

## TD pour renforcer les concepts d'XP

- **Les termes** clés de XP

User story, Sprint, Pair programming, test unitaire, Refactoring, Intégration continue, Client comme partie prenante, Rétrospective

### User Story

Une user story est une description concise d'une fonctionnalité souhaitée par un utilisateur, exprimée du point de vue de cet utilisateur. Elle sert de base pour la planification et le développement des fonctionnalités d'un produit. Une user story typique suit le format suivant :

- En tant que [rôle], Je veux [action], Afin de [résultat].

### Sprint

Un **sprint** est une itération courte et fixe (généralement de 2 à 4 semaines) au cours de laquelle une équipe développe et livre un ensemble de fonctionnalités fonctionnelles. Les sprints permettent de diviser un projet en cycles itératifs, favorisant ainsi la flexibilité et l'adaptation aux changements.

### Pair Programming

Le **pair programming** est une pratique où deux développeurs travaillent ensemble sur le même code, à un seul ordinateur. Un développeur écrit le code tandis que l'autre le révise en temps réel. Cette pratique favorise la qualité du code, le partage des connaissances et l'apprentissage mutuel.

### Test Unitaire

Les tests unitaires sont essentiels pour garantir la qualité du code et faciliter les modifications futures.

### Refactoring

Le **refactoring** est le processus d'amélioration de la structure interne d'un code existant sans en modifier le comportement externe. Il permet d'améliorer la lisibilité, la maintenabilité et l'évolutivité du code.

### Intégration Continue

L'**intégration continue** est une pratique qui consiste à intégrer fréquemment les modifications de code dans un dépôt central et à exécuter automatiquement des tests pour vérifier que le code fonctionne toujours. Cela permet de détecter rapidement les problèmes et de maintenir une base de code stable.

### Client comme Partie Prenante

Dans la méthode XP, le **client** est considéré comme un membre à part entière de l'équipe de développement. Il est impliqué tout au long du projet, fournissant des retours réguliers et aidant à prioriser les fonctionnalités.

### Rétrospective

Une **rétrospective** est une réunion régulière au cours de laquelle l'équipe réfléchit sur le sprint précédent, identifie ce qui a bien fonctionné et ce qui peut être amélioré. L'objectif est d'apprendre des expériences passées et d'améliorer le processus de développement.

**Ces concepts sont interconnectés et contribuent à la mise en œuvre d'une méthode de développement agile et efficace.**

### Exemple de e-commerce

#### Étapes du développement d'un e-commerce avec XP

1. **Définition des user stories:**

- **Exemples:** "En tant que client, je veux pouvoir ajouter un produit à mon panier", "En tant qu'administrateur, je veux pouvoir gérer les catégories de produits".
  - **Priorisation:** Les user stories sont priorisées en fonction de leur valeur métier et de leur complexité.
2. **Planification des sprints:**
    - Les user stories sont regroupées en sprints de durée fixe (généralement 2 à 4 semaines).
    - Les équipes estiment la complexité de chaque user story et planifient les tâches à réaliser pendant le sprint.
  3. **Développement itératif:**
    - Les développeurs travaillent en étroite collaboration avec le client pour implémenter les user stories.
    - Les tests sont effectués régulièrement pour s'assurer de la qualité du code.
    - Le code est refactoré pour améliorer sa lisibilité et sa maintenabilité.
  4. **Rétrospectives:**
    - À la fin de chaque sprint, l'équipe se réunit pour analyser ce qui a bien fonctionné, ce qui n'a pas fonctionné et pour identifier les améliorations à apporter.

#### Exemple concret : Fonctionnalité "Panier"

- **User stories:**
  - "En tant que client, je veux pouvoir ajouter un produit à mon panier."
  - "En tant que client, je veux pouvoir modifier les quantités de produits dans mon panier."
  - "En tant que client, je veux pouvoir supprimer un produit de mon panier."
- **Développement:**
  - Création d'une entité "Panier" dans la base de données.
  - Implémentation des méthodes pour ajouter, supprimer et modifier des produits dans le panier.
  - Écriture de tests unitaires pour vérifier le bon fonctionnement de ces méthodes.
- **Intégration:**
  - Intégration de la fonctionnalité "Panier" dans l'interface utilisateur.
  - Mise à jour de la documentation technique.

#### Exemple concret : Un cours de mathématiques

- **Sprint 1:** Introduction aux nombres entiers, les quatre opérations.
- **Sprint 2:** Les fractions, les décimaux, les pourcentages.
- **Sprint 3:** La géométrie : les formes, les mesures.

#### Les avantages de l'approche XP dans l'éducation

- **Une meilleure adaptation aux besoins des étudiants**
- **Un apprentissage plus actif et collaboratif**
- **Une amélioration continue des méthodes d'enseignement**
- **Une meilleure motivation des étudiants**

## Exercice : Développement d'une application de gestion de bibliothèque (niveau intermédiaire)

**Contexte:** Une équipe de développement doit créer une application web simple pour gérer une bibliothèque. L'application devra permettre aux utilisateurs d'ajouter des livres, de les emprunter, de les rendre et de rechercher des livres par titre ou par auteur.

