Decouvrir l’Agilité

qu'est-ce que l'agilité ?

Agile

- Approche réactive et itérative d'organisation de travail
- Focalisée sur la fonctionnalité et satisfaction client
- Construit en adéquation avec les capacités et limites humaines

Les constats

- Les meilleures idées ne viennent pas forcément au début du projet
  - Il est plus facile de construire par étape que tout imaginer dès le début
- Les besoins peuvent évoluer pendent le projet
- Le formalisme n'est pas naturel
- Chiffrages et Reste à Faire sont difficiles à évaluer : **On ne sait pas estimer la charge restante**

Agile : Une catégorie de méthodes

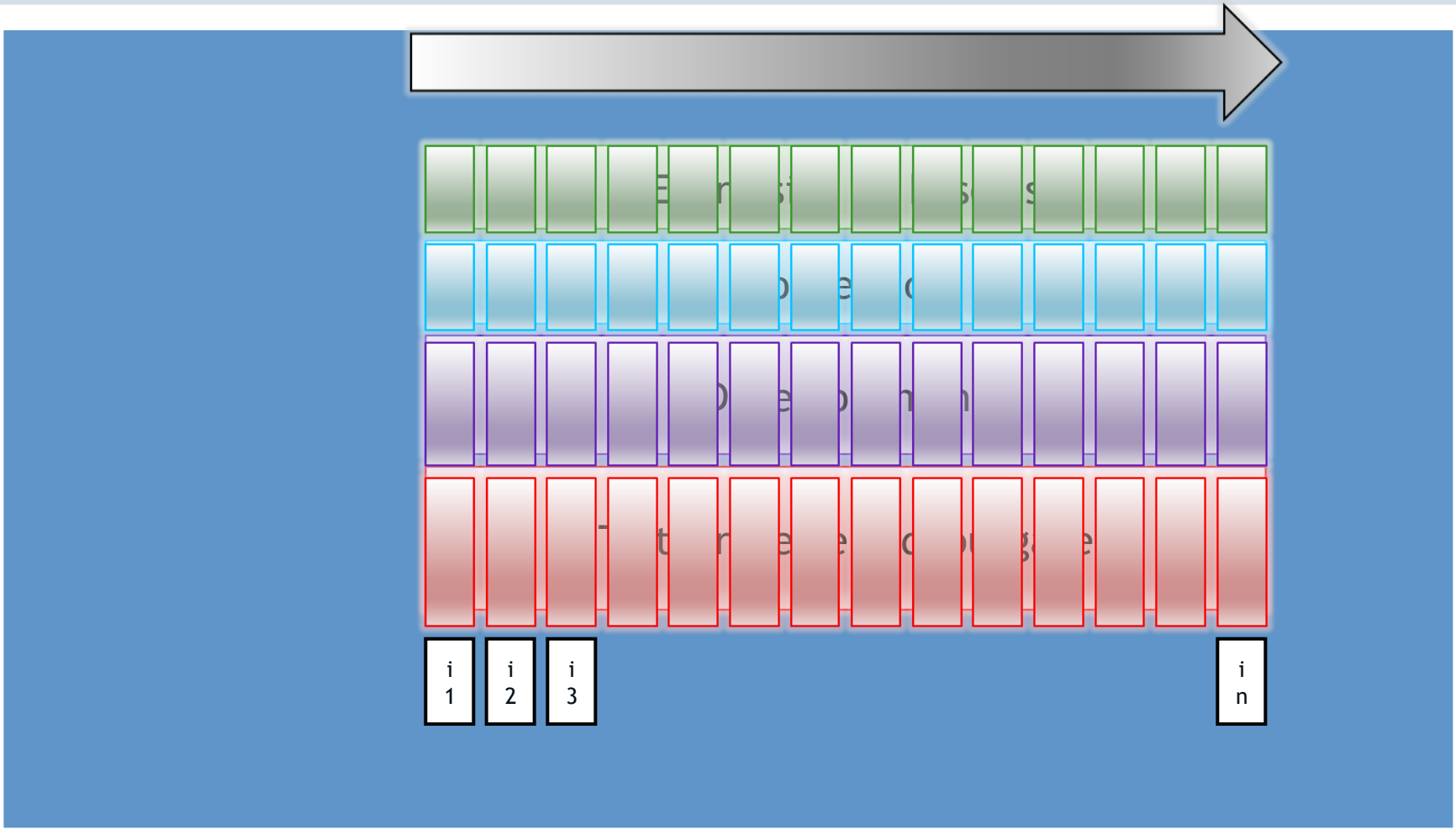
- ‘Agile’ regroupe plusieurs méthodologies :
  - Scrum
  - Extreme Programming (XP)
  - DSDM
  - Crystal
  - …
- Notion officialisée en 2001 avec le Manifeste Agile

