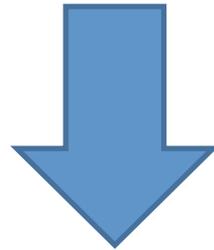


Développement piloté par les tests - TDD

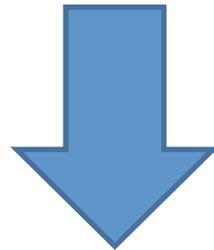
Gestion de projet

- Origine du TDD
- Principe du TDD
- Cycle de développement
- Stratégies d'application
- Idées reçues et limites
- Conclusion

Test First – Extreme Programming – 1996

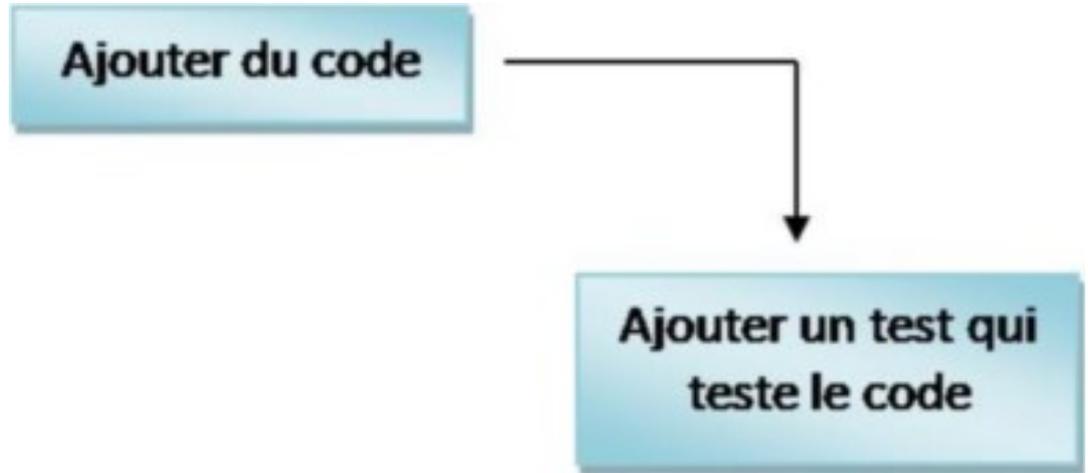


Test Driven - 1996 à 2002

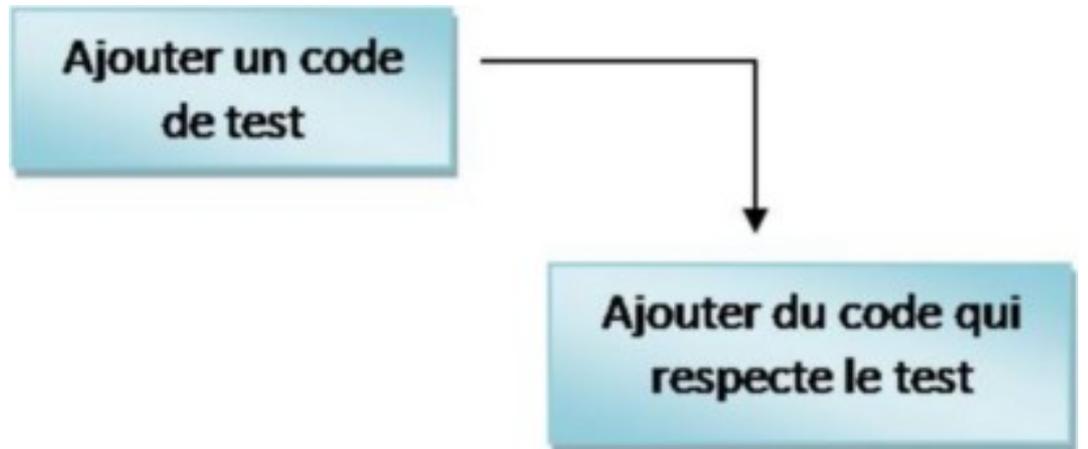


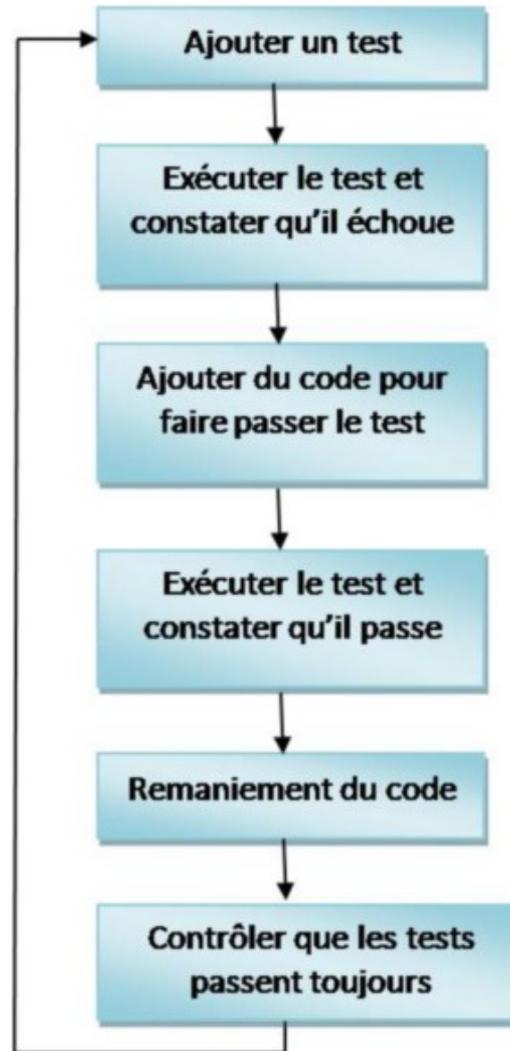
Test Driven Development - 2003

- Méthode traditionnelle



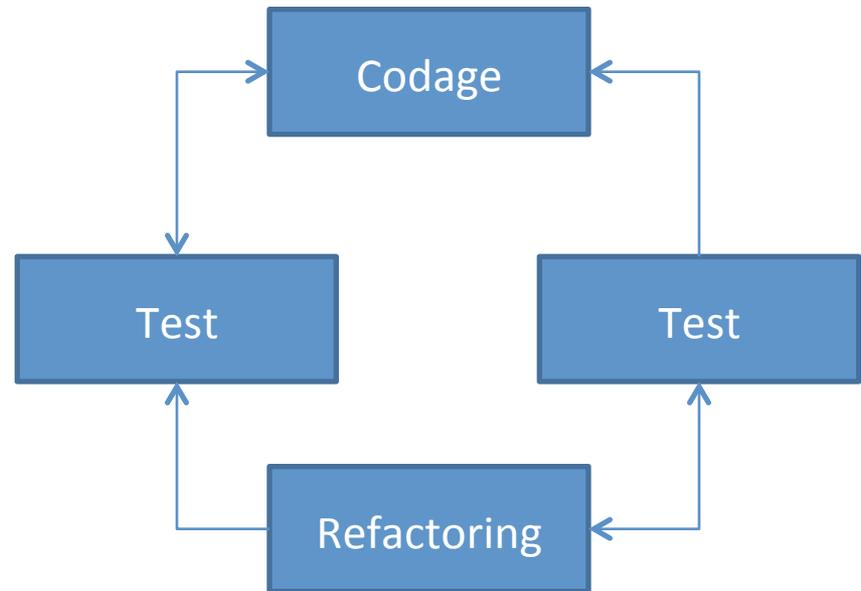
