWMPC1	0,668	0,827	0,790	0,672	0,277	0,847	0,213	-0,134	0,683	0,846	0,778	-0,086	0,744	0,755	0,895	0,843	0,988	1,000	0,992	0,999	0,582	1

Tableau 18.1 : Matrice de corrélation (Pearson) de Petstore Java (Toutes les classes)

Variables	HDiff	HEff	FO	LCOM1	LCOM3	LOC	NOA	NOCC	RFC	WMPC1	CBO	DOIH	NOO	MTMOOP	TNoFCalled	TNoFNewData	TNoFAssert	TNOO	TLOC	TRFC	TINVOK	TWMPC1	
HDiff	1	0,978	0,731	0,648	0,177	0,905	0,566	-0,249	0,813	0,838	0,747	0,047	0,719	0,699	0,802	0,718	0,770	0,797	0,798	0,796	0,666	0,797	
HEff	0,978	1	0,765	0,673	0,316	0,953	0,611	-0,244	0,812	0,866	0,782	0,014	0,770	0,708	0,832	0,759	0,792	0,830	0,830	0,831	0,712	0,830	
FO	0,731	0,765	1	0,554	0,499	0,763	0,407	-0,033	0,629	0,696	0,992	-0,179	0,695	0,861	0,651	0,610	0,572	0,616	0,625	0,627	0,558	0,616	
LCOM1	0,648	0,673	0,554	1	0,477	0,729	0,279	0,099	0,703	0,783	0,538	0,006	0,684	0,516	0,496	0,432	0,538	0,515	0,506	0,505	0,260	0,515	
LCOM3	0,177	0,316	0,499	0,477	1	0,484	0,500	-0,267	0,232	0,388	0,485	-0,243	0,510	0,382	0,371	0,362	0,382	0,354	0,373	0,378	0,259	0,354	
LOC	0,905	0,953	0,763	0,729	0,484	1	0,682	-0,134	0,790	0,925	0,773	-0,013	0,865	0,676	0,821	0,769	0,785	0,819	0,818	0,819	0,703	0,819	
NOA	0,566	0,611	0,407	0,279	0,500	0,682	1	-0,193	0,531	0,659	0,420	0,101	0,700	0,324	0,595	0,649	0,580	0,574	0,571	0,576	0,578	0,574	
NOCC	-0,249	-0,244	-0,033	0,099	-0,267	-0,134	-0,193	1	-0,167	-0,062	-0,060	-0,193	0,046	-0,153	-0,399	-0,376	-0,395	-0,398	-0,397	-0,398	-0,367	-0,398	
RFC	0,813	0,812	0,629	0,703	0,232	0,790	0,531	-0,167	1	0,768	0,635	0,454	0,635	0,796	0,744	0,710	0,729	0,734	0,732	0,733	0,658	0,734	
WMPC1	0,838	0,866	0,696	0,783	0,388	0,925	0,659	-0,062	0,768	1	0,702	-0,010	0,913	0,605	0,806	0,744	0,802	0,813	0,805	0,812	0,678	0,813	
CBO	0,747	0,782	0,992	0,538	0,485	0,773	0,420	-0,060	0,635	0,702	1	-0,176	0,701	0,868	0,671	0,634	0,578	0,641	0,649	0,652	0,593	0,641	
DOIH	0,047	0,014	-0,179	0,006	-0,243	-0,013	0,101	-0,193	0,454	-0,010	-0,176	1	-0,169	0,248	0,082	0,136	0,110	0,069	0,070	0,062	0,144	0,069	
NOO	0,719	0,770	0,695	0,684	0,510	0,865	0,700	0,046	0,635	0,913	0,701	-0,169	1	0,511	0,699	0,661	0,674	0,691	0,687	0,692	0,596	0,691	
MTMOOP	0,699	0,708	0,861	0,516	0,382	0,676	0,324	-0,153	0,796	0,605	0,868	0,248	0,511	1	0,646	0,632	0,575	0,615	0,623	0,623	0,596	0,615	
TNoFCalled	0,802	0,832	0,651	0,496	0,371	0,821	0,595	-0,399	0,744	0,806	0,671	0,082	0,699	0,646	1	0,942	0,967	0,989	0,989	0,876	0,989		
TNoFNewData	0,718	0,759	0,610	0,432	0,362	0,769	0,649	-0,376	0,710	0,744	0,634	0,136	0,661	0,632	0,942	1	0,884	0,913	0,915	0,917	0,877	0,913	
TNoFAssert	0,770	0,792	0,572	0,538	0,382	0,785	0,580	-0,395	0,729	0,802	0,578	0,110	0,674	0,575	0,967	0,884	1	0,966	0,960	0,964	0,824	0,966	
TNOO	0,797	0,830	0,616	0,515	0,354	0,819	0,574	-0,398	0,734	0,813	0,641	0,069	0,691	0,615	0,989	0,913	0,966	1,000	0,998	0,998	0,865	1	
TLOC	0,798	0,830	0,625	0,506	0,373	0,818	0,571	-0,397	0,732	0,805	0,649	0,070	0,687	0,623	0,989	0,915	0,960	0,998	1	0,996	0,862	0,998	
TRFC	0,796	0,831	0,627	0,505	0,378	0,819	0,576	-0,398	0,733	0,812	0,652	0,062	0,692	0,623	0,989	0,917	0,964	0,998	0,996	1	0,883	0,998	
TINVOK	0,666	0,712	0,558	0,260	0,259	0,703	0,578	-0,367	0,658	0,678	0,593	0,144	0,596	0,596	0,876	0,877	0,824	0,865	0,862	0,883	1	0,865	
TWMPC1	0,797	0,830	0,616	0,515	0,354	0,819	0,574	-0,398	0,734	0,813	0,641	0,069	0,691	0,615	0,989	0,913	0,966	1,000	0,998	0,998	0,865	1	

Tableau 18.2 : Matrice de corrélation (Spearman) de Petstore Java (Toutes les classes)