Le manifeste Agile

Plutôt que

Personnes et interactions	Processus et outils
Un produit opérationnel	Documentation exhaustive
Collaboration avec le client	Négociation d'un contrat
Adaptation au changement	Suivi d'un plan

Les solutions Agiles



Les solutions Agiles

- Libérer le génie humain

  pour l'auto-organisation dans un contexte qu'il peut maîtriser :

  - La taille de l'équipe est limitée
  - le domaine du problème est limité

  ➢Petites équipes autogérées

  ➢Portée fonctionnelle restreinte à un moment donné

  ➢Garder un rythme de travail soutenable

  ➢Avancement par itération

Les solutions Agiles

- Toujours focalisées sur le produit final
  - Une vision commune pour l'équipe
    - la satisfaction du client
  - Découper le projet autrement
    - par fonctionnalité
  - Organiser en cycles de développement réduits
    - itérations

Les solutions Agiles

- Collaboration avec le client

  Pourquoi on veut des contrats ?

  - *Instaurer la confiance autrement*
  - *Eviter les effets pervers d'un contrat*

Les solutions Agiles

- Adaptables

  Réactives aux nouveaux besoins

  Réceptives aux nouvelles solutions

  - *Prendre les décisions définitives le plus tard possible*
  - *De courtes itérations permettent de changer de direction sans laisser des éléments à moitié fait*

Exemples de méthodes Agile

Scrum

Scrum : Caractéristiques

- Produire le maximum de valeur pour le minimum de coût
- Besoins capturés dans un backlog de produit priorisé par une personne
- Cycles de développement de 2 à 4 semaines (Sprints) ; équipes autogérées
- Mêlée quotidienne

Scrum : Les Acteurs

- Product Owner
  - Porteur de la vision globale du produit
  - Gère le Backlog du Produit
  - Défini des priorités
  - Accepte ou Rejette les livrables

Scrum : Les Acteurs

- Scrum Master
  - Veille au bon fonctionnement de l'équipe
    - Enlève les obstacles
  - Gardien des pratiques de Scrum
  - Serviteur de l'équipe - Facilitateur
  - N'est pas un chef de projet !

Scrum : Les Acteurs

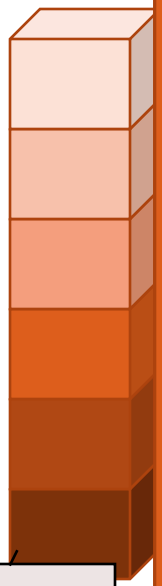
- L'équipe
  - 5 à 9 personnes
  - Autogérée ; les décisions sont prises collectivement
  - Contient toutes les compétences nécessaires pour terminer le sprint
  - Ne change pas pendant un Sprint

Le processus Scrum

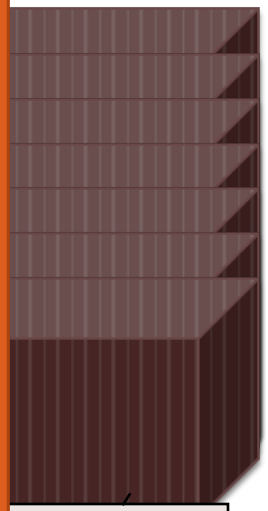
Mêlée quotidienne

- 15 minutes, tous les jours
- Trois questions pour chacun
  - Qu'avez-vous fait hier
  - Qu'allez-vous faire aujourd'hui
  - Quels sont vos problèmes

- Mettre à jour le Backlog du Sprint
- Le reste à faire total pour le Sprint -> burndown chart



Backlog du produit



Produit

Extreme Programming

XP : une autre méthode agile

- L'Extrême Programming (XP) est une méthodologie légère destinée aux projets de petite ou moyenne taille.

