Lean code

Une idée de Chris Parsons

Lean

L'école de gestion de la production dite ***lean*** recherche la performance (en matière de productivité, de qualité, de délais, et enfin de coûts) par l'amélioration continue et l'**élimination des gaspillages**, au nombre de sept :

- production excessive,
- attentes,
- transport et manutention inutiles,
- tâches inutiles,
- stocks,
- mouvements inutiles
- production défectueuse.

L'école de gestion *lean* trouve ses sources au Japon dans le Toyota Production System (TPS). Adaptable à tous les secteurs économiques, le *lean* est actuellement principalement implanté dans l'industrie (et surtout l'industrie automobile).

Application

Le modèle, basé sur le développement itératif et les méthodes agiles met en avant 7 principes :

1. Eliminer les gaspillages : comme pour le lean, le gaspillage est défini comme ce qui n'apporte pas de valeur au produit. La valeur étant définie du point de vue de l'utilisateur.
2. Améliorer l'apprentissage.
3. Retarder l'engagement.
4. Livrer aussi vite que possible.
5. Donner le pouvoir à l'équipe.
6. Intégrer la qualité dès la conception.
7. Considérer le produit dans sa globalité.

Essayons !

MONOPRIX

Itération 0

10 minutes pour trouver un binôme et mettre en place l'environnement de développement

Nous développons un outil en ligne de commande :
Lire l'entrée standard et écrire en sortie.

Utilisez le langage de votre choix

Une base est disponible en C et Ruby à l'adresse suivante :
https://github.com/guillaumerose/checkout-base

Itération 1

10 minutes pour faire une caisse enregistreuse simple.

Les Pommes coûtent 1€, les Bananes 1.50€ et les Cerises 0.75€.

Acceptez un article par ligne. Afficher le total du panier (en centimes) à chaque fois.

Test d'acceptance :

- ➡ Pommes
- ➡ 100
- ➡ Cerises
- ➡ 175
- ➡ Cerises
- ➡ 250

Livraison 1

Vérification

- ➡ Cerises
- ➡ 75
- ➡ Pommes
- ➡ 175
- ➡ Cerises
- ➡ 250
- ➡ Bananes
- ➡ 400
- ➡ Pommes
- ➡ 500

Itération 2

10 minutes pour ajouter les réductions.

Même principe pour la gestion de l'entrée et de la sortie.

2 lots de cerises achetés, 20 centimes de réduction.

Test d'acceptance :

- ➡ Pommes
- ➡ 100
- ➡ Cerises
- ➡ 175
- ➡ Cerises
- ➡ ~~250~~ 230

Livraison 2

Vérification

- ➡ Cerises
- ➡ 75
- ➡ Pommes
- ➡ 175
- ➡ Cerises
- ➡ 230
- ➡ Bananes
- ➡ 380
- ➡ Cerises
- ➡ 455
- ➡ Cerises
- ➡ 510
- ➡ Pommes
- ➡ 610

Itération 3

10 minutes pour ajouter le support du format CSV en entrée.

Les articles peuvent être séparés par des virgules.

Test d'acceptance :

- ➡ Pommes, Cerises, Bananes
- ➡ 325
- ➡ Pommes
- ➡ 425

URGENT

Itération 3A

Changement de programme ! La deadline ne change pas.
Le support du format CSV est repoussé à plus tard.
Un article par ligne suffit pour cette itération.

La réduction pour les cerises passe à 30 centimes.
Un lot de bananes acheté, le second offert.

Test d’acceptance :

- ➡ Cerises
- ➡ 75
- ➡ Cerises
- ➡ 120
- ➡ Bananes
- ➡ 270
- ➡ Bananes
- ➡ 270

Livraison 3

Vérification

- ➡ Cerises
- ➡ 75
- ➡ Pommes
- ➡ 175
- ➡ Cerises
- ➡ 220
- ➡ Bananes
- ➡ 370
- ➡ Pommes
- ➡ 470
- ➡ Bananes
- ➡ 470
- ➡ Cerises
- ➡ 545

Itération 4

10 minutes pour ajouter le support de plusieurs langues.

La réduction pour les cerises repasse à 20 centimes.
Les mots « Apples » et « Mele » correspondent aux Pommes.