Variables	HDiff	HEff	FO	LCOM1	L.COM3	LOC	NOA	NOCC	RFC	WMPC1	CBO	DOIH	NOO	MTMOOP	TNofCalled	TNofNewData	TNofAsset	TNOO	TLOC	TRFC	TINVOK	TWMPC1
HDiff	1	0,800	0,383	0,588	-0,070	0,811	0,446	0,100	0,680	0,714	0,445	0,225	0,386	0,476	0,576	0,470	0,535	0,669	0,651	0,589	0,312	0,669
HEff	0,800	1	0,375	0,749	0,160	0,790	0,434	0,361	0,594	0,713	0,414	0,051	0,380	0,352	0,346	0,226	0,319	0,464	0,442	0,350	0,094	0,464
FO	0,383	0,375	1	0,362	0,291	0,398	0,209	0,109	0,219	0,346	0,974	-0,211	0,236	0,765	0,203	0,314	0,105	0,338	0,300	0,291	0,145	0,338
LCOM1	0,588	0,749	0,362	1	0,206	0,816	0,576	0,642	0,740	0,842	0,358	-0,021	0,714	0,280	0,076	0,052	0,053	0,204	0,174	0,077	-0,095	0,204
L.COM3	-0,070	0,160	0,291	0,206	1	0,315	0,592	-0,116	-0,097	0,122	0,307	-0,322	0,293	0,179	-0,102	-0,074	-0,057	0,034	0,011	-0,095	-0,185	0,034
LOC	0,811	0,790	0,398	0,816	0,315	1	0,643	0,172	0,753	0,831	0,442	0,135	0,606	0,384	0,508	0,428	0,465	0,598	0,590	0,519	0,264	0,598
NOA	0,446	0,434	0,209	0,576	0,592	0,643	1	0,080	0,449	0,474	0,242	0,174	0,309	0,269	0,605	0,585	0,568	0,646	0,645	0,573	0,311	0,646
NOCC	0,100	0,261	0,109	0,642	-0,116	0,172	0,080	1	0,220	0,158	0,089	-0,066	0,059	0,037	-0,141	-0,121	-0,141	-0,146	-0,147	-0,140	-0,092	-0,146
RFC	0,680	0,594	0,219	0,740	-0,097	0,753	0,449	0,320	1	0,763	0,265	0,539	0,530	0,451	0,499	0,412	0,464	0,468	0,487	0,543	0,493	0,468
WMPC1	0,714	0,713	0,346	0,842	0,122	0,831	0,474	0,158	0,763	1	0,369	0,055	0,865	0,224	0,389	0,302	0,363	0,441	0,433	0,380	0,188	0,441
CBO	0,445	0,414	0,974	0,358	0,307	0,442	0,242	0,089	0,265	0,369	1	-0,171	0,236	0,812	0,238	0,348	0,132	0,387	0,349	0,340	0,181	0,387
DOIH	0,225	0,053	-0,211	-0,021	-0,322	0,135	0,174	-0,066	0,539	0,055	-0,171	1	-0,122	0,409	0,458	0,303	0,510	0,288	0,335	0,416	0,476	0,288
NOO	0,386	0,380	0,236	0,714	0,293	0,606	0,309	0,059	0,530	0,865	0,236	-0,122	1	-0,024	0,190	0,184	0,148	0,232	0,223	0,179	0,056	0,232
MTMOOP	0,476	0,352	0,765	0,280	0,179	0,384	0,269	0,037	0,451	0,224	0,812	0,409	-0,024	1	0,442	0,463	0,378	0,483	0,474	0,519	0,423	0,483
TNofCalled	0,576	0,346	0,203	0,076	-0,102	0,508	0,605	-0,141	0,499	0,389	0,238	0,458	0,190	0,442	1	0,861	0,958	0,869	0,919	0,920	0,732	0,869
TNofNewData	0,470	0,226	0,314	0,052	-0,074	0,428	0,585	-0,121	0,412	0,302	0,348	0,303	0,184	0,463	0,861	1	0,701	0,824	0,837	0,829	0,603	0,824
TNofAsset	0,535	0,319	0,105	0,053	-0,057	0,465	0,568	-0,141	0,464	0,363	0,132	0,510	0,148	0,378	0,958	0,701	1	0,767	0,828	0,839	0,703	0,767
TNOO	0,669	0,464	0,338	0,204	0,034	0,598	0,646	-0,146	0,468	0,441	0,387	0,289	0,232	0,483	0,869	0,824	0,767	1	0,986	0,917	0,544	1,000
TLOC	0,651	0,442	0,300	0,174	0,011	0,590	0,645	-0,147	0,487	0,433	0,349	0,335	0,223	0,474	0,919	0,837	0,828	0,986	1	0,936	0,604	0,986
TRFC	0,589	0,350	0,291	0,077	-0,095	0,519	0,573	-0,140	0,543	0,380	0,340	0,416	0,179	0,519	0,920	0,829	0,839	0,917	0,936	1	0,834	0,917
TINVOK	0,312	0,094	0,145	-0,095	-0,185	0,264	0,311	-0,092	0,493	0,188	0,181	0,476	0,056	0,423	0,732	0,603	0,703	0,544	0,604	0,834	1	0,544
TWMPC1	0,669	0,464	0,338	0,204	0,034	0,598	0,646	-0,146	0,468	0,441	0,387	0,289	0,232	0,483	0,869	0,824	0,767	1,000	0,986	0,917	0,544	1

Tableau 18.3 : Matrice de corrélation (Pearson) de Petstore AspectJ (Toutes les classes)

Variables	HDiff	HEff	FO	LCOM1	LCOM3	LOC	NOA	NOCC	RFC	WMPC1	CBO	DOI11	NOO	MTMOOP	TNofCalled	TNofNewData	TNofAsset	TNOO	TLOC	TRFC	TINVOK	TWMPC1
HDiff	1	0,985	0,144	0,335	-0,208	0,823	0,645	-0,091	0,847	0,539	0,199	0,643	0,261	0,466	0,699	0,656	0,661	0,727	0,723	0,726	0,663	0,727
HEff	0,985	1	0,154	0,410	0,044	0,864	0,668	-0,105	0,826	0,547	0,210	0,607	0,290	0,429	0,698	0,654	0,649	0,724	0,722	0,725	0,651	0,724
FO	0,144	0,154	1	0,395	0,390	0,202	0,173	0,018	0,041	0,379	0,985	-0,285	0,407	0,606	0,098	0,096	0,026	0,129	0,122	0,135	0,057	0,129
LCOM1	0,335	0,410	0,395	1	0,274	0,505	0,311	0,170	0,373	0,565	0,383	-0,102	0,564	0,285	0,321	0,243	0,234	0,404	0,399	0,405	0,055	0,404
LCOM3	-0,208	0,044	0,390	0,274	1	0,394	0,724	-0,309	-0,110	0,062	0,356	-0,295	0,320	0,181	0,048	0,027	0,143	0,009	0,027	0,132	0,165	0,009
LOC	0,823	0,864	0,202	0,505	0,394	1	0,634	-0,074	0,788	0,716	0,244	0,481	0,562	0,346	0,602	0,579	0,550	0,622	0,624	0,623	0,540	0,622
NOA	0,645	0,668	0,173	0,311	0,724	0,634	1	-0,061	0,567	0,492	0,192	0,455	0,393	0,359	0,768	0,775	0,740	0,745	0,747	0,766	0,716	0,745
NOCC	-0,091	-0,105	0,018	0,170	-0,309	-0,074	-0,061	1	-0,092	-0,085	0,000	-0,095	0,060	-0,019	-0,160	-0,147	-0,163	-0,166	-0,168	-0,166	-0,155	-0,166
RFC	0,847	0,826	0,041	0,373	-0,110	0,788	0,567	-0,092	1	0,606	0,085	0,797	0,320	0,491	0,678	0,628	0,671	0,660	0,666	0,676	0,672	0,660
WMPC1	0,539	0,547	0,379	0,565	0,062	0,716	0,492	-0,085	0,606	1	0,405	0,106	0,841	0,215	0,375	0,327	0,359	0,380	0,383	0,394	0,328	0,380
CBO	0,199	0,210	0,985	0,383	0,356	0,244	0,192	0,000	0,085	0,405	1	-0,258	0,423	0,623	0,127	0,130	0,046	0,167	0,159	0,171	0,085	0,167
DOI11	0,643	0,607	-0,285	-0,102	-0,295	0,481	0,455	-0,095	0,797	0,106	-0,258	1	-0,179	0,471	0,674	0,639	0,699	0,627	0,637	0,645	0,706	0,627
NOO	0,261	0,290	0,407	0,564	0,320	0,562	0,393	0,060	0,320	0,841	0,423	-0,179	1	-0,012	0,171	0,170	0,130	0,190	0,187	0,188	0,091	0,190
MTMOOP	0,466	0,429	0,606	0,285	0,181	0,346	0,359	-0,019	0,491	0,215	0,623	0,471	-0,012	1	0,528	0,495	0,499	0,513	0,517	0,536	0,551	0,513
TNofCalled	0,699	0,698	0,098	0,321	0,048	0,602	0,768	-0,160	0,678	0,375	0,127	0,674	0,171	0,528	1	0,955	0,969	0,977	0,986	0,988	0,947	0,977
TNofNewData	0,656	0,654	0,096	0,243	0,027	0,579	0,775	-0,147	0,628	0,327	0,130	0,639	0,170	0,495	0,955	1	0,899	0,946	0,950	0,945	0,892	0,946
TNofAsset	0,661	0,649	0,026	0,234	0,143	0,550	0,740	-0,163	0,671	0,359	0,046	0,699	0,130	0,499	0,969	0,899	1	0,930	0,938	0,953	0,963	0,930
TNOO	0,727	0,724	0,129	0,404	0,009	0,622	0,745	-0,166	0,678	0,375	0,127	0,674	0,171	0,528	1	0,955	0,969	0,977	0,986	0,988	0,907	1,000
TLOC	0,723	0,722	0,122	0,399	0,027	0,624	0,747	-0,168	0,666	0,383	0,159	0,637	0,187	0,517	0,986	0,950	0,938	0,998	1	0,991	0,915	0,998
TRFC	0,726	0,725	0,135	0,405	0,132	0,623	0,766	-0,166	0,676	0,394	0,171	0,645	0,188	0,536	0,988	0,945	0,953	0,988	0,991	1	0,949	0,988
TINVOK	0,663	0,651	0,057	0,055	0,165	0,540	0,716	-0,155	0,672	0,328	0,085	0,706	0,091	0,551	0,947	0,892	0,963	0,907	0,915	0,949	1	0,907
TWMPC1	0,727	0,724	0,129	0,404	0,009	0,622	0,745	-0,166	0,660	0,380	0,167	0,627	0,190	0,513	0,977	0,946	0,930	1,000	0,998	0,998	0,907	1

Tableau 18.4 : Matrice de corrélation (Spearman) de Petstore AspectJ (Toutes les classes)

