Test-Driven Development

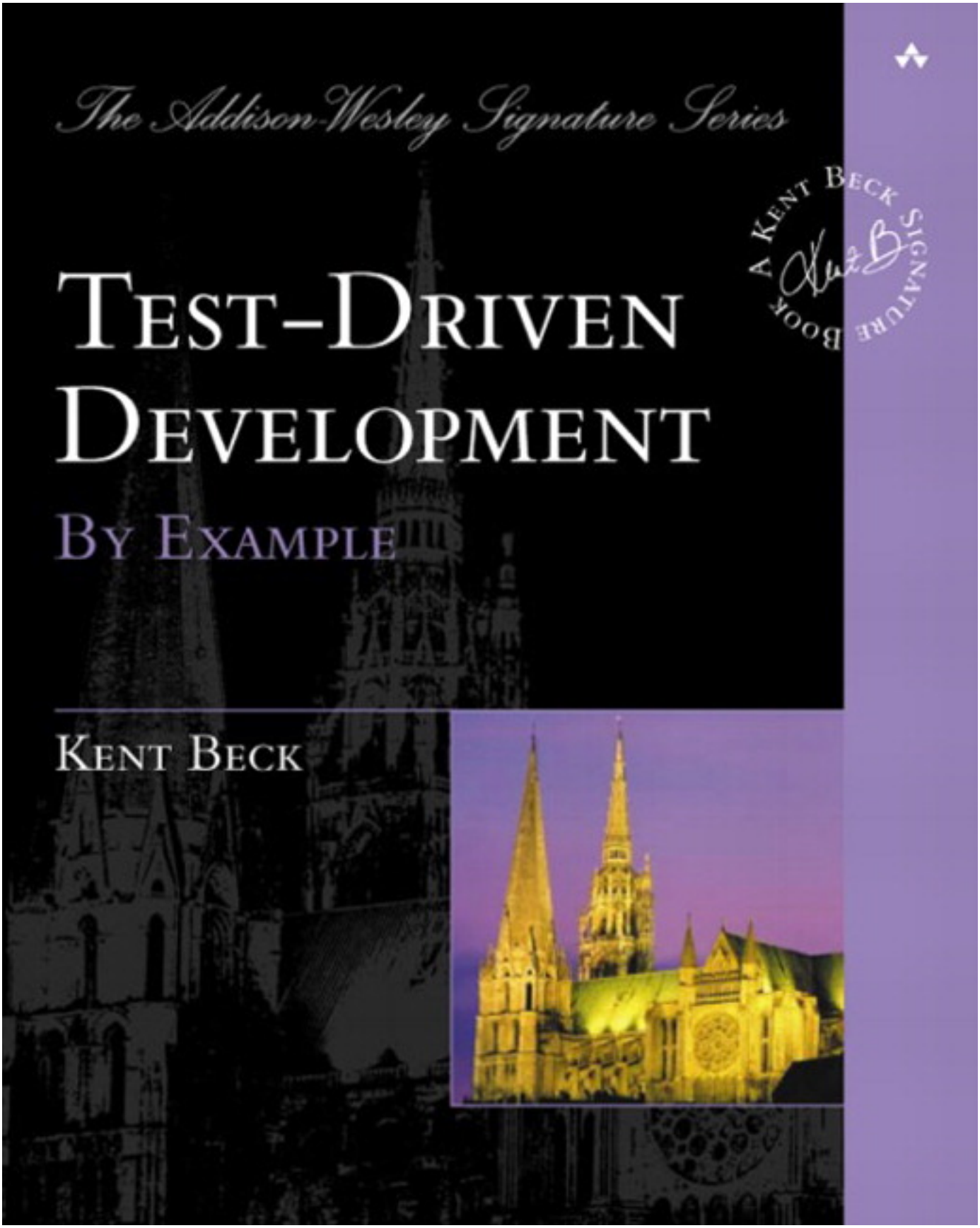
par Matthieu Brucher (http://matthieu-brucher.developpez.com/) (Blog)

Date de publication : 15/05/2006

Dernière mise à jour : 28/08/2006

Critique du livre Test-Driven Development de *Kent Beck*

I - Description



Du code propre qui fonctionne - tout de suite. Ceci est apparemment une contradiction responsable de beaucoup de souffrances en programmation. Test-Driven Development, ou développement basé sur les tests, répond à cette contradiction par un paradoxe : testez le programme avant de l'écrire.

