

Plan

1. Historique
2. Définition
3. Les valeurs d'un projet XP
4. Les principales pratiques de l'XP
5. Cycle de vie d'un projet XP
6. Membres d'équipe et rôles
7. Avantages et inconvénients de l'XP

Historique

- D'après une étude américaine menée en 1994 sur 8000 projets, 16% n'ont pas respecté les délais et le budget initial et, pire, 32% n'ont jamais abouti:
 - Le planning difficilement maîtrisé ou impliquant de nombreuses heures sup;
 - Des besoins souvent mal identifiés ou mal compris en raison d'un cahier des charges mal étudié ou incomplet;
 - La livraison finale du produit qui est parfois buguée.

Historique

- XP est apparu durant une période de forte croissance de l'informatique en entreprise et de l'émergence du freeware.
- S'inscrit dans la famille des méthodes agiles, développées à cette époque.
 - Création officielle : XP a été créé en 1996 par Kent Beck.
 - Origine : Développé lors d'un projet pour Chrysler (projet C3) pour mieux répondre aux besoins clients.
 - Diffusion : En 1999, Beck publie "*Extreme Programming Explained*", qui popularise la méthode XP.



Définition

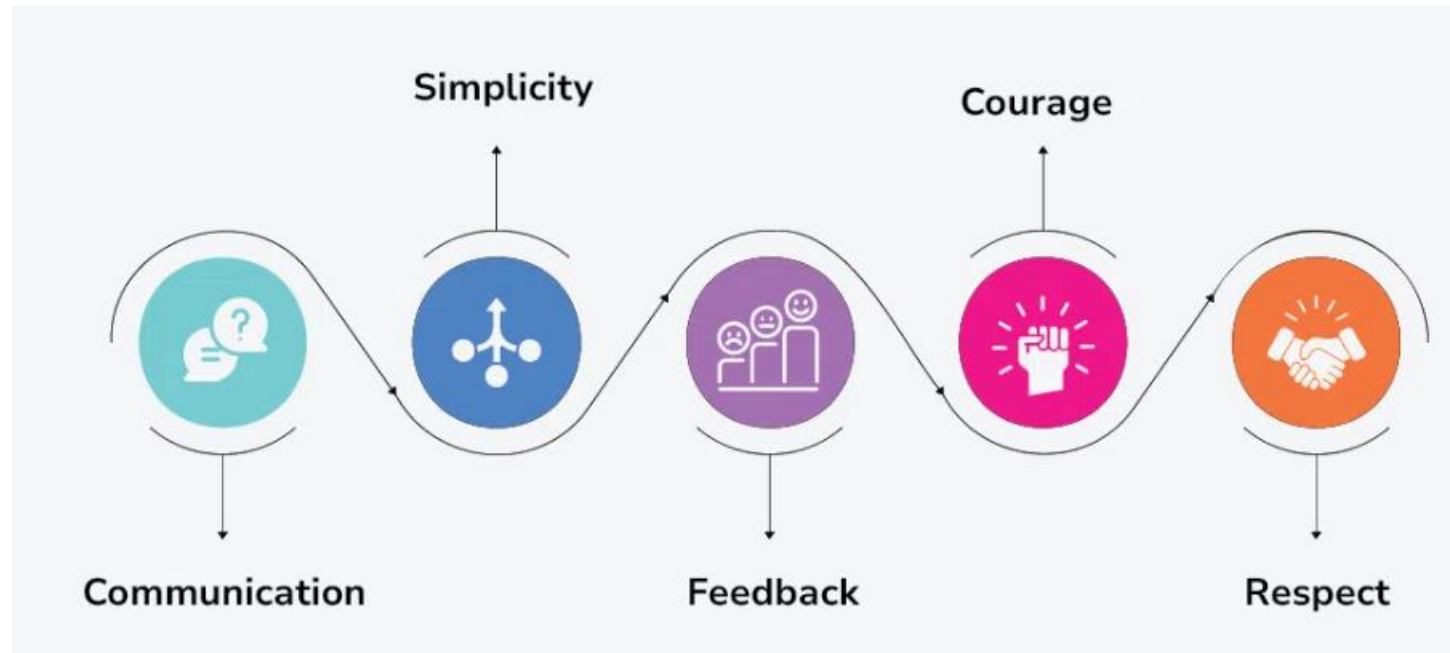
■ Extreme Programming (XP)

- XP est une méthodologie efficace de développement rapide de logiciels qui exige que les membres d'une équipe de développeurs soient physiquement proches les uns des autres.
- Aider les petites et moyennes équipes à produire des logiciels de haute qualité tout en s'adaptant rapidement aux exigences changeantes.
- XP encourage des versions fréquentes, un travail collaboratif et un haut niveau d'implication du client.