Variables	HDDIT	HEIT	FO	LCOM1	LCOM3	LOC	NOA	NOCC	RFC	WMPC1	CBO	DOIH	NOO	MTMOOP	TNofCalled	TNofNewData	TNofAssert	TNOO	TLOC	TRFC	TINVOK	TWMPC1
HDDIT	1	0,521	0,436	0,287	0,254	0,639	0,512	-0,105	0,478	0,662	0,459	0,027	0,575	0,410	0,358	0,284	0,474	0,678	0,634	0,644	0,358	0,678
HEIT	0,521	1	0,562	0,361	0,194	0,919	0,643	-0,017	0,500	0,928	0,510	-0,050	0,481	0,467	0,283	0,236	0,346	0,486	0,401	0,427	0,188	0,486
FO	0,436	0,562	1	0,442	0,304	0,678	0,620	-0,134	0,736	0,650	0,937	0,052	0,572	0,920	0,476	0,450	0,251	0,506	0,523	0,558	0,420	0,506
LCOM1	0,287	0,361	0,442	1	0,251	0,531	0,381	0,015	0,477	0,539	0,501	-0,028	0,819	0,415	0,510	0,442	0,363	0,530	0,565	0,452	0,166	0,530
LCOM3	0,254	0,194	0,304	0,251	1	0,336	0,358	-0,072	0,274	0,289	0,318	-0,090	0,341	0,281	0,195	0,158	0,273	0,311	0,290	0,272	0,109	0,311
LOC	0,639	0,919	0,678	0,531	0,336	1	0,691	-0,073	0,639	0,951	0,648	-0,042	0,684	0,585	0,447	0,372	0,516	0,701	0,599	0,639	0,316	0,701
NOA	0,512	0,643	0,620	0,381	0,358	0,691	1	-0,042	0,535	0,727	0,584	-0,081	0,546	0,531	0,439	0,408	0,237	0,429	0,557	0,420	0,252	0,429
NOCC	-0,105	-0,017	-0,134	0,015	-0,072	-0,073	-0,042	1	-0,110	-0,018	-0,161	-0,212	0,036	-0,205	-0,061	-0,057	-0,063	-0,058	0,029	-0,078	-0,075	-0,058
RFC	0,478	0,500	0,736	0,477	0,274	0,639	0,535	-0,110	1	0,634	0,677	0,358	0,599	0,684	0,408	0,364	0,315	0,513	0,494	0,507	0,311	0,513
WMPC1	0,662	0,928	0,650	0,539	0,289	0,951	0,727	-0,018	0,634	1	0,619	-0,017	0,708	0,553	0,378	0,312	0,436	0,631	0,572	0,587	0,308	0,631
CBO	0,459	0,510	0,937	0,501	0,318	0,648	0,584	-0,161	0,677	0,619	1	0,069	0,612	0,983	0,501	0,460	0,278	0,567	0,585	0,583	0,386	0,567
DOIH	0,027	-0,050	0,052	-0,028	-0,090	-0,042	-0,081	-0,212	0,358	-0,037	0,069	1	-0,002	0,216	-0,013	-0,008	-0,018	-0,009	0,001	0,046	0,104	-0,009
NOO	0,575	0,481	0,572	0,819	0,341	0,684	0,546	0,036	0,599	0,708	0,612	-0,002	1	0,511	0,576	0,479	0,576	0,746	0,727	0,683	0,343	0,746
MTMOOP	0,410	0,467	0,920	0,415	0,281	0,585	0,531	-0,205	0,684	0,553	0,983	0,216	0,511	1	0,450	0,420	0,213	0,497	0,520	0,530	0,376	0,497
TNofCalled	0,358	0,283	0,476	0,510	0,195	0,447	0,439	-0,061	0,408	0,378	0,501	-0,013	0,576	0,450	1	0,983	0,458	0,672	0,819	0,628	0,334	0,672
TNofNewData	0,284	0,236	0,450	0,442	0,158	0,372	0,408	-0,057	0,364	0,312	0,460	-0,008	0,479	0,420	0,983	1	0,319	0,571	0,747	0,543	0,303	0,571
TNofAssert	0,474	0,346	0,251	0,363	0,273	0,516	0,237	-0,063	0,315	0,436	0,278	-0,018	0,576	0,213	0,458	0,319	1	0,760	0,586	0,654	0,267	0,760
TNOO	0,678	0,486	0,506	0,530	0,311	0,701	0,429	-0,058	0,513	0,631	0,567	-0,009	0,746	0,497	0,672	0,571	0,760	1	0,835	0,886	0,401	1,000
TLOC	0,634	0,401	0,523	0,565	0,290	0,599	0,557	0,029	0,494	0,572	0,585	0,001	0,727	0,520	0,819	0,747	0,586	0,835	1	0,804	0,462	0,835
TRFC	0,644	0,427	0,558	0,452	0,272	0,639	0,420	-0,078	0,507	0,587	0,583	0,046	0,683	0,530	0,628	0,543	0,654	0,886	0,804	1	0,780	0,886
TINVOK	0,358	0,188	0,420	0,166	0,109	0,316	0,252	-0,075	0,311	0,308	0,386	0,104	0,343	0,376	0,334	0,303	0,267	0,401	0,462	0,780	1	0,401
TWMPC1	0,678	0,486	0,506	0,530	0,311	0,701	0,429	-0,058	0,513	0,631	0,567	-0,009	0,746	0,497	0,672	0,571	0,760	1,000	0,835	0,886	0,401	1

Tableau 19.1 : Matrice de corrélation (Pearson) de HSQLBD Java (Toutes les classes)

Variables	HDRIT	HITIT	FO	LCOM1	LCOM3	LOC	NOA	NOCC	RFC	WMPC1	CBO	DOIH	NOO	MTMOOP	TNofCalled	TNofNewData	TNofAsset	TNOO	TLOC	TRFC	TINVOK	TWMPC1
HDRIT	1	0,942	0,468	0,118	0,165	0,855	0,578	-0,314	0,666	0,839	0,517	0,191	0,595	0,476	0,759	0,643	0,668	0,816	0,811	0,757	0,506	0,816
HITIT	0,942	1	0,573	0,241	0,371	0,961	0,616	-0,319	0,732	0,883	0,627	0,165	0,653	0,574	0,806	0,657	0,709	0,870	0,858	0,820	0,571	0,870
FO	0,468	0,573	1	0,433	0,294	0,648	0,493	-0,134	0,689	0,608	0,960	0,163	0,546	0,925	0,677	0,631	0,401	0,648	0,673	0,690	0,609	0,648
LCOM1	0,118	0,241	0,433	1	0,446	0,385	0,151	0,243	0,502	0,459	0,430	0,050	0,632	0,401	0,293	0,195	0,374	0,358	0,344	0,392	0,428	0,358
LCOM3	0,165	0,371	0,294	0,446	1	0,498	0,414	0,008	0,317	0,350	0,306	-0,143	0,359	0,269	0,263	0,158	0,368	0,306	0,291	0,302	0,246	0,306
LOC	0,855	0,961	0,648	0,385	0,498	1	0,658	-0,256	0,775	0,907	0,688	0,176	0,736	0,622	0,816	0,660	0,726	0,884	0,865	0,851	0,641	0,884
NOA	0,578	0,616	0,493	0,151	0,414	0,658	1	-0,168	0,475	0,597	0,511	0,060	0,496	0,468	0,614	0,574	0,473	0,589	0,595	0,585	0,472	0,589
NOCC	-0,314	-0,319	-0,134	0,243	0,008	-0,256	-0,168	1	-0,219	-0,124	-0,187	-0,267	0,105	-0,248	-0,214	-0,117	-0,176	-0,174	-0,197	-0,167	-0,088	-0,174
RFC	0,666	0,732	0,689	0,502	0,317	0,775	0,475	-0,219	1	0,776	0,742	0,549	0,688	0,771	0,752	0,681	0,616	0,760	0,794	0,772	0,666	0,760
WMPC1	0,839	0,883	0,608	0,459	0,350	0,907	0,597	-0,124	0,776	1	0,650	0,144	0,876	0,565	0,851	0,729	0,766	0,918	0,895	0,892	0,715	0,918
CBO	0,517	0,627	0,960	0,430	0,306	0,688	0,511	-0,187	0,742	0,650	1	0,192	0,557	0,967	0,699	0,656	0,425	0,687	0,708	0,705	0,588	0,687
DOIH	0,191	0,165	0,163	0,050	-0,143	0,176	0,060	-0,267	0,549	0,144	0,192	1	0,054	0,367	0,232	0,266	0,194	0,148	0,240	0,212	0,263	0,148
NOO	0,595	0,653	0,546	0,632	0,359	0,736	0,496	0,105	0,688	0,876	0,557	0,054	1	0,437	0,723	0,634	0,689	0,768	0,742	0,774	0,730	0,768
MTMOOP	0,476	0,574	0,925	0,401	0,269	0,622	0,468	-0,248	0,771	0,565	0,967	0,367	0,437	1	0,645	0,623	0,374	0,612	0,654	0,636	0,527	0,612
TNofCalled	0,759	0,806	0,677	0,293	0,263	0,816	0,614	-0,214	0,752	0,851	0,699	0,232	0,723	0,645	1	0,878	0,796	0,902	0,935	0,931	0,790	0,902
TNofNewData	0,643	0,657	0,631	0,195	0,158	0,660	0,574	-0,117	0,681	0,729	0,656	0,266	0,634	0,623	0,878	1	0,566	0,731	0,789	0,791	0,754	0,731
TNofAsset	0,668	0,709	0,401	0,374	0,368	0,726	0,473	-0,176	0,616	0,766	0,425	0,194	0,689	0,374	0,796	0,566	1	0,752	0,754	0,750	0,641	0,752
TNOO	0,816	0,870	0,648	0,358	0,306	0,884	0,589	-0,174	0,760	0,918	0,687	0,148	0,768	0,612	0,902	0,731	0,752	1	0,968	0,952	0,685	1,000
TLOC	0,811	0,858	0,673	0,344	0,291	0,865	0,595	-0,197	0,794	0,895	0,708	0,240	0,742	0,654	0,935	0,789	0,754	0,968	1	0,951	0,721	0,968
TRFC	0,757	0,820	0,690	0,392	0,302	0,851	0,585	-0,167	0,772	0,892	0,705	0,212	0,774	0,636	0,931	0,791	0,750	0,952	0,951	1	0,854	0,952
TINVOK	0,506	0,571	0,609	0,428	0,246	0,641	0,472	-0,088	0,666	0,715	0,588	0,263	0,730	0,527	0,790	0,754	0,641	0,685	0,721	0,854	1	0,685
TWMPC1	0,816	0,870	0,648	0,358	0,306	0,884	0,589	-0,174	0,760	0,918	0,687	0,148	0,768	0,612	0,902	0,731	0,752	1,000	0,968	0,952	0,685	1