- Basé sur des valeurs
  - Coloration culturelle anglo-saxonne

- Basé sur des principes
  - Analogue des Best Practices dans RUP

- Basé sur des pratiques concrètes
  - Discutées ci-dessous

XP : les valeurs (1)

- La communication
  - Développement = Effort collectif de création
  - Avoir une vision commune et pouvoir se synchroniser
    - → Qualité de la communication
  - Communication directe et le contact humain
    - Faiblesse pour la traçabilité et la structuration
    - Augmentation de la réactivité
  - Communication écrite présente, en général par du code

- La simplicité
  - « La chose la plus simple qui puisse marcher »
  - Eviter la complexité inutile dans le code
  - Toute duplication doit être éliminée

XP : les valeurs (2)

- Feedback rapide
  - Boucles de feedback pour réduire les risques
    - Connaître l'état du projet
    - Rectifier le tir si nécessaire
  - Facteur de qualité
    - Acquisition d'expérience
    - Amélioration constante du travail

- Le courage
  - Les développeurs acceptent de construire des prototypes jetables.
  - Se lancer dans un projet non entièrement spécifié
  - Accepter de remanier une portion de code devenue complexe
  - Appliquer les valeurs de feedback et de communication
    - Accepter de montrer ses propres limites
    - Maintenir une transparence complète

XP : les principes (1)

- **Le client (maîtrise d'ouvrage) pilote lui-même le projet**, et ce de très près grâce à des cycles itératifs extrêmement courts (1 ou 2 semaines)
- **L'équipe livre très tôt dans le projet une première version** du logiciel, et les livraisons de nouvelles versions s'enchaînent ensuite à un rythme soutenu pour obtenir un feedback maximal sur l'avancement des développements
- **L'équipe s'organise elle-même pour atteindre ses objectifs,** en favorisant une collaboration maximale entre ses membres
- **L'équipe met en place des tests automatiques pour toutes les fonctionnalités** qu'elle développe, ce qui garantit au produit un niveau de robustesse très élevé
- **Les développeurs améliorent sans cesse la structure interne du logiciel** pour que les évolutions y restent faciles et rapides

XP : Rôles

- Programmeur
- Client
- Testeur
- Tracker
- Manager
- Coach

XP : programmeur

- Retour du programmeur comme rôle central
- Pratiques XP
  - Programmation en binôme
  - Tests unitaires
  - Conception simple
  - Remaniement
  - Responsabilité collective du code
  - Règles de codage
  - Intégration continue
  - Rythme durable

XP : client

- Pas nécessairement le client contractuel
- Assistant à maîtrise d'ouvrage
- Représentant des utilisateurs
- Intégré à l'équipe de développement
- A défaut quelqu'un pour agir comme client « artificiel »
  - Chef de projet
  - Ingénieur chargé des spécifications
- Description informelle d'une fonctionnalité ou d'une interaction avec l'utilisateur
- Le plus simple possible
- **Au démarrage du projet**
  - Des scénarios initiaux sont dégagés et présentés
  - Ils sont classés par priorité et répartis en itérations

XP :client (2)

- Tests de recette
  - But : préciser les contours des scénarios
  - Données concrètes levant les ambiguïtés
  - Preuve que le système fait ce qu'il demandait
  - A chaque fin d'itération tous les tests de recette doivent passer avec succès
- Pratiques XP :
  - Planification itérative
  - Rédaction des scénarios clients
  - Séances de planification
  - Tests de recette
  - Intégration continue
  - Rythme durable

XP : testeur (1)

- Définit et automatise les tests de recette
- Conseille le client sur la testabilité d'une fonctionnalité
- Utilisation d'outils différents pour scripter
- Garant du sentiment de réussite sur le projet
- Test fonctionnel réussi → Progression
- Communiquer pour motiver (graphique de progression…)

XP : testeur (2)

- Pratiques XP :
  - Suivi des tests (planification itérative)
  - Tests de recette
  - Intégration continue
  - Rythme durable

XP : tracker

- Suivre les tâches en cours d'itération
- Interroger les programmeurs
  - Savoir où ils en sont
  - Ne pas les mettre sur des charbons ardents
  - Attention à ne pas dériver en discussion technique
- Détecter les problèmes avant qu'il ne soit trop tard
  - Révélateur
  - Pas de prise d'initiative
- Il fait en sorte que la tâche de 3 jours en prenne 4 et non 6
- De préférence pas un programmeur, mais quelqu'un d'extérieur
  - Pour éviter les discussions techniques
  - A défaut, ce rôle peut tourner entre les programmeurs à chaque itération

Pratiques XP : Planification itérative

XP : manager

- Position dans l'organisation
  - Supérieur hiérarchique des programmeurs
  - Ne fait pas partie intégrante de l'équipe
  - Chef du service auquel appartient l'équipe
  - Chef de projet global (dans le cadre d'un sous-projet)
- Responsabilités
  - Il s'occupe matériellement de l'équipe
  - Il demande des comptes (sur les engagements pris)
  - Interface avec l'extérieur (dans le cadre d'un sous-projet)
- Pratiques XP :
  - Scénarios client
  - Planification itérative
  - Rythme durable

