Test-driven developement et pair programming

Étienne Frank, Grégory Burri

***Résumé*—Cet article à pour but de présenter les approches *TDD* (*test-driven development*) et *pair-programming* dans le domaine du développement logiciel, de mettre en avant leurs atouts ainsi que de les inscrire dans des méthodologies de développement connues.**

***Index Terms*—TDD, test-driven development, pair-programming.**

I. Introduction

CECI est un test.

II. *TDD*

A. Motivations

B. Fonctionnement

C. Bénéfices

D. Correction de bugs

La correction du bug peut aussi suivre une forme s'approchant du *TDD*. Lorsqu'un bug doit être corrigé, l'on va d'abord ajouter un ou plusieurs tests qui vont échoués afin de mettre en évidence le bug. Le code incriminé va ensuite être corrigé afin que le test passe. Une phase de *refactoring* peut

Cette approche est à la fois importante pour la documentation du bug corrigé mais également pour éviter toute régression ultérieure.

E. Intégration avec une approche agile

Les tests unitaires -> s'ajoute un processus d'intégration continue

III. Cas pratique

A. But

Afin d'évaluer ces deux approches de manière concrète nous les avons mis en œuvre à l'aide d'un petit cas pratique. Cet exemple ne se veut en aucun cas exhaustive et ne montre qu'une introduction échelle réduite.

B. Outils utilisés

IV. Conclusion

Annexe A
Proof of the First Zonklar Equation

Appendix one text goes here.

Annexe B

Appendix two text goes here.

Acknowledgment

The authors would like to thank...

Références

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England : Addison-Wesley, 1999.

PLACE PHOTO HERE

Michael Shell Biography text here.

John Doe Biography text here.

Jane Doe Biography text here.