Tableau 19.2 : Matrice de corrélation (Spearman) de HSQLBD Java (Toutes les classes)

Variables	HDIR	HEIT	FO	LCOM1	LCOM2	LOC	NOA	NOCC	RFC	WMPCI	CBO	DOB	NOO	MTMOOP	NofCalled	NofNewData	NofAssert	TNOO	TLOC	TRPC	TINVOK	TWMPCI
HDIR	1	0,644	0,368	0,283	0,251	0,609	0,499	-0,081	0,437	0,688	0,406	0,034	0,561	0,350	0,288	0,224	0,507	0,619	0,688	0,622	0,302	0,619
HEIT	0,644	1	0,529	0,485	0,255	0,875	0,523	0,007	0,519	0,934	0,494	-0,070	0,584	0,422	0,238	0,175	0,500	0,549	0,514	0,557	0,456	0,549
FO	0,368	0,529	1	0,407	0,301	0,631	0,561	-0,129	0,667	0,609	0,918	0,052	0,497	0,898	0,338	0,312	0,253	0,396	0,421	0,405	0,430	0,396
LCOM1	0,283	0,485	0,407	1	0,266	0,560	0,335	0,010	0,489	0,604	0,477	-0,001	0,845	0,375	0,421	0,359	0,416	0,491	0,532	0,501	0,482	0,491
LCOM3	0,251	0,255	0,301	0,266	1	0,389	0,407	-0,066	0,249	0,313	0,326	-0,091	0,334	0,283	0,231	0,206	0,304	0,346	0,313	0,347	0,131	0,346
LOC	0,609	0,875	0,631	0,560	0,389	1	0,570	-0,019	0,566	0,895	0,606	-0,032	0,665	0,533	0,496	0,430	0,693	0,743	0,659	0,749	0,467	0,743
NOA	0,499	0,523	0,561	0,335	0,407	0,570	1	-0,021	0,487	0,612	0,544	-0,076	0,517	0,483	0,424	0,396	0,280	0,410	0,569	0,414	0,248	0,410
NOCC	-0,081	0,007	-0,129	0,010	-0,066	-0,069	-0,021	1	-0,084	0,001	-0,159	-0,209	0,057	-0,208	-0,054	-0,054	-0,046	-0,046	0,045	-0,047	-0,044	-0,046
RFC	0,437	0,519	0,667	0,489	0,249	0,566	0,487	-0,084	1	0,625	0,599	0,377	0,573	0,605	0,223	0,180	0,282	0,371	0,370	0,380	0,395	0,371
WMPCI	0,688	0,934	0,609	0,604	0,313	0,895	0,612	0,001	0,625	1	0,585	-0,030	0,755	0,498	0,321	0,256	0,548	0,619	0,600	0,628	0,507	0,619
CBO	0,406	0,494	0,918	0,477	0,326	0,606	0,544	-0,159	0,599	0,585	1	0,074	0,553	0,979	0,378	0,336	0,293	0,476	0,512	0,483	0,386	0,476
DOB	0,034	-0,070	0,052	-0,001	-0,091	-0,032	-0,076	-0,209	0,377	-0,030	0,074	1	0,018	0,238	-0,021	-0,022	0,004	-0,001	-0,005	0,004	0,169	-0,001
NOO	0,561	0,584	0,497	0,845	0,334	0,665	0,517	0,057	0,573	0,755	0,553	0,018	1	0,437	0,440	0,363	0,541	0,652	0,658	0,661	0,517	0,652
MTMOOP	0,350	0,422	0,898	0,375	0,283	0,533	0,483	-0,208	0,605	0,498	0,979	0,238	0,437	1	0,329	0,295	0,229	0,404	0,440	0,411	0,357	0,404
NofCalled	0,288	0,238	0,338	0,421	0,231	0,496	0,424	-0,054	0,223	0,321	0,378	-0,021	0,440	0,329	1	0,991	0,585	0,646	0,798	0,645	0,173	0,646
NofNewData	0,224	0,175	0,312	0,359	0,206	0,430	0,396	-0,054	0,180	0,256	0,336	-0,022	0,363	0,295	0,991	1	0,516	0,562	0,739	0,561	0,144	0,562
NofAssert	0,507	0,500	0,253	0,416	0,304	0,693	0,280	-0,046	0,282	0,548	0,293	0,004	0,541	0,229	0,585	0,516	1	0,928	0,690	0,928	0,350	0,928
TNOO	0,619	0,549	0,396	0,491	0,346	0,743	0,410	-0,046	0,371	0,619	0,476	-0,001	0,652	0,404	0,646	0,562	0,928	1	0,829	1,000	0,340	1,000
TLOC	0,608	0,514	0,421	0,532	0,313	0,659	0,569	0,045	0,370	0,600	0,512	-0,005	0,658	0,440	0,798	0,739	0,690	0,829	1	0,828	0,250	0,829
TRPC	0,622	0,557	0,405	0,501	0,347	0,749	0,414	-0,047	0,380	0,628	0,483	0,004	0,661	0,411	0,645	0,561	0,928	1,000	0,828	1	0,367	1,000
TINVOK	0,302	0,456	0,430	0,482	0,131	0,467	0,248	-0,044	0,395	0,507	0,386	0,169	0,517	0,357	0,173	0,144	0,350	0,340	0,250	0,367	1	0,340
TWMPCI	0,619	0,549	0,396	0,491	0,346	0,743	0,410	-0,046	0,371	0,619	0,476	-0,001	0,652	0,404	0,646	0,562	0,928	1,000	0,829	1,000	0,340	1

Tableau 19.3 : Matrice de corrélation (Pearson) de HSQLBD AspectJ (Toutes les classes)

