



## :: Méthodologie XP ::

---

LT La Salle Avignon [2004 tv]

---



## Introduction

---

- **L'eXtremme Programming (XP)** est un ensemble de pratiques qui couvrent une grande partie des activités de la réalisation d'un logiciel :
    - ♦ planification
    - ♦ organisation de l'équipe de développement
    - ♦ échanges avec le client
    - ♦ Programmation.
  - XP est une méthode de développement **légère** et qui pousse ses pratiques à l'**extrême** pour réaliser les objectifs :
    - ♦ Développer vite et juste
-



## La méthodologie XP

---

- La méthodologie XP se décompose de la manière suivante :
    - Les **phases** (exploration, engagement et pilotage)
    - Les **variables** (le coût, le temps, la qualité et l'étendue des fonctionnalités)
    - Les **valeurs** (la communication, la simplicité, le *feedback* et le courage)
    - Les **activités** (coder, intégrer, tester, écouter et modéliser)
    - Les **13 pratiques** (conception simple, remaniement, tests unitaires, tests de recette, programmation en binôme, responsabilité collective du code, règles de codage, métaphore, intégration continue, livraisons fréquentes, planification itérative, client sur site et rythme durable)
- 



## Les phases

---

- Chaque cycle XP doit produire une livraison fonctionnelle (*release*).
  - Un cycle de développement XP est décomposé en 3 moments (ou **phases**) :
    - L'**exploration** : définition des besoins du moment.
    - L'**engagement** : choix des besoins qui vont être implémentés en fonction de la valeur que leur accorde le client.
    - Le **pilotage** : implémentation des besoins et *feedback* immédiat et impact du produit afin d'orienter la phase d'exploration du cycle suivant.
-



## Les variables

---

- XP considère que la satisfaction de 4 variables va contribuer au développement efficace d'un produit fonctionnel. Ces variables sont :
    - Le **coût** : le développement se fait par petits modules, cela permet de suivre de près les dépenses pour ne pas dépasser les budgets.
    - Le **temps** : le développement par modules permet de suivre l'évolution du développement en fonction du temps passé.
    - La **qualité** : pour XP, la qualité doit toujours être maximum. C'est la seule variable qui ne peut pas varier.
    - L'**étendue** : le nombre de fonctionnalité va dépendre du temps et du budget à disposition. C'est le client qui va décider de ce qui sera développé ou pas afin de garder la plus grande qualité possible, en restant dans les délais et les budgets.
- 



## Les valeurs

---

- Afin de satisfaire les 4 variables, XP met en évidence 4 valeurs essentielle au travers de la méthode :
    - la **communication**,
    - la **simplicité**,
    - le *feedback* et
    - Le **courage**.
-



## Les activités

---

- Les activités de XP sont les étapes minimales garantissant la réussite du développement du projet :
    - ♦ **coder** et **intégrer** pour produire une livraison pour le client
    - ♦ **tester** pour rendre la livraison réellement fonctionnelle
    - ♦ **écouter** les besoins du client, ainsi que les changements qu'il veut apporter.
    - ♦ **modéliser** (faire du *design*) afin de réaliser l'architecture la plus simple possible.
- 



## Les pratiques de programmation

---

- **Conception simple** (*simple design*) : les développeurs implémentent toujours la solution la plus simple qui puisse fonctionner.
  - **Remaniement** (*refactoring*) : les développeurs n'hésitent pas à revenir sur le code écrit pour le rendre plus «propre» et plus simple
  - **Développement piloté par les tests unitaires** (*test-first programming, unit tests, developer tests*) : les développeurs écrivent des tests automatiques pour le code qu'ils produisent avant même d'écrire le code en question.
  - **Tests de recette** (*acceptance tests, customer tests*) : le client précise très explicitement ses besoins et les objectifs des programmeurs en participant à la rédaction de tests de recette. Comme les tests unitaires, les tests de recette doivent être automatiques afin de pouvoir vérifier tous les jours la non-régression du produit.
-



## Les pratiques de collaboration

---

- **Programmation en binôme** (*pair programming*) : lorsqu'ils écrivent le code de l'application, les développeurs travaillent systématiquement à deux sur la même machine.
  - **Responsabilité collective du code** (*collective code ownership*) : tous les développeurs de l'équipe sont amenés à travailler sur toutes les parties de l'application, avec le devoir d'améliorer le code sur lequel ils interviennent, même s'ils n'en sont pas les auteurs initiaux.
  - **Règles de codage** (*coding standards*) : les développeurs se plient à des règles de codage définies par l'équipe elle-même, afin de rendre le code homogène avec le reste de l'application, et ainsi à faciliter l'intervention d'autres développeurs.
  - **Métaphore** (*metaphor*) : les développeurs n'hésitent pas à recourir aux métaphores pour décrire la structure interne du logiciel
  - **Intégration continue** (*continuous integration*) : les développeurs synchronisent leurs développements aussi souvent que possible, au moins une fois par jour.
- 



## Les pratiques de gestion de projet

---

- **Livraisons fréquentes** (*frequent releases*) : l'équipe livre des versions du logiciel à un rythme régulier, aussi élevé que possible, la fréquence précise étant fixée par le client.
  - **Planification itérative** (*planning game*) : la planification du projet est réalisée conjointement par le client et l'équipe de développement, au cours de séances dédiées, organisées régulièrement tout au long du projet.
  - **Client sur site** (*on-site customer, whole team*) : le client est littéralement intégré à l'équipe de développement pour arbitrer les priorités, et définir précisément ses besoins, en bénéficiant du *feedback* immédiat grâce aux livraisons fréquentes.
  - **Rythme durable** (*sustainable pace*) : l'équipe adopte des horaires qui lui permettent de conserver tout au long du projet l'énergie nécessaire pour produire un travail de qualité et mettre en oeuvre efficacement les autres pratiques.
-



## Le cycle standard d'XP

---

1. Le client écrit ses besoins sous forme de scénarios.
  2. Les développeurs évaluent le coût de chaque scénario, en collaboration avec le client.
  3. Le client choisit les scénarios à intégrer à la prochaine livraison.
  4. Chaque développeur prend la responsabilité d'une tâche pour la réalisation d'un scénario.
  5. Le développeur choisit un partenaire.
  6. Le binôme écrit les tests unitaires correspondant au scénario à implémenter.
  7. Le binôme prépare l'implémentation en réorganisant le code existant, puis il procède à l'implémentation proprement dite.
  8. Le binôme intègre ses développements à la version d'intégration.
- 



## Planification collective

---

- Le projet est décomposée en une suite de « tours » appelés **itérations** (durée de 1 à 3 semaines selon les projets) et de **livraisons** au rythme d'une livraison toutes les 2 ou 3 itérations en général.
  - Le **cycle des livraisons** porte sur la planification des fonctionnalités décrites sous forme de scénarios client (*user stories*).
  - Le **cycle des itérations** porte sur la planification des activités (tâches) menées par les développeurs.
  - Les **tâches** sont donc les activités des développeurs qui permettent de réaliser un scénario client.
-