XP : coach

- Garant du processus
  - Il réunit tout les autres rôles
  - Vérifie que chaque rôle est respecté
  - Ses buts ultimes :
    - Equipe autonome
    - Chaque programmeur est en amélioration continue
  - Ses qualités
    - Expert de la méthode XP
    - Expert technique
    - Programmeur chevronné
    - Architecte
    - Pédagogue et sensible
    - Sang-froid
  - Pratiques XP : toutes

XP : combinaison possible des rôles

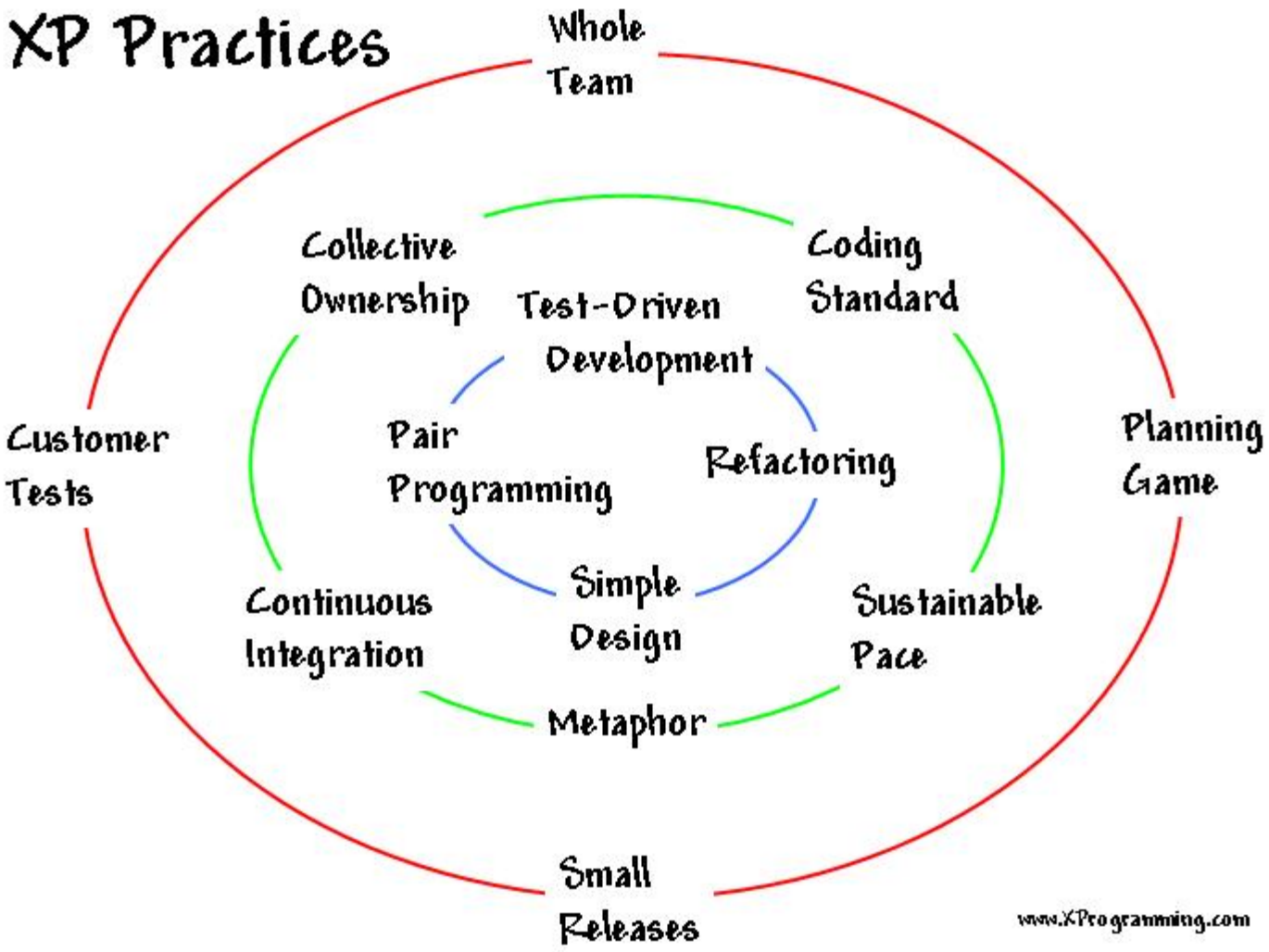
	Programmeur	Client	Programmeur	Tracker	Manager	Coach
Programmeur		✗	!	!	✗	!
Client	✗		✓	✗	✗	✗
Testeur	!	✓		✗	✗	
Tracker	!	✗	✗		!	!
Manager	✗	✗	✗	!		
Coach	!	✗	✗	!	✗	