- Méthode TDD





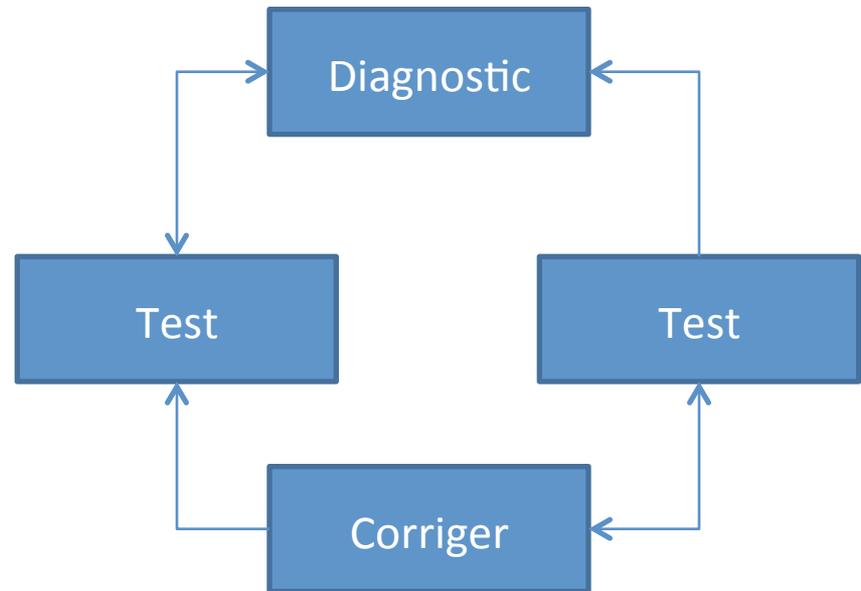
Développement

- Technique itérative
- Tests réguliers



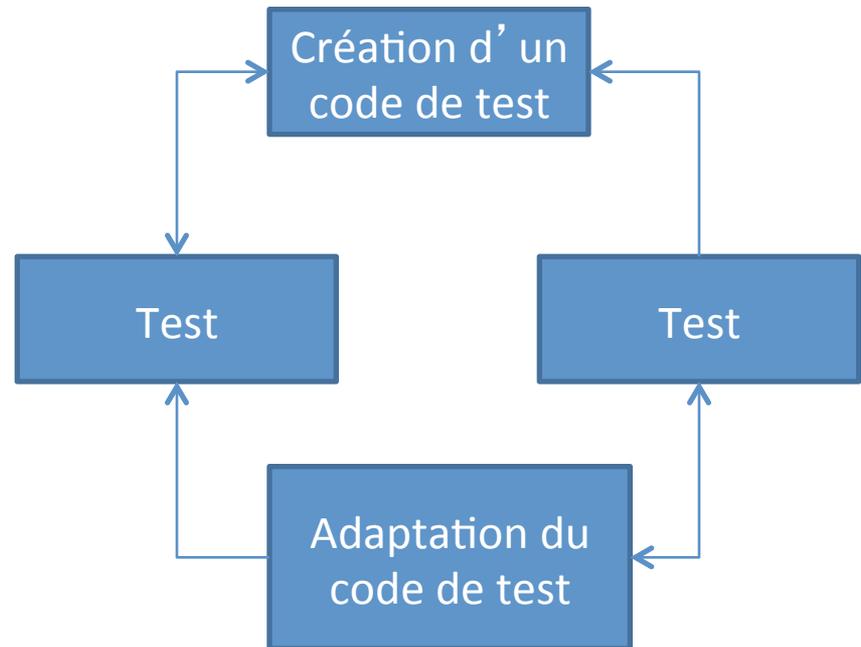
Débogage

- Analyse du problème
- Création d'un test le décrivant
- Correction du bug
- Tests de validation



Code hérité – Legacy code

- Étude du code existant
- Écriture d'un test pour confirmer le bug
- Correction du code jusqu'à validation du test



- Concevoir les tests comme des spécifications
- Forte implication du client
- Outils pour mettre en place ces tests

