[](https://d3timt52sxdbq0.cloudfront.net/wp-content/uploads/2017/06/methodesagiles.jpg)

Les méthodes Agiles – La liste complète des méthodologies Agiles

Une fois qu’une organisation décide d’adopter une gestion de développement Agile, il reste encore à choisir la méthodologie la plus adaptée à son projet. En effet, les méthodes Agiles disponibles sont nombreuses et peuvent être source de confusion. Les méthodes Agiles les plus populaires en usage aujourd’hui sont:

* L’eXtrême Programming (XP),
* Scrum,
* Feature Driven Development (FDD),
* Lean Software Development,
* Agile Unified Process (Agile UP ou AUP),
* Crystal
* et Dynamic Systems Development Method (DSDM) .

Cet article analysera comment ces méthodologies manifestent les valeurs agile et une attention particulière sera apportée pour mettre en évidence les similitudes et les différences entre chaque méthodologie.

Les Méthodes Agiles

Extreme Programming (XP)

Extreme Programming, ou XP, est une méthode agile de gestion de projet particulièrement bien adaptée aux projets de développement informatique. Elle a été conçue par Kent Beck pour accélérer les développements alors qu’il travaillait pour la société Chrysler. L’idée lui est venue alors qu’il devait intervenir sur un logiciel de paie écrit en langage Smalltalk ayant accumulé une dette technique considérable, le rendant particulièrement complexe à maintenir et à faire évoluer.

Le principe fondamental de la [méthode XP](https://www.nutcache.com/fr/blog/la-methode-agile-xp/) est de faire collaborer étroitement tous les acteurs du projet et d’opter pour des itérations de développement très courtes. La planification des tâches reste très souple et l’estimation des charges simplifiée par des projections à très court terme. Ainsi la correspondance entre ce qu’attend le client et les réalisations est garantie. Les fonctionnalités sont livrées régulièrement, afin d’être testées et validée au travers de prototypes opérationnels.

L’Extreme Programming préconise également le travail en binôme des développeurs, facilitant ainsi la production d’un code simple, facilement lisible et maintenable.

Scrum

La [méthode agile Scrum](https://www.nutcache.com/fr/blog/methodologie-scrum/) particulièrement destinée à la gestion de projets informatiques tient son nom du monde du rugby. Le principe de Scrum est de pouvoir modifier la direction prise par le projet au fur et à mesure de son avancement. C’est exactement ce qui se passe lors d’un match de rugby, lors d’une mêlée (« scrum » en anglais).

La mêlée est donc une phase essentielle au rugby comme dans la gestion de projet. Si les conditions de réussite ne sont pas remplies, alors il faut réorienter le projet pour repartir sur de meilleures bases. Le client est étroitement impliqué grâce à la livraison régulière de prototypes opérationnels permettant de valider les développements. Cette gestion dynamique permet de s’assurer de la correspondance entre le besoin exprimé et le produit livré, et de réorienter au besoin les futurs développements.

Feature Driven Development (FDD)

Le Développement Dirigé par les Fonctionnalités, en français, est une méthode de gestion de projet basée sur la gestion des risques. Les développements sont organisés en itérations courtes autour de fonctionnalités testables par l’utilisateur. L’utilisateur est ainsi impliqué dans les développements, peut suivre l’avancement du projet et la validation des fonctionnalités. Aucune méthode de programmation n’est préconisée, c’est réellement la fonctionnalité qui est mise en avant.

Un projet géré par FDD est découpé en plusieurs grandes étapes. La première consiste en la constitution d’un modèle général du produit, qui va définir le périmètre global de réalisation. Vient ensuite la construction de la liste complète des fonctionnalités à réaliser. Il est impératif lors de cette étape d’impliquer au maximum le client. Ces fonctionnalités sont finalement regroupées en fonction de leurs caractéristiques communes et priorisées. Les deux dernières étapes verront le jour sous la forme d’itérations, et englobent d’une part la conception technique des fonctionnalités, puis leur réalisation.

Cette méthode donne une importance plus grande à la phase de conception, quitte à démarrer la réalisation plus lentement, afin d’avoir un modèle plus solide.

Lean Software Development

Le Lean Software Development est basé sur sept grands principes.

* Éliminer les gaspillages (finition partielle, processus inutiles, fonctionnalités non nécessaires, modification de l’équipe, retards…),
* Favoriser l’apprentissage (multiplication des sources d’apprentissage, synchronisation des équipes…),
* Reporter les décisions (jusqu’au dernier moment raisonnable, pour éviter de longues discussions sources de pertes de temps et les décisions irrévocables),
* Livrer vite (livraisons rapides et régulières de façon à avoir un retour client rapide également),
* Responsabiliser l’équipe (favoriser l’autonomie et le leadership des équipes, partir du principe que les intervenants connaissent leur travail, faciliter le développement de l’expertise),
* Construire la qualité (elle doit être placée au cœur du projet, de la conception à la réalisation),
* Optimiser le système dans son ensemble (mise en place de mesures de performances complètes, pour avoir en permanence une vision globale du produit, et gérer les différentes interactions et dépendances).

Avec la [méthode Lean](https://www.nutcache.com/fr/blog/la-gestion-de-projet-lean/), la qualité est réellement placée au cœur de la gestion du projet, en optimisant notamment l’ensemble des processus d’apprentissage, de décision, de livraison et de mesure de performances.