✓ bonne combinaison

✗ mauvais combinaison

! combinaison envisageable mais risquée

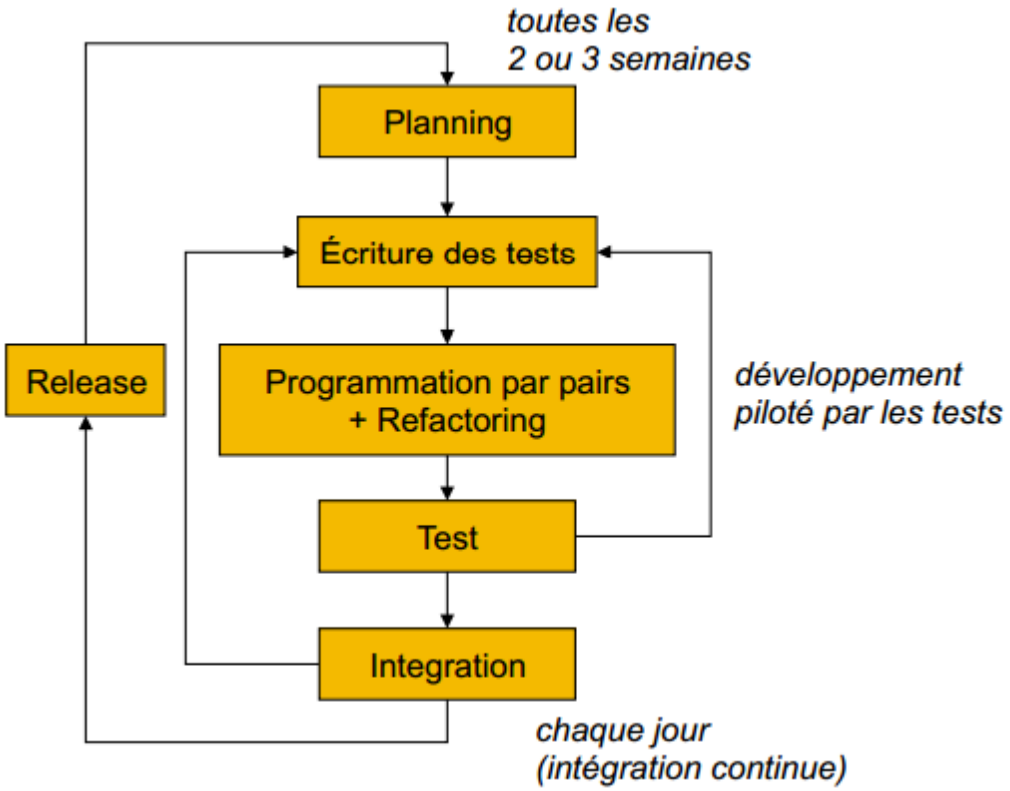
Pratiques XP



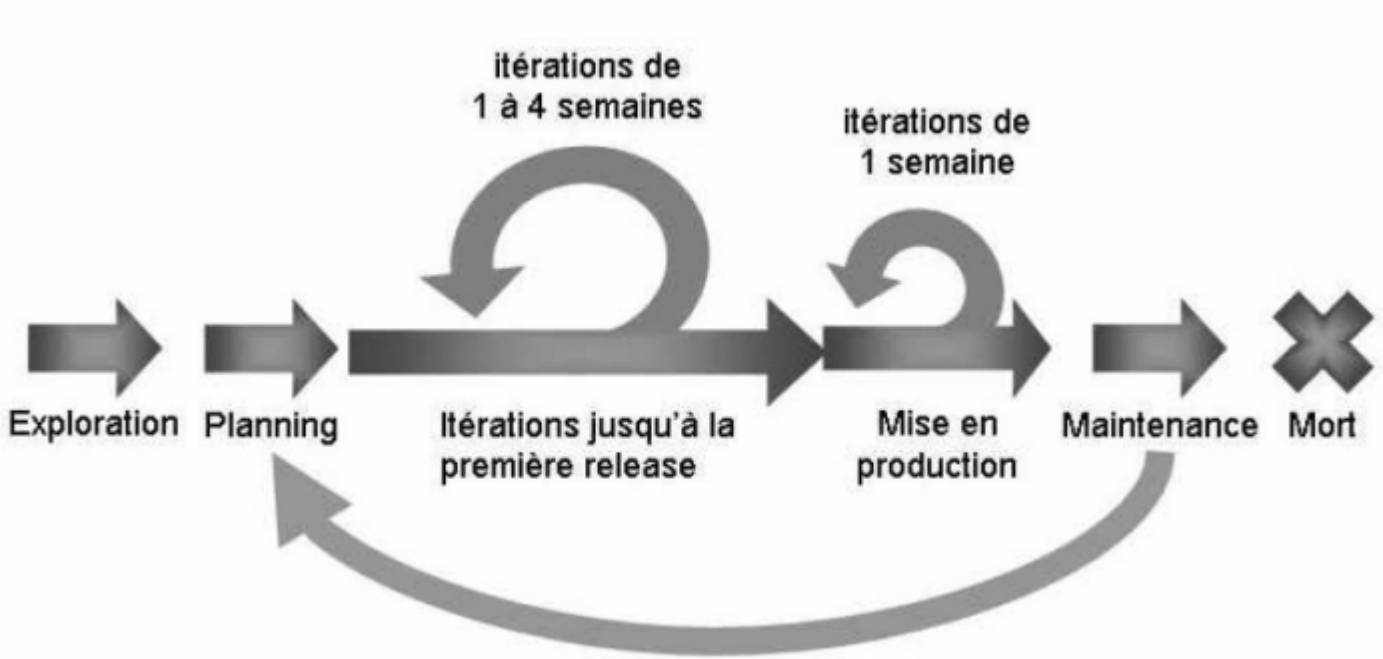
XP : pratiques

Programmation	Développement piloté par les tests Conception simple Remaniement
Collaboration	Programmation en binôme Responsabilité collective du code Règles de codage Intégration continue
Gestion de projet	Client sur site Rythme durable Livraisons fréquentes Planification itérative

XP : vue d'ensemble



XP : planification globale du projet



XP : les pratiques (1)

- Refactoring
  - Simplification de la conception après ajout d'une fonctionnalité.
  - La nouvelle conception doit passer les tests

- Programmation en binôme (Pair Programming)
  - Deux personnes sur une même machine
    - L'un tape le code
    - L'autre suggère les améliorations
  - Conséquences
    - Meilleure qualité de code
    - Changer les rôles
    - Changer les binômes

- Intégration continue
  - Construire la totalité du système deux fois par jour.
  - Intègre le déroulement de tous les tests.
  - S'assurer que les modifications n'entraîne aucune régression

XP : les pratiques (2)

- Tests TDD : Test Driven Developpement
  - Ecrire les tests avant d'écrire le code.
  - Ecriture de tests fonctionnels en partie par l'utilisateur.

- Ecoute
  - Faire en sorte que les échanges puissent avoir lieu partout et concernent les problèmes d'aujourd'hui.
  - Le client doit être disponible en permance.