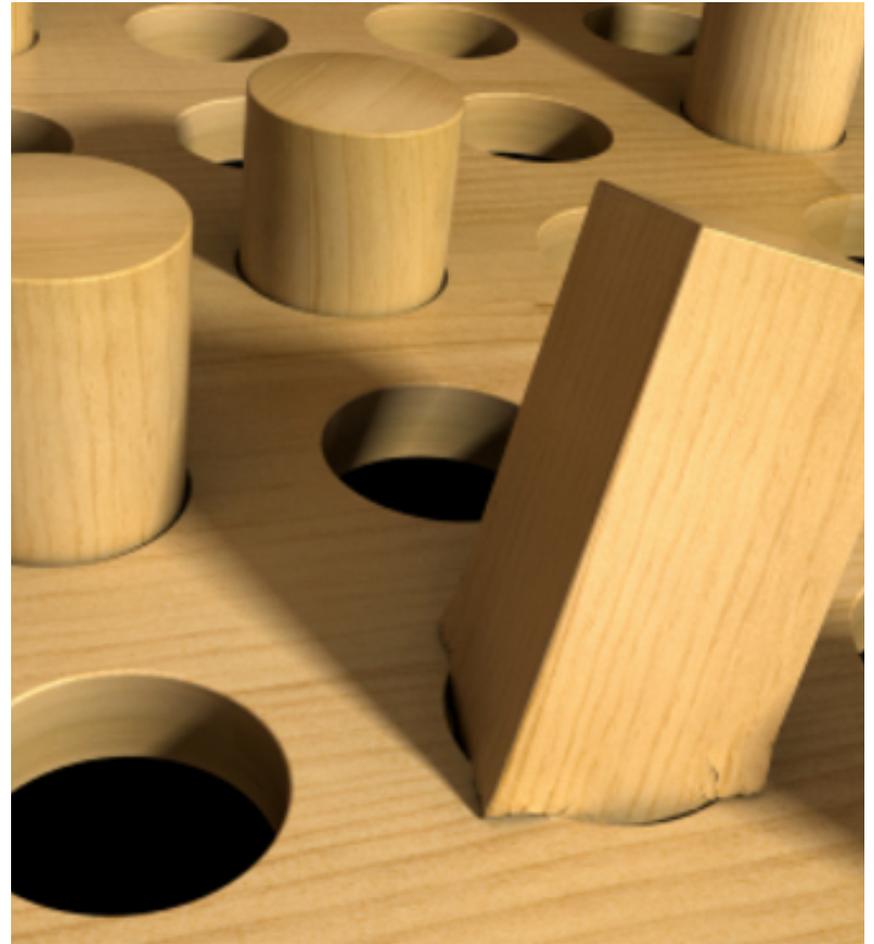


- Ce n'est pas mon travail
- Tester prend du temps
- Tester ou faire des essais, ça revient au même
- Tester c'est ennuyeux
- Tester ne paie pas à long terme



- TDD n'est pas une solution miracle
- TDD n'empêche pas de commettre des erreurs
- Un certain nombre de pièges attendent le développeur

- Le dégueulasse
- L'escargot
- L'escroc
- Le héros local
- L'inspecteur



- Augmente la qualité du code
- Respecte les spécifications
- Inspire d'autres méthodes comme *Behaviour Driven Development*
- Quelques pièges à éviter