### Scrum : Le cœur de l'agilité

**Scrum** est la méthode agile la plus populaire et la plus largement utilisée. Elle structure le travail en courtes itérations appelées **sprints**, généralement d'une durée de deux à quatre semaines. Chaque sprint est un cycle complet de développement, de la conception à la livraison d'un incrément du produit.

### Les rôles clés dans Scrum

- **Le Product Owner:** Il définit la vision du produit, priorise les fonctionnalités et agit comme l'interface entre l'équipe de développement et les clients.
- **Le Scrum Master:** Il facilite le processus Scrum, assure que l'équipe suit les règles et élimine les obstacles. Il est le garant de l'application de la méthode.
- **L'équipe de développement:** Elle est auto-organisée et responsable de la réalisation du produit.

### Les cérémonies Scrum

- **La planification du sprint:** Au début de chaque sprint, l'équipe sélectionne les éléments du backlog (Un backlog est une liste de tâches ou d'éléments de travail qui doivent être accomplis) produit à réaliser.
- **La daily scrum:** Une courte réunion quotidienne où chaque membre de l'équipe partage ce qu'il a accompli, ce qu'il prévoit de faire et les obstacles rencontrés.
- **La revue de sprint:** À la fin de chaque sprint, l'équipe présente le travail accompli aux parties prenantes.
- **La rétrospective:** L'équipe se réunit pour identifier ce qui s'est bien passé, ce qui pourrait être amélioré et met en place des actions correctives.

### Les artefacts Scrum

- **Le Product Backlog:** Une liste ordonnée de toutes les fonctionnalités souhaitées pour le produit.
- **Le Sprint Backlog:** Un sous-ensemble du Product Backlog qui contient les éléments à réaliser pendant un sprint spécifique.
- **L'Incrément:** Le résultat concret d'un sprint, un produit potentiellement livrable.

### Les valeurs de Scrum

- **L'engagement:** Les membres de l'équipe s'engagent à atteindre l'objectif du sprint.
- **La focalisation:** L'équipe se concentre sur le travail à accomplir pendant le sprint.
- **L'ouverture:** L'équipe travaille de manière transparente et ouverte.
- **Le respect:** Les membres de l'équipe se respectent mutuellement.
- **Le courage:** L'équipe a le courage de prendre des décisions difficiles.

## Les avantages de Scrum

- **Flexibilité:** Adaptation rapide aux changements.
- **Transparence:** Visibilité sur l'avancement du projet.
- **Collaboration:** Travail d'équipe renforcé.
- **Qualité:** Livraison régulière d'incrément de qualité.
- **Satisfaction client:** Implication du client tout au long du projet.

## Projet : Développement d'une application de livraison de repas

**Contexte:** Une startup souhaite développer une application mobile permettant aux utilisateurs de commander des repas auprès de différents restaurants et de se les faire livrer.

### Mise en place de Scrum:

1. **Constitution de l'équipe:**
  - **Product Owner:** Le fondateur de la startup, qui a une vision claire du produit et des besoins des utilisateurs.
  - **Scrum Master:** Un consultant agile expérimenté, chargé de faciliter le processus Scrum et de résoudre les problèmes.
  - **Équipe de développement:** Des développeurs front-end, back-end, des designers et un testeur.
2. **Création du Product Backlog:**
  - **User stories:**
    - "En tant qu'utilisateur, je veux pouvoir rechercher des restaurants par type de cuisine."
    - "En tant que restaurant, je veux pouvoir gérer mes menus et mes commandes."
    - "En tant que livreur, je veux pouvoir accéder à ma liste de livraisons et mettre à jour leur statut."
  - **Priorisation:** Les user stories sont priorisées en fonction de leur valeur métier et de leur complexité.
3. **Sprints de deux semaines:**
  - **Planification du sprint:** L'équipe sélectionne les user stories à réaliser pendant le sprint en fonction de la capacité de l'équipe et de la priorité.
  - **Daily Scrum:** Une réunion quotidienne de 15 minutes où chaque membre de l'équipe partage ses avancées, ses blocages et ses plans pour la journée.
  - **Développement:** L'équipe travaille sur les tâches définies pendant la planification.
  - **Revue de sprint:** À la fin du sprint, l'équipe démontre les fonctionnalités développées au Product Owner et aux autres parties prenantes.
  - **Rétrospective:** L'équipe identifie les points forts, les points faibles et les améliorations à apporter au processus.

### Exemple d'un sprint:

- **Sprint 1:** Développement de la fonctionnalité de recherche de restaurants par type de cuisine et création d'un prototype de l'interface utilisateur.
- **Sprint 2:** Intégration du paiement en ligne et mise en place du système de notifications pour les utilisateurs et les livreurs.
- **Sprint 3:** Test de l'application en conditions réelles avec un groupe d'utilisateurs bêta et correction des bugs.