■ Objectifs principaux :

- Développer vite : Garder les changements faciles pour maintenir la vitesse du projet.
- Développer juste : Se concentrer sur les besoins réels du client.

Les valeurs d'un projet XP



Les valeurs d'un projet XP

➤ La Communication :

- L'absence de communication est certainement l'un des défauts les plus graves qui mettent en péril un projet.
- Diverses pratiques XP tendent à rendre la communication omniprésente entre tous les intervenants :
 - entre développeurs (programmation en binôme),
 - entre développeurs et managers (tests, estimations),
 - entre développeurs et clients (tests, spécifications).

Les valeurs d'un projet XP

➤ La Simplicité :

- Cette valeur repose sur l'idée qu'il est moins coûteux de développer un système simple aujourd'hui, quitte à ajouter des fonctionnalités plus tard si nécessaire.
- Concevoir un système complexe dès le départ peut entraîner des coûts inutiles.
- Un système simple est souvent plus adapté aux besoins actuels et évolutifs.

Les valeurs d'un projet XP

➤ Le Feedback :

- Le retour est immédiat pour les développeurs grâce aux tests unitaires.
- Les clients reçoivent un retour en quelques jours grâce aux tests fonctionnels, offrant une vision continue de l'état du système.
- Avantages d'un feedback permanent :
- Pour le client :
 - Avoir une vue claire de l'avancement du projet.
 - Détecter rapidement les écarts par rapport au planning et aux attentes.
 - Corriger les écarts rapidement.
- Pour les développeurs :
 - Identifier et corriger les erreurs plus facilement.

Les valeurs d'un projet XP

➤ **Le Courage :**

- Le courage est essentiel pour le client et les développeurs dans un projet XP.
- Le client doit avoir le courage de:
 - Prioriser clairement ses exigences.
 - Reconnaître que certains besoins peuvent être flous.
- Le développeur doit avoir le courage de:
 - Modifier l'architecture même à un stade avancé du développement.
 - Supprimer du code existant et accepter qu'il est parfois plus rapide et efficace de réécrire une portion de code plutôt que d'adapter l'ancien.

Les valeurs d'un projet XP

➤ **Le Respect :**

- Les relations entre l'équipe de développement et le client et même entre eux doivent être basées sur le respect mutuel.

Les principaux pratiques de l'XP

1. Planning game
2. Petites Releases.
3. Utilisation des métaphores.
4. Conception simple.
5. Tests (unitaires et fonctionnels).
6. Refactoring du code.
7. Programmation en binôme.
8. Appropriation collective du code.
9. Intégration continue.
10. Pas de surcharge de travail.
11. Client sur site.
12. Standards de code.