Une nouvelle idée ? Pas du tout. Depuis l'aube des calculs sur ordinateur, les programmeurs ont spécifié les entrées et les sorties de leur programme avant de l'écrire précisement. TDD reprend cette vieille idée, la mixe avec les langages modernes et les environnements de programmation et en fait un mets vous garantissant d'être rassasié de code clair qui fonctionne - tout de suite.

Les développeurs font tous les jours face à des défis, pourtant ils ne sont pas toujours préparés à trouver la meilleure solution. Le plus souvent, de grands projets génèrent beaucoup de stress et du mauvais code. Pour obtenir la force et le courage de surmonter cette tâche apparemment herculéenne, les programmeurs devrait regarder vers TDD, un jeu de techniques qui encourage les designs simples et les tests qui inspirent confiance.

En développant avec des tests automatiques puis en éliminant les doublons, chaque développeur peut écrire du code dans lequel on peut avoir confiance, sans bug quelque soit sa complexité. De plus, TDD encourage les programmeurs à apprendre rapidement, communiquer plus clairement et à obtenir un feedback constructif.

Les lecteurs apprendront à:

- Résoudre des tâches compliquées, commençant par la simplicité et continuant vers plus de complexité.
- Ecrire des tests automatiques avant de programmer
- Définir un design en refactorant pour ajouter des éléments petit à petit.
- Créer des tests pour de la logique plus complexe.
- Utiliser des patterns pour décider quels tests doivent être écrits.
- Créer des tests à l'aide d'xUnit, l'architecture au coeur de plusieurs outils de tests orientés vers les programmeurs.

Ce livre suit 2 projets à l'aide de TDD, du début à la fin, illustrant les techniques que les programmeurs peuvent utiliser pour améliorer facilement et de manière importante la qualité de leur travail. Les exemples sont suivis de références aux patterns TDD et au refactoring. Avec son accent sur les méthodes dites *agiles* et les stratégies de développement rapide, TDD encouragera les lecteurs à embrasser ces techniques sous-utilisées mais puissantes.

II - Table des matières

III - Critique : A appliquer d'urgence partout !

On parle Java dans tout le livre, mais chaque conseil est pour tous les langages. De plus, le framework a été porté pour fonctionner avec presque tous les langages. L'approche de *Kent Beck* est agréable et didactique. On construit vraiment au fur et à mesure, petit pas par petit pas. Le premier exemple est simple, gérer de l'argent dans différentes devises, et ce petit exemple nous mène déjà loin.

En fait, les exemples servent juste à bien nous asséner le principe de *Beck*, tester en construisant progressivement. Si on a compris qu'on doit toujours tout tester et qu'aucun code ne doit être buggé quand on arrête le travail, on a compris le livre. Ensuite, l'auteur nous donne des astuces, des *patterns* pour avancer, comme écrire des tests isolés, ne dépendant pas de toute l'architecture afin de ne pas déprimer - important... -, faire des raisonnements inductifs pour généraliser une réponse, ...

Depuis la lecture de ce livre, je me suis mis aux tests. J'avoue qu'avant, c'était la dernière de mes préoccupations. Maintenant, mon code est plus propre - on dirait une pub de lessive, là -, et surtout bardé de tests au cas où j'aurai une modification au niveau de l'architecture à faire - refactoring, et ça arrive souvent -. En même temps, je n'applique pas sa solution à 100%, c'est tout de même très difficile, ça demande du temps d'écrire le test avant d'avoir la solution qui va avec, mais on voit aussi comment on avance, c'est bien ;)

IV - Liens annexes

Critique sur la page de livres Conception

Achat sur Amazon.fr

Lien vers le site de l'éditeur