

Le site Permalivre va bientôt sortir !

Faire en sorte que chacun puisse produire sa propre alimentation, de façon écologique, voilà l'ambition folle du site de vente en ligne Permalivre qui, selon son directeur Marketing, « *permettra à tous d'acheter pour un prix dérisoire toute la littérature autour de la permaculture et de l'agriculture accessible à tous, locale, solidaire et respectueuse de l'environnement.* »

Promis d'ici moins de 3 mois, ce site sera développé par une petite équipe informatique suivant les principes de l'agilité et une méthode agile, dénommée Scrum. Vous ne connaissez pas l'agilité ? Pas de panique, on vous explique tout, et on vous présente l'équipe !

C'est quoi, l'Agilité ?

L'agilité est un ensemble de pratiques de développement qui promeut, entre autres :

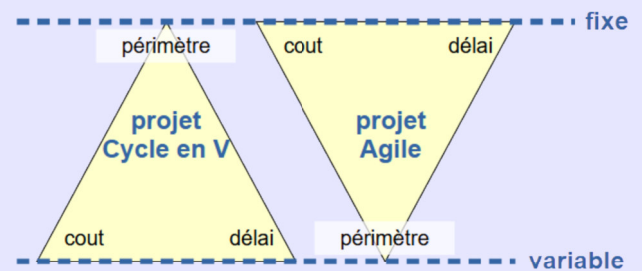
- La collaboration avec le client**
- Une construction itérative et incrémentale**
- L'excellence technique**
- L'adaptation au changement**

Retrouvez les 4 valeurs et les 12 principes de l'Agilité sur la page des fondateurs : agilemanifesto.org

Et Scrum ?

Scrum est une des méthodes agiles, qui définit certains rôles, rythme le développement en **sprints** de durée fixe, et introduit des **cérémonies**.

Le triangle de fer de l'agilité



Dans un projet classique « cycle en V », le périmètre est fixé, et on augmente les couts et les délais jusqu'à ce que tout soit développé. En agilité, c'est l'inverse ! On ajuste le périmètre pour tenir les couts et les délais, quitte à déprioriser des fonctionnalités.

Présentation de l'équipe

Le Scrum Master

Le Scrum Master est un facilitateur qui accompagne l'équipe, assure le respect des principes et pratiques Scrum, la bonne tenue des outils, des cérémonies, des tableaux de suivi.

Le Product Owner

C'est le représentant du client et des utilisateurs. Il priorise le Product Backlog pour maximiser la valeur délivrée.

Les développeurs

L'ouverture du site Permalivre a été...

☐ Un franc succès ! (€ > 7)

L'équipe de développement a su garder la tête froide malgré les fortes contraintes de cout et de délai imposées par leur client.

Non seulement ils ont su développer un **MVP (Minimum Viable Product)** assurant une expérience utilisateur minimale, mais ont offert des fonctionnalités supplémentaires qui répondent réellement aux besoins des clients.

☐ Un succès mitigé ! (€ ≤ 7)

La promesse de livrer un site en 3 mois a été tenue : le site est fonctionnel, le contenu est bien présent, même si les internautes raillent le manque de fonctionnalités par rapport aux géants du web marchand.

Le Directeur Marketing vient d'annoncer que l'équipe de développement a été reconduite pour la future montée en puissance du site.

☐ Une catastrophe ! (échec critique)

Le site, clairement pas fini, voire abandonné en cours de route, est qualifié d'« Epic Fail » par les internautes, qui raillent l'absence d'une expérience d'achat complète (le MVP n'a pas été réalisé) et le nombre invraisemblable de bugs.

L'équipe a été licenciée et le Directeur Marketing de Permalivre annonce se reconvertir dans la vente d'aspirateurs.

Concepts-clés

Product backlog : liste organisée et priorisée de tout ce qui pourrait être fait pour construire ou améliorer le produit.

Sprint backlog : ensemble des éléments qui vont être réalisés pendant un sprint.

User Story : description courte et simple d'une fonctionnalité, prise du point de vue de son utilisateur.

Épic : grande unité de travail qui doit être décomposée en User Stories plus petites.

Sprint : durée fixe des itérations Scrum. La durée des sprints est fixée par l'équipe au démarrage du projet, et est en général entre deux et quatre semaines.

Done : le DoD (Definition of Done) est un accord crucial au sein d'une équipe Scrum qui établit l'ensemble des critères à atteindre pour qu'une unité de travail soit considéré comme terminé.

MVP (Minimum Viable Product) : version minimale d'un produit, mais suffisante pour apporter une valeur métier et recueillir le feedback des utilisateurs.

Les cérémonies

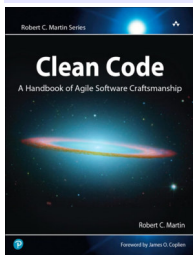
Sprint planning : En début de sprint, le sprint planning définit les objectifs à réaliser pour le sprint à venir. Cela inclut une sélection de fonctionnalités (ou user stories), leur estimation (par le planning poker par exemple) et des choix de méthode et d'organisation.

Daily scrum : réunion quotidienne rapide où chacun décrit ses réalisations de la veille, ses objectifs de la journée et les problèmes qu'il rencontre.

Démonstration : La démonstration permet de partager avec les parties prenantes un état d'avancement du projet. On n'y présente en général que ce qui est terminé et prêt. C'est l'occasion de recueillir le maximum de feedback de la part des parties prenantes.

Rétrospective : En fin de sprint, l'équipe se réunit pour faire le point sur la période écoulée et proposer des améliorations pour les sprints suivants. C'est le maillon indispensable pour l'amélioration continue et l'auto-organisation de l'équipe.

Un code de qualité



Le code de qualité s'apprend par les bonnes pratiques de conception (**design patterns**), du **pair programming** (programmation à deux), de la **revue de code** par les pairs formalisée (par exemple, pas de code en production tant qu'il n'a pas été relu par quelqu'un d'autre de l'équipe). Pour apprendre, il existe des design patterns et des pratiques de **Clean Code** pour tous les langages et les frameworks.

Vers toujours plus d'automatisation

En agilité, le code évolue vite, est souvent refactorisé, est livré fréquemment, et donc, une bonne dose d'automatisation est nécessaire pour garantir la stabilité de la qualité du code.



Intégration continue
Tests unitaires automatisés
Tests fonctionnels automatisés
Déploiement continu

Les priorités

Maîtriser la dette technique

La dette technique est tout ce qui contribue à rendre le code plus difficile à maintenir et à faire évoluer : absence de commentaires, code mal écrit, duplication du code, mauvaises modélisations, mauvais choix d'architecture, de bibliothèques, ...

La dette technique peut être maîtrisée par un suivi strict de la qualité du code, des outils de mesure de dette, la capacité à refactoriser.

Satisfaire le client et les utilisateurs

Le client s'exprime via le **Product Owner (PO)**. Si le PO est indisponible, un **Proxy Product Owner (PPO)** est désigné.

Le PO et l'équipe ne doivent pas oublier la satisfaction des utilisateurs finaux de l'application. Cette satisfaction peut se mesurer par des enquêtes régulières ou des feedbacks intégrés. Idéalement, un panel d'utilisateurs interagit directement avec les développeurs sur le plateau projet.