Variables	HDFIT	HEIT	FO	LCOM4	LCOM3	LOC	NOCC	R/C	WMPCL	CBO	DOIH	NOO	MTMOOP	NoFCalled	NoNewData	NoFasserri	TNOO	TLOC	TRFC	TINVK	TWMPCL	
HDFIT	1	0.940	0.433	0.122	0.147	0.839	0.577	-0.275	0.656	0.485	0.183	0.563	0.436	0.731	0.610	0.696	0.793	0.782	0.790	0.453	0.793	
HEIT	0.940	1	0.532	0.268	0.352	0.946	0.616	-0.284	0.715	0.874	0.591	0.157	0.616	0.528	0.767	0.615	0.736	0.847	0.825	0.846	0.400	0.847
FO	0.433	0.532	1	0.427	0.281	0.628	0.467	-0.119	0.626	0.566	0.957	0.151	0.488	0.916	0.641	0.602	0.410	0.623	0.642	0.626	0.349	0.623
LCOM4	0.122	0.268	0.427	1	0.450	0.410	0.143	0.213	0.542	0.500	0.436	0.055	0.678	0.379	0.272	0.190	0.362	0.548	0.339	0.359	0.310	0.348
LCOM3	0.147	0.352	0.281	0.450	1	0.507	0.447	0.014	0.281	0.311	0.306	-0.149	0.329	0.254	0.236	0.149	0.329	0.316	0.293	0.317	0.114	0.316
LOC	0.839	0.946	0.628	0.410	0.507	1	0.657	-0.204	0.749	0.890	0.665	0.172	0.700	0.588	0.801	0.653	0.762	0.877	0.853	0.878	0.428	0.877
NOA	0.577	0.616	0.467	0.143	0.447	0.657	1	-0.111	0.447	0.588	0.485	0.060	0.475	0.434	0.608	0.559	0.519	0.583	0.591	0.584	0.323	0.583
NOCC	-0.276	-0.284	-0.119	0.213	0.014	-0.294	-0.111	1	-0.191	-0.103	-0.173	-0.253	0.126	-0.234	-0.171	-0.109	-0.140	-0.141	-0.147	-0.140	-0.069	-0.141
R/C	0.656	0.715	0.626	0.542	0.281	0.749	0.447	-0.191	1	0.764	0.689	0.562	0.664	0.715	0.685	0.614	0.599	0.725	0.746	0.729	0.422	0.725
WMPCL	0.836	0.974	0.566	0.500	0.311	0.890	0.588	-0.103	0.764	1	0.688	0.137	0.856	0.511	0.813	0.690	0.806	0.884	0.856	0.888	0.500	0.884
CBO	0.485	0.591	0.257	0.436	0.306	0.665	0.485	-0.173	0.689	0.608	1	0.186	0.495	0.962	0.663	0.616	0.448	0.666	0.680	0.668	0.456	0.666
DOIH	0.163	0.157	0.151	0.055	-0.149	0.172	0.060	-0.243	0.562	0.137	0.186	1	0.044	0.369	0.213	0.252	0.149	0.154	0.235	0.161	0.208	0.154
NOO	0.543	0.616	0.488	0.678	0.329	0.700	0.475	0.126	0.664	0.866	0.905	0.044	1	0.360	0.670	0.580	0.694	0.723	0.690	0.731	0.539	0.723
MTMOOP	0.436	0.528	0.916	0.379	0.254	0.588	0.434	-0.234	0.715	0.511	0.962	0.369	0.360	1	0.598	0.574	0.373	0.584	0.620	0.586	0.402	0.584
NoFCalled	0.731	0.767	0.641	0.272	0.236	0.801	0.608	-0.171	0.685	0.813	0.663	0.213	0.670	0.598	1	0.922	0.811	0.913	0.939	0.914	0.469	0.913
NoNewData	0.610	0.615	0.602	0.190	0.149	0.653	0.559	-0.169	0.614	0.690	0.616	0.252	0.860	0.574	0.922	1	0.663	0.750	0.812	0.753	0.402	0.740
NoFasserri	0.696	0.736	0.410	0.362	0.329	0.762	0.519	-0.140	0.599	0.800	0.448	0.149	0.694	0.373	0.811	0.663	1	0.819	0.804	0.822	0.529	0.819
TNOO	0.793	0.847	0.623	0.348	0.316	0.877	0.583	-0.141	0.725	0.884	0.666	0.154	0.723	0.584	0.913	0.750	0.819	1	0.968	0.998	0.459	1.000
TLOC	0.782	0.825	0.642	0.339	0.293	0.853	0.591	-0.147	0.46	0.836	0.680	0.235	0.690	0.620	0.939	0.812	0.894	1	0.968	0.943	0.968	
TRFC	0.790	0.846	0.626	0.359	0.317	0.878	0.584	-0.140	0.729	0.888	0.668	0.161	0.731	0.586	0.914	0.753	0.822	0.998	0.968	1	0.494	0.998
TINVK	0.353	0.400	0.249	0.310	0.114	0.428	0.323	-0.069	0.422	0.500	0.256	0.205	0.239	0.309	0.469	0.402	0.529	0.459	0.439	0.494	1	0.459
TWMPCL	0.793	0.847	0.623	0.348	0.316	0.877	0.583	-0.141	0.725	0.884	0.666	0.154	0.723	0.584	0.913	0.750	0.819	1.000	0.968	0.998	0.459	1

Tableau 19.4 : Matrice de corrélation (Spearman) de HSQLBD AspectJ (Toutes les classes)

Variables	HDiff	HEff	FO	LCOM1	LCOM3	LOC	NOA	NOCC	RFC	WMPC1	CBO	DOIH	NOO	MTMOOP	TNofCalled	TNofNewData	TNofAssert	TNOO	TLOC	TRFC	TINVOK	TWMPCL
HDiff	1	0,763	0,507	0,248	0,163	0,671	0,511	-0,131	0,199	0,599	0,468	0,167	0,434	0,448	0,547	0,506	0,570	0,703	0,654	0,553	0,377	0,706
HEff	0,763	1	0,549	0,522	0,211	0,803	0,528	-0,034	0,232	0,756	0,536	0,123	0,622	0,484	0,678	0,571	0,740	0,774	0,799	0,596	0,386	0,777
FO	0,507	0,549	1	0,770	0,383	0,833	0,699	0,062	0,352	0,800	0,895	0,196	0,813	0,830	0,683	0,658	0,525	0,656	0,666	0,708	0,638	0,661
LCOM1	0,248	0,522	0,770	1	0,284	0,812	0,687	0,217	0,387	0,829	0,761	0,199	0,873	0,704	0,799	0,724	0,672	0,788	0,811	0,763	0,610	0,790
LCOM3	0,163	0,211	0,383	0,284	1	0,366	0,543	0,174	0,152	0,357	0,358	0,025	0,378	0,321	0,315	0,333	0,197	0,246	0,274	0,341	0,402	0,244
LOC	0,671	0,803	0,833	0,812	0,366	1	0,782	-0,002	0,369	0,944	0,810	0,192	0,860	0,742	0,853	0,818	0,741	0,922	0,914	0,862	0,679	0,923
NOA	0,511	0,528	0,699	0,687	0,543	0,782	1	-0,009	0,301	0,742	0,659	0,123	0,685	0,598	0,648	0,657	0,427	0,639	0,640	0,652	0,562	0,641
NOCC	-0,131	-0,034	0,062	0,217	0,174	-0,002	-0,009	1	-0,106	0,074	0,027	-0,253	0,151	-0,050	0,005	0,013	-0,013	0,055	0,015	0,086	0,104	0,056
RFC	0,199	0,232	0,352	0,387	0,152	0,369	0,301	-0,106	1	0,321	0,436	0,848	0,321	0,591	0,623	0,581	0,509	0,559	0,597	0,607	0,538	0,557
WMPC1	0,599	0,756	0,800	0,829	0,357	0,944	0,742	0,074	0,321	1	0,735	0,097	0,948	0,638	0,843	0,818	0,730	0,952	0,924	0,863	0,649	0,951
CBO	0,468	0,536	0,895	0,761	0,358	0,810	0,659	0,027	0,436	0,735	1	0,312	0,735	0,968	0,694	0,663	0,556	0,657	0,670	0,719	0,648	0,661
DOIH	0,167	0,123	0,196	0,199	0,025	0,192	0,123	-0,253	0,848	0,097	0,312	1	0,065	0,533	0,149	0,114	0,145	0,072	0,127	0,106	0,130	0,075
NOO	0,434	0,622	0,813	0,873	0,378	0,860	0,685	0,151	0,321	0,948	0,735	0,065	1	0,625	0,809	0,797	0,671	0,881	0,856	0,854	0,660	0,882
MTMOOP	0,448	0,484	0,830	0,704	0,321	0,742	0,598	-0,050	0,591	0,638	0,968	0,533	0,625	1	0,627	0,591	0,504	0,564	0,594	0,639	0,594	0,571
TNofCalled	0,547	0,678	0,683	0,799	0,315	0,853	0,648	0,005	0,623	0,843	0,694	0,149	0,809	0,627	1	0,947	0,815	0,873	0,951	0,878	0,732	0,873
TNofNewData	0,506	0,571	0,658	0,724	0,333	0,818	0,657	0,013	0,581	0,818	0,663	0,114	0,787	0,591	0,947	1	0,620	0,820	0,876	0,874	0,763	0,821
TNofAssert	0,570	0,740	0,525	0,672	0,197	0,741	0,427	-0,013	0,509	0,730	0,556	0,145	0,671	0,504	0,815	0,620	1	0,809	0,853	0,732	0,546	0,808
TNOO	0,703	0,774	0,656	0,788	0,246	0,922	0,639	0,055	0,559	0,952	0,657	0,072	0,881	0,464	0,873	0,820	0,809	1	0,957	0,877	0,637	1,000
TLOC	0,654	0,799	0,666	0,811	0,274	0,914	0,640	0,015	0,597	0,924	0,670	0,127	0,856	0,594	0,951	0,876	0,853	0,957	1	0,857	0,671	0,957
TRFC	0,553	0,596	0,708	0,763	0,341	0,862	0,652	0,086	0,607	0,863	0,719	0,106	0,854	0,639	0,878	0,874	0,732	0,877	0,857	1	0,945	0,877
TINVOK	0,377	0,386	0,638	0,610	0,402	0,679	0,562	0,104	0,538	0,649	0,648	0,130	0,660	0,594	0,732	0,763	0,546	0,637	0,671	0,945	1	0,672
TWMPCL	0,706	0,777	0,661	0,790	0,244	0,923	0,641	0,056	0,557	0,951	0,661	0,075	0,882	0,571	0,873	0,821	0,808	1,000	0,957	0,877	0,672	1