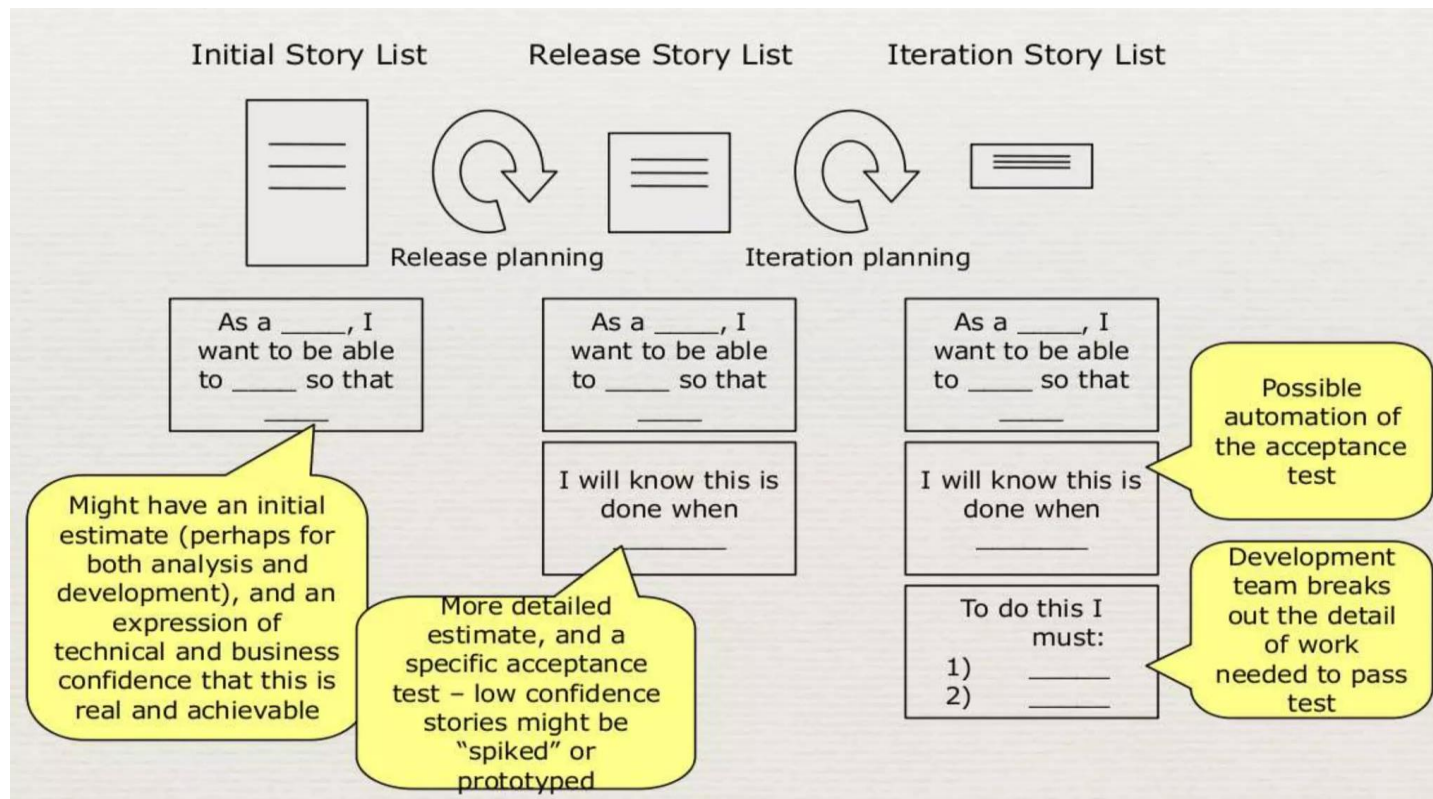
Les principaux pratiques de l'XP

1. Planning game

- Cette pratique a pour but de planifier uniquement les releases. La planification se fait sous forme de jeu auquel participent les développeurs et le client . Cette pratique est constituée de deux phases :
 - Exploration : Le client exprime ses besoins en fonctionnalités, rédigées sous forme de "user stories".
 - Engagement : Les "user stories" sont triées selon leur valeur pour le client et les risques, puis classées par priorité.
 - Direction : Mise à jour du planning pour la prochaine release.