- Conception
  - Chasser toutes les redondances (once and only once)

XP : les pratiques (3)

- **Scenarios *(Stories)***
  - La planification du travail est basée sur la quantification de fonctionnalités. Les fonctionnalités sont décrites en scénarios d'utilisation pertinents pour le client. L'objectif est de faciliter la communication entre client et développeurs.
- Gestion du projet
  - Les commerciaux participent aux décisions concernant l'étendue du système.
  - Les techniciens participent aux estimations de coûts et de délais.

- Le client est sur le site de développement
  - Le client est présent à plein temps
  - C'est un utilisateur avec une vision globale des besoins
  - Il écrit les « user story », donne les priorités, effectue les tests unitaires
  - Le client est sur place pour répondre à toutes les questions des développeurs

XP : les pratiques (4)

- Petites Livraisons
  - De l'ordre de deux semaines
  - Diminuer les risques d'estimation
  - Diminuer les risques techniques
  - Améliorer le feed-back

- Utilisation des métaphores
  - Améliorer la communication
  - Eviter un formalisme technique mal compris des utilisateurs

XP : les pratiques (5)

- *Coding Standard*
  - Définir des standards de code
    - Permet la pratique suivante …
  - Appropriation collective du code
    - Toute l'équipe est sensée connaître tout le code
    - Le code n'appartient à personne
- *Apply Modeling Standards*
  - *User stories*
  - CRC (*Class Responsability Collaborator*)