Tableau 20.1 : Matrice de corrélation (Pearson) de JHotdraw Java (Toutes les classes)

Variables	Hdiff	Hfit	FO	LCOM1	LCOM3	LOC	NOA	NOCC	RFC	WMPC1	CBO	DOB1	NOO	MTMOOP	TNo(Called)	TNo(NewData)	TNo(Asset)	TNOO	TLOC	TRFC	TINVOK	TWMP1
Hdiff	1	0,972	0,638	0,316	0,205	0,859	0,639	-0,248	0,485	0,690	0,673	0,311	0,481	0,618	0,692	0,676	0,528	0,726	0,743	0,655	0,500	0,737
Hfit	0,972	1	0,709	0,467	0,282	0,919	0,648	-0,209	0,549	0,736	0,759	0,334	0,557	0,696	0,756	0,732	0,598	0,779	0,799	0,727	0,578	0,787
FO	0,638	0,709	1	0,613	0,366	0,776	0,489	0,021	0,492	0,698	0,904	0,222	0,639	0,771	0,659	0,667	0,508	0,662	0,693	0,698	0,642	0,680
LCOM1	0,316	0,467	0,613	1	0,426	0,664	0,349	0,433	0,410	0,754	0,568	0,099	0,817	0,479	0,644	0,592	0,614	0,621	0,655	0,710	0,690	0,643
LCOM3	0,205	0,282	0,366	0,426	1	0,384	0,763	0,190	0,157	0,382	0,368	-0,036	0,425	0,292	0,257	0,274	0,197	0,244	0,263	0,254	0,308	0,250
LOC	0,859	0,919	0,776	0,664	0,384	1	0,686	-0,127	0,586	0,881	0,778	0,254	0,761	0,669	0,839	0,810	0,691	0,861	0,869	0,833	0,684	0,862
NOA	0,639	0,648	0,489	0,349	0,763	0,686	1	-0,119	0,305	0,641	0,453	0,072	0,576	0,366	0,628	0,629	0,443	0,633	0,639	0,577	0,473	0,641
NOCC	-0,248	-0,209	0,021	0,433	0,190	-0,127	-0,119	1	-0,214	0,063	-0,026	-0,386	0,197	-0,152	-0,043	-0,014	-0,039	-0,015	-0,028	-0,012	0,030	-0,011
RFC	0,485	0,549	0,492	0,410	0,157	0,586	0,305	-0,214	1	0,431	0,583	0,770	0,350	0,771	0,589	0,550	0,566	0,495	0,547	0,622	0,640	0,493
WMPC1	0,690	0,736	0,698	0,754	0,382	0,881	0,641	0,063	0,431	1	0,667	0,049	0,924	0,494	0,883	0,858	0,706	0,937	0,931	0,874	0,676	0,937
CBO	0,673	0,759	0,904	0,568	0,368	0,778	0,453	-0,026	0,583	0,667	1	0,322	0,592	0,903	0,672	0,698	0,503	0,676	0,707	0,713	0,649	0,690
DOB1	0,311	0,334	0,222	0,099	-0,036	0,254	0,072	-0,386	0,770	0,049	0,322	1	-0,066	0,635	0,168	0,146	0,241	0,071	0,138	0,177	0,257	0,079
NOO	0,481	0,557	0,639	0,817	0,425	0,761	0,576	0,197	0,350	0,924	0,592	-0,066	1	0,389	0,786	0,760	0,651	0,835	0,826	0,813	0,662	0,833
MTMOOP	0,618	0,696	0,771	0,479	0,292	0,669	0,366	-0,152	0,771	0,494	0,903	0,635	0,389	1	0,578	0,596	0,457	0,546	0,597	0,615	0,594	0,565
TNo(Called)	0,692	0,756	0,659	0,644	0,257	0,839	0,628	-0,043	0,589	0,883	0,672	0,168	0,786	0,578	1	0,929	0,843	0,928	0,965	0,959	0,808	0,928
TNo(NewData)	0,676	0,732	0,667	0,592	0,274	0,810	0,629	-0,014	0,550	0,858	0,698	0,146	0,760	0,596	0,929	1	0,657	0,898	0,914	0,893	0,762	0,899
TNo(Asset)	0,528	0,598	0,508	0,614	0,197	0,691	0,443	-0,039	0,566	0,706	0,503	0,241	0,651	0,457	0,843	0,657	1	0,730	0,781	0,801	0,690	0,726
TNOO	0,726	0,779	0,662	0,621	0,244	0,861	0,633	-0,015	0,495	0,937	0,676	0,071	0,835	0,546	0,928	0,898	0,730	1	0,986	0,909	0,678	1,000
TLOC	0,743	0,799	0,693	0,655	0,263	0,869	0,639	-0,028	0,547	0,931	0,707	0,138	0,826	0,597	0,965	0,914	0,781	0,986	1	0,941	0,780	0,986
TRFC	0,655	0,727	0,698	0,710	0,254	0,833	0,577	-0,012	0,622	0,874	0,713	0,177	0,813	0,615	0,959	0,893	0,801	0,909	0,941	1	0,930	0,909
TINVOK	0,500	0,578	0,642	0,690	0,308	0,684	0,473	0,030	0,640	0,676	0,649	0,257	0,662	0,594	0,808	0,762	0,690	0,678	0,780	0,930	1	0,726
TWMP1	0,737	0,787	0,680	0,643	0,250	0,862	0,641	-0,011	0,493	0,937	0,690	0,079	0,833	0,565	0,928	0,899	0,726	1,000	0,986	0,909	0,726	1

Tableau 20.2 : Matrice de corrélation (Spearman) de JHotdraw Java (Toutes les classes)