## Avantages de l'utilisation de Scrum dans ce projet:

- **Flexibilité:** La startup peut rapidement s'adapter aux retours des utilisateurs et aux évolutions du marché.
- **Transparence:** Les progrès du projet sont visibles pour tous les acteurs.
- **Collaboration:** L'équipe travaille de manière collaborative et les membres sont responsabilisés.
- **Qualité:** Les fonctionnalités sont livrées régulièrement et testées en continu.

**Exemple :** Un site e-commerce de vente de vêtements constate un taux d'abandon de panier élevé, notamment lors du processus de paiement.

### 1. Définition du produit et du backlog

- **Product Owner:** Le responsable e-commerce définit la vision : "Améliorer l'expérience utilisateur lors du processus de paiement pour réduire le taux d'abandon de panier de 20% en 3 mois."
- **Backlog produit:**
  - Simplifier le formulaire de paiement
  - Proposer plus d'options de paiement
  - Afficher des garanties de sécurité
  - Mettre en place un système de rappel de panier abandonné

### 2. Sprints itératifs (exemple : Sprint 1)

- **Planification:** L'équipe décide de se concentrer sur la simplification du formulaire de paiement.
- **Développement:**
  - Réduire le nombre de champs obligatoires
  - Améliorer la lisibilité du formulaire
  - Tester différentes mises en page
- **Revue:** L'équipe présente la nouvelle version du formulaire à l'équipe produit et aux parties prenantes.
- **Rétrospective:** L'équipe identifie les difficultés rencontrées et les améliorations possibles pour le prochain sprint.

### 3. Sprints suivants

- **Sprint 2:** Proposer plus d'options de paiement (PayPal, Apple Pay, etc.)
- **Sprint 3:** Afficher des garanties de sécurité (certificat SSL, avis de clients, etc.)
- **Sprint 4:** Mettre en place un système de rappel de panier abandonné par email.

### 4. Mesure et amélioration continue

- **KPI:** Suivre le taux d'abandon de panier, le temps moyen passé sur la page de paiement, le taux de conversion.
- **Analyses:** Utiliser des outils d'analyse web pour identifier les points de friction.
- **Itérations:** Continuer à améliorer le processus de paiement en fonction des résultats.

## Résultats attendus

- **Réduction du taux d'abandon de panier:** En simplifiant le processus de paiement et en rassurant les clients, le taux d'abandon devrait diminuer.
- **Amélioration de l'expérience utilisateur:** Les clients apprécieront un processus de paiement plus fluide et sécurisé.
- **Augmentation du chiffre d'affaires:** Une meilleure expérience utilisateur devrait se traduire par une augmentation des ventes.

## Points clés

- **Flexibilité:** Scrum permet de s'adapter rapidement aux changements et aux retours des utilisateurs.
- **Collaboration:** L'équipe travaille ensemble pour trouver les meilleures solutions.
- **Mesure:** Les résultats sont suivis et analysés régulièrement pour améliorer le processus.
- **Amélioration continue:** Le processus est constamment optimisé.

**Cet exemple montre comment Scrum peut être utilisé pour résoudre un problème concret dans le domaine du e-commerce. En appliquant cette méthodologie de manière itérative et en se concentrant sur la satisfaction client, il est possible d'obtenir des résultats significatifs.**

## Le choix entre XP et scrum

Le choix entre Scrum et XP dépendra en grande partie de votre contexte spécifique, de votre équipe et des objectifs de votre projet. Les deux méthodologies agiles ont leurs propres forces et sont souvent complémentaires.

**Scrum et XP (eXtreme Programming)** sont deux des méthodologies agiles les plus populaires. Elles partagent de nombreux principes communs, comme l'itération, l'incrémentation et la collaboration, mais elles ont également des différences notables.

## Quand choisir Scrum ?

- **Flexibilité et adaptabilité:** Scrum est particulièrement adapté aux projets complexes et changeants, où les exigences peuvent évoluer rapidement.
- **Équipes auto-organisées:** Si votre équipe est mature et autonome, Scrum peut favoriser la prise de décision et la résolution de problèmes au sein de l'équipe.
- **Focus sur la valeur métier:** Scrum met l'accent sur la livraison régulière de valeur au client, ce qui est idéal pour les projets où les besoins du client sont prioritaires.

## Quand choisir XP ?

- **Qualité du code et tests:** XP est réputé pour ses pratiques rigoureuses de développement, telles que les tests unitaires, l'intégration continue et la refactoring. Si la qualité du code est une priorité absolue pour votre projet, XP peut être un excellent choix.
- **Petites équipes colocalisées:** XP fonctionne mieux avec des équipes de petite taille qui travaillent dans le même espace physique.
- **Projets complexes du point de vue technique:** XP peut être utile pour gérer des projets avec des exigences techniques élevées.