Test d'acceptance :

- ➡ Cerises
- ➡ 75
- ➡ Apples
- ➡ 175
- ➡ Cerises
- ➡ 230
- ➡ Bananes
- ➡ 380
- ➡ Bananes
- ➡ 380

Livraison 4

Vérification

- ➡ Cerises
- ➡ 75
- ➡ Apples
- ➡ 175
- ➡ Cerises
- ➡ 230
- ➡ Bananes
- ➡ 380
- ➡ Pommes
- ➡ 480
- ➡ Mele
- ➡ 580

Itération 5

10 minutes pour permettre différents types de réduction selon la langue pour chaque article.

Le support du CSV sera demandé à la prochaine itération.
3 lots de « Apple » valent 2€
2 lots de « Mele » valent 1.50€

Test d'acceptance :

- ➡ Mele
- ➡ 100
- ➡ Apples
- ➡ 200
- ➡ Apples
- ➡ 300
- ➡ Pommes
- ➡ 400
- ➡ Apples
- ➡ 400
- ➡ Mele
- ➡ 450
- ➡ Cerises
- ➡ 525
- ➡ Cerises
- ➡ 580

URGENT

Itération 5A

Un déploiement a mal tourné sur des nouvelles caisses.
On tente de résoudre le problème.

10 minutes pour permettre différents types de réduction selon la langue pour chaque article.

3 lots de « Apple » valent 2€
2 lots de « Mele » valent 1€

Test d'acceptance :

- ➡ Mele, Apples, Apples, Pommes, Apples, Mele, Cerises, Cerises, Bananes
- ➡ 680

Livraison 5

Vérification

- ➡ Cerises, Apples
- ➡ 175
- ➡ Cerises
- ➡ 230
- ➡ Apples, Mele, Bananes
- ➡ 580
- ➡ Apples, Pommes
- ➡ 680
- ➡ Mele
- ➡ 680
- ➡ Pommes
- ➡ 780

Itération 6

10 minutes pour fixer un bug et créer la super-réduction.

Bug : supporter CSV et les articles un par un
2 lots de « Mele » valent 1.50€
4 Pommes achetés, 1€ de réduction sur la facture globale
5 fruits achetés, 2€ de réduction

Tests d'acceptance :

- ➡ Mele, Apples, Apples, Mele
- ➡ 250

- ➡ Bananes
- ➡ 200

- ➡ Mele, Apples, Apples, Pommes, Mele
- ➡ 150

Livraison 6

Vérification

- ➡ Mele, Apples, Apples, Pommes, Mele
- ➡ 150
- ➡ Bananes
- ➡ 350

Conclusions

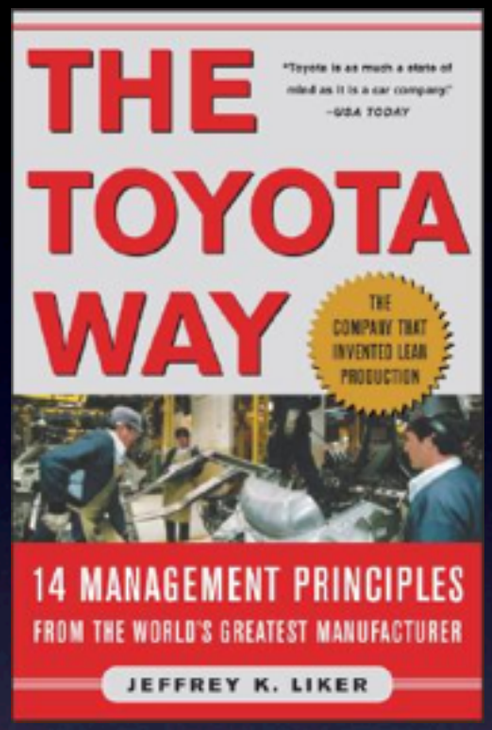
Avez-vous utilisé un système de contrôle de versions ?
Auriez-vous voulu retourner en arrière ? CTRL+Z ?

Avez-vous utilisé des tests unitaires ? A partir de quelle itération ?

Avez-vous écrit et maintenu tous les tests d'acceptance ?
Vous êtes vous fiés uniquement aux spécifications ?

Qu'avez-vous fait pour le support du CSV lorsqu'il n'était plus nécessaire ?

Plus sur le concept « Lean »



En parler ?
Paris Software Craftsmanship
@swcraftparis

Présentation adaptée
http://chrismdp.github.com/2011/05/lean-code-slides-and-feedback/