Variables	HDiff	Hdiff	HEIT	FO	LCOM1	LCOM3	LOC	NOA	NOCC	RFC	WMPC1	CBO	DOIH	NOO	MTMOOP	TNoCalled	TNoNewData	TNoAssert	TNOO	TLOC	TRFC	TRVOK	TWMPC1
HDiff	1	0,767	0,504	0,250	0,158	0,676	0,514	-0,128	0,203	0,608	0,468	0,171	0,442	0,448	0,459	0,447	0,628	0,683	0,656	0,508	0,277	0,683	
HEIT	0,767	1	0,542	0,510	0,209	0,801	0,519	-0,033	0,224	0,755	0,527	0,120	0,621	0,475	0,448	0,403	0,777	0,721	0,745	0,513	0,252	0,721	
FO	0,504	0,542	1	0,775	0,381	0,830	0,700	0,089	0,353	0,798	0,901	0,200	0,811	0,836	0,460	0,462	0,323	0,519	0,468	0,594	0,563	0,519	
LCOM1	0,250	0,510	0,775	1	0,285	0,812	0,679	0,212	0,381	0,827	0,764	0,198	0,873	0,708	0,552	0,501	0,515	0,647	0,605	0,676	0,568	0,647	
LCOM3	0,158	0,209	0,381	0,285	1	0,369	0,546	0,169	0,151	0,352	0,357	0,031	0,373	0,321	0,231	0,222	0,032	0,146	0,189	0,230	0,257	0,146	
LOC	0,676	0,801	0,830	0,812	0,369	1	0,781	0,018	0,370	0,948	0,813	0,196	0,864	0,745	0,768	0,740	0,767	0,887	0,898	0,804	0,604	0,887	
NOA	0,514	0,519	0,700	0,679	0,546	0,781	1	0,006	0,302	0,738	0,662	0,129	0,681	0,601	0,612	0,594	0,346	0,557	0,566	0,544	0,445	0,557	
NOCC	-0,128	-0,033	0,089	0,212	0,169	0,018	0,006	1	-0,106	0,111	0,053	-0,259	0,204	-0,031	0,060	0,051	0,081	0,128	0,075	0,199	0,228	0,128	
RFC	0,203	0,224	0,353	0,381	0,151	0,370	0,302	-0,106	1	0,317	0,437	0,849	0,316	0,593	0,428	0,424	0,267	0,373	0,403	0,425	0,401	0,373	
WMPC1	0,608	0,755	0,798	0,827	0,352	0,948	0,738	0,111	0,317	1	0,736	0,099	0,948	0,639	0,770	0,740	0,817	0,944	0,922	0,885	0,692	0,944	
CBO	0,468	0,527	0,901	0,764	0,357	0,813	0,662	0,053	0,437	0,736	1	0,315	0,736	0,967	0,450	0,454	0,307	0,497	0,455	0,555	0,516	0,497	
DOIH	0,171	0,120	0,200	0,198	0,031	0,196	0,129	-0,259	0,849	0,099	0,315	1	0,067	0,537	0,037	0,044	-0,034	-0,040	0,005	-0,029	-0,014	-0,040	
NOO	0,442	0,621	0,811	0,873	0,373	0,864	0,681	0,204	0,316	0,948	0,736	0,067	1	0,626	0,691	0,657	0,699	0,821	0,778	0,853	0,744	0,821	
MTMOOP	0,448	0,475	0,836	0,708	0,321	0,745	0,601	-0,031	0,593	0,639	0,967	0,537	0,626	1	0,349	0,359	0,190	0,354	0,333	0,409	0,391	0,354	
TNoCalled	0,459	0,448	0,460	0,552	0,231	0,768	0,612	0,060	0,428	0,770	0,450	0,037	0,691	0,349	1	0,988	0,598	0,798	0,872	0,840	0,741	0,798	
TNoNewData	0,447	0,403	0,462	0,501	0,222	0,740	0,594	0,051	0,424	0,740	0,454	0,044	0,657	0,359	0,988	1	0,530	0,761	0,835	0,827	0,749	0,761	
TNoAssert	0,628	0,777	0,323	0,515	0,032	0,767	0,346	0,081	0,267	0,817	0,307	-0,034	0,699	0,190	0,598	0,530	1	0,878	0,850	0,706	0,445	0,878	
TNOO	0,683	0,721	0,519	0,647	0,146	0,887	0,557	0,128	0,373	0,944	0,497	-0,040	0,821	0,354	0,798	0,761	0,878	1	0,968	0,898	0,667	1,000	
TLOC	0,656	0,745	0,468	0,605	0,189	0,898	0,566	0,075	0,403	0,922	0,455	0,005	0,778	0,333	0,872	0,835	0,850	0,968	1	0,877	0,658	0,968	
TRFC	0,508	0,513	0,594	0,676	0,230	0,804	0,544	0,199	0,425	0,885	0,555	-0,029	0,853	0,409	0,840	0,827	0,706	0,898	0,877	1	0,926	0,898	
TRVOK	0,277	0,252	0,563	0,568	0,257	0,604	0,445	0,228	0,401	0,692	0,516	-0,014	0,744	0,391	0,741	0,749	0,445	0,667	0,658	0,926	1	0,667	
TWMPC1	0,683	0,721	0,519	0,647	0,146	0,887	0,557	0,128	0,373	0,944	0,497	-0,040	0,821	0,354	0,798	0,761	0,878	1,000	0,968	0,898	0,667	1	

Tableau 20.3 : Matrice de corrélation (Pearson) de AJHotdraw AspectJ (Toutes les classes)

Variables	HDiff	HEff	FO	LCOM4	LCOM3	LOC	NOA	NOCC	RFC	WMPC1	CBO	DOB	NOO	MTMOOP	TNofCalled	TNofNewData	TNofAsset	TNOO	TLOC	TRFC	TINVOK	TWMPG1
HDiff	1	0,973	0,640	0,330	0,198	0,863	0,653	-0,223	0,515	0,702	0,676	0,328	0,493	0,626	0,677	0,621	0,513	0,669	0,694	0,610	0,446	0,669
HEff	0,973	1	0,706	0,477	0,276	0,919	0,661	-0,184	0,576	0,747	0,760	0,348	0,567	0,703	0,713	0,653	0,542	0,702	0,729	0,657	0,506	0,702
FO	0,640	0,706	1	0,621	0,356	0,769	0,494	0,044	0,500	0,705	0,911	0,230	0,641	0,780	0,644	0,643	0,430	0,628	0,648	0,656	0,616	0,628
LCOM1	0,330	0,477	0,621	1	0,414	0,670	0,349	0,456	0,414	0,752	0,584	0,111	0,815	0,498	0,527	0,422	0,456	0,504	0,532	0,571	0,573	0,504
LCOM3	0,198	0,276	0,356	0,414	1	0,386	0,758	0,215	0,157	0,363	0,366	-0,022	0,415	0,298	0,154	0,140	0,095	0,139	0,154	0,098	0,067	0,139
LOC	0,863	0,919	0,769	0,670	0,386	1	0,704	-0,091	0,606	0,896	0,778	0,265	0,773	0,674	0,865	0,787	0,689	0,862	0,879	0,837	0,707	0,862
NOA	0,653	0,661	0,494	0,349	0,758	0,704	1	-0,106	0,339	0,643	0,457	0,089	0,575	0,379	0,578	0,528	0,420	0,543	0,553	0,496	0,401	0,543
NOCC	-0,223	-0,184	0,044	0,456	0,215	-0,091	-0,106	1	-0,206	0,092	-0,096	-0,369	0,224	-0,134	-0,138	-0,120	-0,136	-0,121	-0,143	-0,086	-0,035	-0,121
RFC	0,515	0,576	0,500	0,414	0,157	0,606	0,339	-0,206	1	0,463	0,595	0,788	0,374	0,787	0,554	0,561	0,405	0,516	0,557	0,562	0,558	0,516
WMPC1	0,702	0,747	0,705	0,752	0,363	0,896	0,643	0,092	0,463	1	0,672	0,063	0,924	0,505	0,906	0,816	0,744	0,929	0,928	0,910	0,787	0,929
CBO	0,676	0,760	0,911	0,584	0,366	0,778	0,457	-0,006	0,595	0,672	1	0,331	0,597	0,903	0,661	0,665	0,423	0,640	0,665	0,674	0,635	0,640
DOB	0,328	0,348	0,230	0,111	-0,022	0,265	0,089	-0,369	0,788	0,063	0,331	1	-0,052	0,642	0,138	0,173	0,133	0,089	0,136	0,102	0,110	0,089
NOO	0,493	0,567	0,641	0,815	0,415	0,773	0,575	0,224	0,374	0,924	0,597	-0,052	1	0,400	0,777	0,689	0,652	0,800	0,793	0,818	0,778	0,800
MTMOOP	0,626	0,703	0,780	0,498	0,298	0,674	0,379	-0,134	0,787	0,505	0,903	0,642	0,400	1	0,541	0,563	0,344	0,503	0,541	0,537	0,511	0,503
TNofCalled	0,677	0,713	0,644	0,527	0,154	0,865	0,578	-0,138	0,554	0,906	0,661	0,138	0,777	0,541	1	0,951	0,752	0,968	0,979	0,962	0,846	0,968
TNofNewData	0,621	0,653	0,643	0,422	0,140	0,787	0,528	-0,120	0,561	0,816	0,665	0,173	0,699	0,563	0,951	1	0,608	0,872	0,893	0,902	0,839	0,872
TNofAsset	0,513	0,542	0,430	0,456	0,095	0,689	0,420	-0,136	0,405	0,744	0,423	0,133	0,652	0,344	0,752	0,608	1	0,799	0,787	0,702	0,544	0,799
TNOO	0,669	0,702	0,628	0,504	0,139	0,862	0,543	-0,121	0,516	0,929	0,640	0,089	0,800	0,503	0,968	0,872	0,799	1	0,993	0,962	0,806	1,000
TLOC	0,694	0,729	0,648	0,532	0,154	0,879	0,553	-0,143	0,557	0,928	0,665	0,136	0,793	0,541	0,979	0,893	0,787	0,993	1	0,962	0,815	0,993
TRFC	0,610	0,657	0,656	0,571	0,098	0,837	0,496	-0,086	0,562	0,910	0,674	0,102	0,818	0,537	0,962	0,902	0,702	0,962	0,962	1	0,924	0,962
TINVOK	0,446	0,506	0,616	0,573	0,067	0,707	0,401	-0,035	0,558	0,787	0,635	0,110	0,778	0,511	0,846	0,839	0,544	0,806	0,815	0,924	1	0,806
TWMPG1	0,669	0,702	0,628	0,504	0,139	0,862	0,543	-0,121	0,516	0,929	0,640	0,089	0,800	0,503	0,968	0,872	0,799	1,000	0,993	0,962	0,806	1