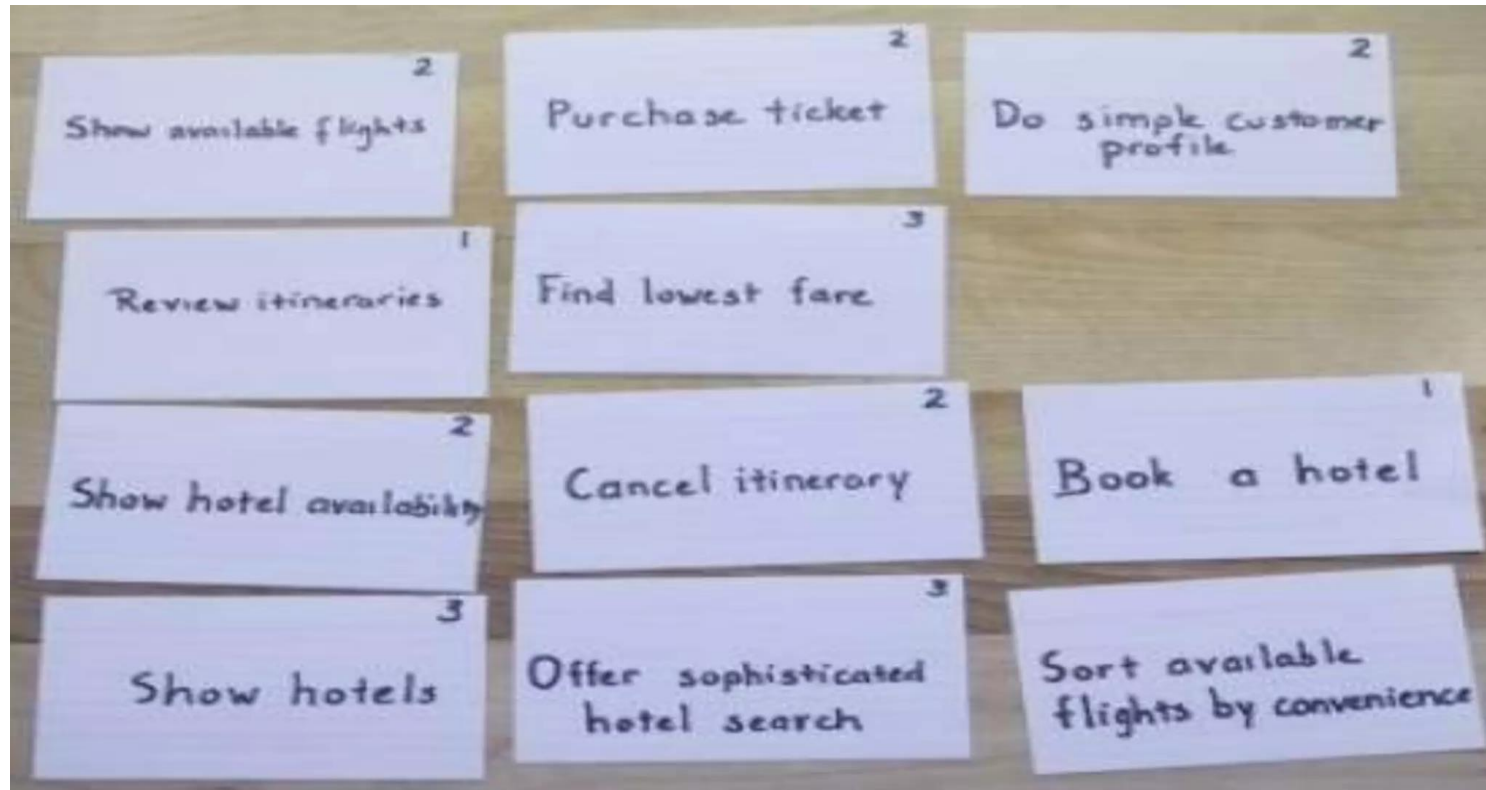
Les principaux pratiques de l'XP

➤ Exemple sur «User Stories »



Les principaux pratiques de l'XP

➤ Exemple sur «L'engagement »



Les principaux pratiques de l'XP

2. Petites Releases

- Gestion des risques : Les releases doivent être fréquentes pour mieux gérer les risques.
- Évolution minimale : Chaque nouvelle version doit apporter de petites améliorations, mais avec un maximum de valeur ajoutée.
- Fonctionnalités complètes : Intégrer des fonctionnalités finies et prêtes à l'emploi dans chaque release.

Les principales pratiques de l'XP

3. Utilisation des métaphores

- Utiliser des métaphores pour décrire l'architecture du système.
- Avoir une vision globale du système (comprendre ses éléments ainsi que leurs interactions).

Les principaux pratiques de l'XP

4. Conception simple

- Développer la solution la plus simple possible et éviter de développer plus que ce dont on a besoin.
- Exigences de XP :
 - Satisfaire tous les tests.
 - Pas de duplication de logique.
 - Utiliser le minimum de classes et de méthodes nécessaires.
- Principe YAGNI : "You Ain't Gonna Need It" – éviter d'ajouter des fonctionnalités non essentielles.

Les principaux pratiques de l'XP

5. Tests (unitaires et fonctionnels)

- Les tests unitaires sont écrits et effectués pour vérifier le bon fonctionnement des méthodes et des constructeurs.
- Les tests fonctionnels sont conçus par le client pour:
 - Vérifier le fonctionnement global du système,
 - Contrôler l'évolution du projet,
 - Affiner l'expression de ses besoins.
- Il est recommandé d'écrire les tests avant le code de l'application.

Les principaux pratiques de l'XP

6. Refactoring du code

- Simplifier et structurer le code, en veillant à ce que tous les tests soient satisfaits.
- Faciliter l'ajout de nouvelles fonctionnalités.
- Produire un code plus réfléchi, modulaire et sans duplications.
- Rendre le code plus facile à maintenir.