Tableau 20.4 : Matrice de corrélation (Spearman) de AJHotdraw AspectJ (Toutes les classes)

Références bibliographiques

[Alexander04] Alexander, R. T., Bieman, J. M., Andrews, A. A., *Towards the systematic testing of aspect-oriented programs*. Technical Report CS-4-105, Colorado State University, Fort Collins, Colorado, 2004.

[AOPmetrics] <http://aopmetrics.tigris.org>

[AspectJ] <http://www.eclipse.org/aspectj/>

[Basili96] V.R. Basili, L.C. Briand, and W.L. Melo, "A Validation of Object-Oriented Design Metrics as Quality Indicators," *IEEE Transactions on Software Engineering*. vol. 22, no. 10, pp. 751-761, Oct. 1996.

[Baudry03] B. Baudry, Y. Le Traon, and G. Sunyé. Testability analysis of a UML class diagram. In *Proceedings of the Ninth International Software Metrics Symposium (METRICS03)*, pp. 54–66. IEEE Computer Society, 2003.

[Baudry031] B. Baudry, B., Le Traon, Y., Sunyé, G., Jézéquel, J.M., Measuring and Improving Design Patterns Testability, *Proceedings of the 9th International Software Metrics Symposium (METRICS)*, IEEE Computer Society, 2003.

[Binder94] Binder, R.V., Design for Testability in Object-Oriented Systems, *Communications of the ACM*, Vol. 37, 1994.

[Bruntink04] M. Bruntink and A.V. Deursen, Predicting class testability using object-oriented metrics, *Proceedings of the IEEE International Workshop on Source Code Analysis and Manipulation*, pp. 136–145, 2004.

[Bruntink06] Magiel Bruntink & Arie van Deursen. An empirical study into class testability. *Journal of Systems and Software*, vol. 79,no.9, pp. 1219–1232, 2006.

[Delamare09] Romain Delamare : THÈSE/UNIVERSITÉ DE RENNES 1. PhD. in computer science from the Université de Rennes 1. postdoctoral researcher at the University of Alabama, in the Software Engineering Group, 2009.

[Ceccato04] M. Ceccato and P. Tonella Measuring the Effects of Software Aspectization. In WCRE Workshop on Aspect Reverse Engineering (WARE). Published at the workshop Web site : <http://homepages.cwi.nl/~tourwe/ware/submissions.html>, 2004.

[Ceccato05] Ceccato, M., Tonella, P., Ricca, F., Is AOP code easier or harder to test than OOP code? In *On-line Proceedings of the First Workshop on Testing Aspect-Oriented Programs (WTAOP 2005)*, Chicago, Illinois, 2005.

[Chidamber94] Chidamber, S.R., Kemerer, C.F., A Metrics suite for object Oriented Design, *IEEE Transactions on Software Engineering*, Vol. 20, No. 6, pp. 476-493, June 1994.

[Chowdhary09] Vishal Chowdhary, Practicing Testability in the Real World, Proceedings of the 2009 International Conference on Software Testing Verification and Validation, pp. 260-268, April 01-04, 2009.

[CodePro] CodePro, <http://www.instantiations.com/codepro/index.html>.

[Freedman91] R. Freedman. Testability of software components. *IEEE Transactions on Software Engineering*, 17(6) pp. 553–564, June 1991.

[Garcia06] Alessandro Garcia, Cláudio Sant'Anna, Eduardo Figueiredo, Uirá Kulesza, Carlos Lucena et Arndt von Staa. Modularizing Design Patterns with Aspects: A Quantitative Study. Software Engineering Laboratory - Computer Science Department Pontifical Catholic University of Rio de Janeiro – PUC-Rio, 2006.

[GoF] Baumer, D., Riehle, D., Siberski, W., and Wulf, M. Role Object Pattern. Proceedings of PLoP '97. Technical Report WUCS-97-34. Washington University Dept. of Computer Science, 1997.

[Halstead77] M. H. Halstead. Elements of Software Science (Operating and programming systems series). Elsevier Science Inc., New York, NY, USA, 1977.

[Hannemann02] Hannemann, J., Kiczales, G. Design Pattern Implementation in Java and AspectJ. *Proc. OOPSLA '02*, pp. 161-173, Nov 2002.

[Henderson96] Henderson-Sellers, B., Object-Oriented Metrics Measures of Complexity, Prentice-Hall, 1996.

[JODE] JODE, <http://jode.sourceforge.net>.

[Jungmayr02] S. Jungmayr. Identifying test-critical dependencies. In Proceedings of the International Conference on Software Maintenance, pp. 404–413. IEEE Computer Society, October 2002.

[JUnit] JUnit, <http://www.junit.org/index.htm>.

[Khan09] Khan R A & Mustafa K, Metric Based Testability Model for Object Oriented Design (MTMOOD), ACM SIGSOFT Software Engineering Volume 34 , Issue 2 March 2009.

[Khoshgoftaar00] T.M. Khoshgoftaar, Edward B. Allen et Z. Xu: Predicting testability of program modules using a neural network. In : Proceedings of the 3rd IEEE Symposium on Application-Specific Systems and Software Engineering Technology, pp. 57–62. 2000.

[Kiczales97] Kiczales, G. et al. Aspect-Oriented Programming. *Proc. Of ECOOP '97, LNCS 1241*, Finland, pp. 220-242, June 1997.

[Kout11] A. Kout, F. Touré, M. Badri. An empirical analysis of a testability model for object-oriented programs. ACM SIGSOFT Software Engineering. Volume 36 Issue 4, July 2011.

[McGregor96] J. McGregor and S. Srinivas. A measure of testing effort. In Proceedings of the Conference on Object-Oriented Technologies, pp. 129–142. USENIX Association, June 1996.

[Metrics] <http://metrics.sourceforge.net/>

[Munoz 08] Freddy Munoz, Benoit Baudry & Olivier Barais. Improving maintenance in AOP through an interaction specification framework. In ICSM'09 : Proceedings of the 24th International Conference on Software Maintenance, 2008.

[Together] <http://www.borland.com>.

[Voas92] J. Voas. PIE: A dynamic failure-based technique. IEEE Transactions on Software Engineering, 18(8): pp. 717–727, August 1992.

[Voas93] J. Voas and K. Miller. Semantic metrics for software testability. Journal of Systems and Software, 20: pp. 207–216, March 1993.

[XLSTAT] <http://www.xlstat.com>.

[Xuand06] D. Xuand W. Xu. State-based incremental testing of aspect oriented programs. In *Int. Conf. Aspect-Oriented Software Development*, pp. 180–189, 2006.

[Zhao07] Chuan Zhao and Roger T. Alexander: Testing Aspect-Oriented Programs as Object-Oriented Programs. Washington State University, School of Electrical Engineering and Computer Science, Pullman, Washington. ACM New York, NY, USA ©2007.