Les principales pratiques de l'XP

7. Programmation en binôme

- Deux personnes écrivent le code ensemble sur une seule machine.
- Rôles distincts :
 - Pilote ("driver") : Celui qui utilise le clavier et se concentre sur une portion précise du code.
 - Partenaire ("partner") : Observe avec recul, propose des solutions alternatives et identifie les problèmes plus généraux.
- Les binômes ne doivent pas être statiques : chacun change de partenaire relativement pour un meilleur partage des connaissances



Les principaux pratiques de l'XP

8. Appropriation collective du code

- Toute l'équipe est sensée connaître la totalité du code.
- Cela implique que tout le monde peut intervenir pour faire des ajouts ou des modifications sur une portion de code qu'il n'a pas écrite lui-même si cela s'avère nécessaire.

Les principaux pratiques de l'XP

9. Intégration continue

- Après chaque fin de tâche, le code nouvellement écrit doit être intégré à l'existant de manière à avoir à tout moment un existant fonctionnel qui passe avec succès tous les tests.

Les principales pratiques de l'XP

10. Pas de surcharge de travail

- L'objectif est de ne pas dépasser 40 heures de travail par semaine pour les développeurs.
 - Une équipe ne doit jamais être surchargée plus de deux semaines d'affilée.
 - Respecter la **vélocité** de l'équipe (rythme de travail soutenable).
- Ce n'est pas une question de travailler moins, mais de reconnaître que la surcharge de travail est nuisible et désagréable.

Les principaux pratiques de l'XP

11. Client sur site

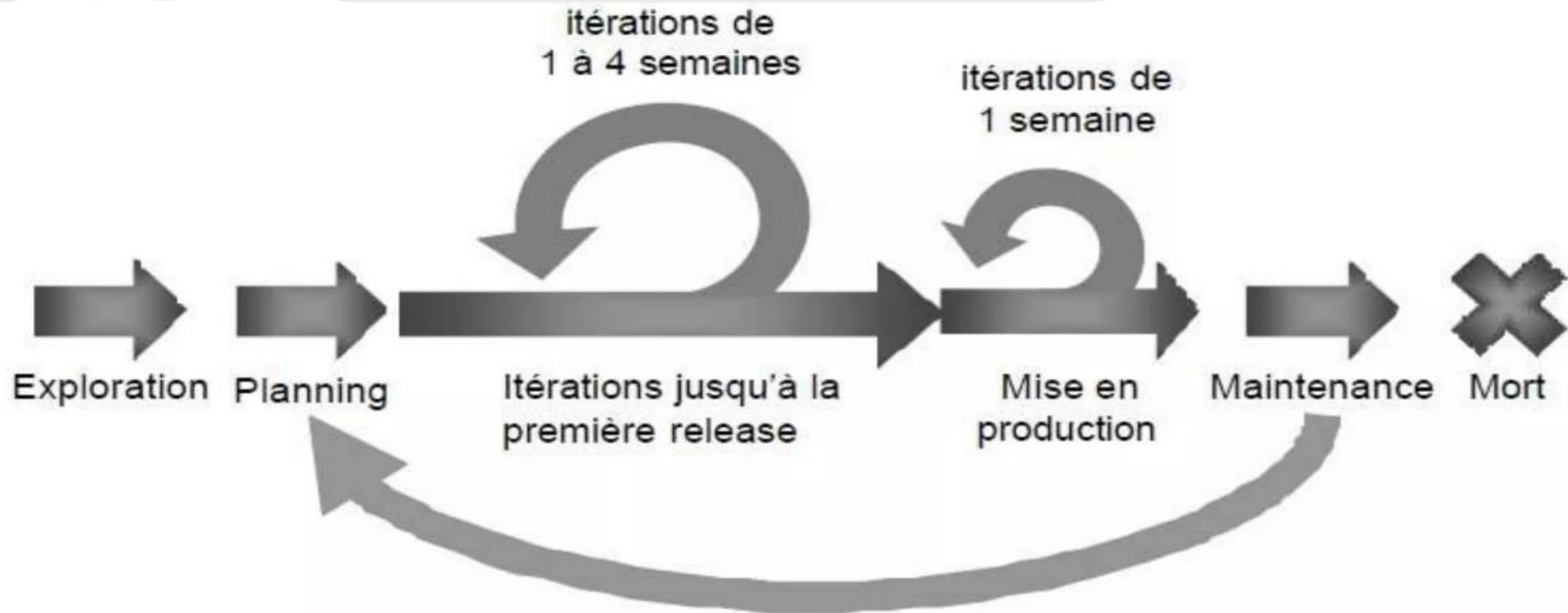
- Une personne représentant le client est intégrée à l'équipe et doit être présente à temps plein pendant toute la durée du projet.
- Rôle du client :
 - Avoir une vision globale du projet.
 - Préciser les besoins et fixer les priorités (prioriser les user stories).
 - Établir les tests fonctionnels.

Les principaux pratiques de l'XP

12. Standards de code

- Il est nécessaire de disposer de normes de nommage et de programmation pour que chacun puisse lire et comprendre facilement le code produit par les autres.

Cycle de vie d'un projet XP



Cycle de vie d'un projet XP

■ Exploration

- Les développeurs se penchent sur des questions d'ordre technique destinées à explorer les différentes possibilités d'architecture pour le système.
- Le client s'habitue à exprimer ses besoins sous forme de user stories que les développeurs devront estimer en termes de temps de développement.

■ Planning

- Le planning de la première release (uniquement des fonctionnalités essentielles), à enrichir par la suite.
- Le planning game dure un ou deux jours et la première release est en général programmée pour deux à six mois plus tard.

Cycle de vie d'un projet XP

- Itérations jusqu'à la première release
 - Phase de développement en itérations de 1 à 4 semaines.
 - Chaque itération livre un ensemble de fonctionnalités validées par les tests fonctionnels.
 - La première itération est consacrée à la mise en place de l'architecture du système.
 - Des réunions quotidiennes rassemblent l'équipe pour partager l'avancement du projet.
- Mise en production
 - Itérations plus courtes pour renforcer le feedback.
 - Les développeurs effectuent des réglages fins pour améliorer les performances.
 - Résultat : un produit complet avec toutes les fonctionnalités essentielles et pleinement fonctionnel.

Cycle de vie d'un projet XP

■ Maintenance

- Assurer le bon fonctionnement continu du système.
- Ajouter des fonctionnalités secondaires.
- Chaque ajout de fonctionnalités secondaires mène à de nouvelles releases, nécessitant une phase d'exploration rapide.

■ Mort

- La fin d'un projet intervient quand le client n'arrive plus à écrire de user stories supplémentaires.

Membres d'équipe et rôles

- Développeur : Élément principal du projet XP, chargé d'écrire le code, d'ajouter des fonctionnalités, de simplifier et d'optimiser le code. Le développeur doit aussi savoir bien communiquer et faire preuve de courage.
- Client : L'autre moitié essentielle de l'équipe XP. Le client connaît les besoins du projet, tandis que le développeur sait comment les réaliser.

Membres d'équipe et rôles

- Testeur : Puisque les tests unitaires sont effectués par les développeurs, le testeur aide le client à sélectionner et à rédiger les tests fonctionnels.
- Tracker : La « conscience » de l'équipe. Il aide à estimer le temps nécessaire pour chaque user story et surveille l'avancement du projet par rapport au planning.

Membres d'équipe et rôles

- Coach : Responsable du processus global, il réajuste les procédures et cadre le projet de manière non intrusive.
- Consultant : Apporte les connaissances nécessaires pour que l'équipe puisse résoudre ses problèmes elle-même, sans fournir de solutions toutes faites.

Avantages et inconvénients de l'XP

1- Avantages

- Simplicité de la mise en œuvre
- Adapté aux changements
- Favorise la communication entre développeurs et développeur/client
- Grande satisfaction du client et efficacité plus importante.
- Le code source est la principale production du projet.
- Culture d'entreprise adaptée:
 - pas de mérite basé sur les heures supplémentaires;
 - pas d'attachement aux méthodes linéaires et aux tonnes de documents comme reflet de la qualité.

Avantages et inconvénients de l'XP

2- Inconvénients

- Méthodologie déroutante.
- Trop d'anarchie.
- Pas assez de documentation.
- Concerne les petits et moyens projets seulement.
- Nécessite une forte implication du client.

Avantages et inconvénients de l'XP

- Utiliser XP quand ...

- Ne pas utiliser XP si ...

Avantages et inconvénients de l'XP

■ Utiliser XP quand ...

- Besoins flous
- Périmètre mal défini
- Petite équipe (jusqu'à 12 personnes)
- Site unique
- Pas de sous-traitance

■ Ne pas utiliser XP si ...

- Périmètre bien défini :
- Besoins changeront peu
- Équipe de plus de 12 personnes
- Développement multi-sites
- Développement offshore
- Projets critiques



Questions !