Agile Unified Process (Agile UP or AUP)

Agile Unified Process (ou Processus Unifié Agile) est une version simplifiée du [Rational Unified Process](https://www.nutcache.com/fr/blog/methode-rup/), ou RUP. Il s’agit d’une méthode de développement d’applications métier utilisant les techniques agiles du TDD (Test Driven Development ou développement piloté par les tests), du MDD (Model Driven Development ou développement piloté par le modèle) et de la gestion du changement.

La méthode est divisée en quatre phases :

* Lancement : identification du périmètre du projet, définition de la ou des architectures potentielles pour le système, implication des intervenants et obtention du budget.
* Conception : définir l’architecture du système et démonstration de sa pertinence.
* Réalisation : développement du logiciel lors d’un processus incrémental dans l’ordre de priorité des fonctionnalités.
* Livraison : validation et déploiement du système en production.

Crystal (Clear/Orange)

La [méthode Crystal Clear](https://www.nutcache.com/fr/blog/methode-agile-crystal/) est particulièrement adaptée aux petites équipes de développement. Idéalement, l’équipe est composée d’un architecte et de deux à six ou sept développeurs, situés à proximité les uns des autres, de façon à faciliter la communication, dans un local calme. Des tableaux blancs servent de supports afin que tous aient un accès rapide à toutes les informations. Les rythmes de développement et de livraison sont rapides (toutes les deux semaines ou une fois par mois) afin que les utilisateurs puissent passer les tests.

Durant tout le processus de développement, l’équipe se remet en question en permanence afin d’améliorer continuellement sa façon de travailler.

Dynamic Systems Development Method (DSDM)

Cette méthode s’articule autour de neufs grands principes qui sont la participation des utilisateurs, l’autonomie de l’équipe projet, la transparence des développements, l’adéquation avec le besoin, le développement itératif et incrémental, la réversibilité permanente, la synthèse du projet, les tests automatisés et continus et enfin la coopération entre tous les intervenants.

Le projet commence par une [étude de faisabilité](https://www.nutcache.com/fr/blog/etude-de-faisabilite-projet/) afin de décider s’il faut le faire ou non. Un rapport est rédigé, et éventuellement, un prototype est créé pour démontrer la faisabilité de l’application. S’il a été décidé de continuer, une analyse fonctionnelle est réalisée et les spécifications sont rédigées. A partir de ce moment-là, des itérations de conception technique et de développement sont mises en place, puis au final, l’application est livrée en production.

Adaptive software development (ASD)

ASD est une méthode de développement rapide d’applications. Le principe consiste à automatiser et à industrialiser un maximum de processus. Des outils de modélisation sont utilisés et une usine logicielle est mise en place de façon à générer un maximum de code informatique automatiquement. Un atelier de génie logiciel (AGL) permet ensuite aux développeurs de modifier l’application ainsi générée. Pour finir, une usine de livraison assure l’automatisation de l’ensemble des processus de déploiement.

La méthode ASD est indépendante de toute méthodologie ou langage de programmation.

Behavior driven development (BDD)

Dans cette méthode, c’est le langage naturel qui est mis en avant. Plutôt que de décrire les solutions techniques à mettre en œuvre, ce sont les objectifs des fonctionnalités qui sont décrites par les utilisateurs. Ces derniers sont donc particulièrement impliqués dans le processus de fabrication. Le comportement cible de l’application est donc décrit au travers d’exemples permettant aux développeurs de mieux cerner les objectifs. Les expressions « Etant donné », « quand », « alors » et « et » sont employées dans les scénarios décrits, qui sont également utilisés pour créer des séries de tests de non régression. Les scénarios sont écrits collectivement avec le client, les développeurs et toutes les équipes impliquées dans le processus.

Conception pilotée par le domaine (DDD domain-driven design)

Le Domain Driven Design est une technique de conception d’applications informatiques. La conception est centrée sur le domaine métier et non sur les aspects techniques. Elle permet aux équipes techniques et fonctionnelles de communiquer ensemble afin d’obtenir un modèle commun de l’application, compréhensible par tous. Les développeurs acquièrent ainsi une meilleure connaissance du fonctionnel et de son vocabulaire spécifique, et l’équipe métier a une meilleure vision des contraintes techniques.

Grâce à cet outil, on obtient une meilleure communication entre les différents intervenants et une conception modulaire facilitant à terme la maintenabilité de l’application et la réutilisabilité de ses composants.

Test driven development (TDD)

Le développement piloté par les tests est une technique de développement qui associe l’écriture des tests unitaires, la programmation et le remaniement du code. Les [tests unitaires](https://www.nutcache.com/fr/blog/tests-unitaires/) sont écrits avant le code. Chaque test décrit un élément de la fonctionnalité. Le développeur écrit le minimum de code possible pour que le test passe, et qu’il échoue pour des raisons prévisibles. Au fur et à mesure de l’avancement, le code source doit être simplifié autant que nécessaire et continuer à passer les tests. Les tests s’accumulent durant le développement et sont exécutés automatiquement de façon très régulière.

Le TDD est toujours associé à des outils d’automatisation de tests unitaires liés au langage de programmation utilisé.

Rational Unified Process (RUP)

Cette méthode est un mélange des pratiques classiques et agiles de gestion de projet. Les développements sont itératifs et incrémentaux. Chaque itération respecte un cycle comprenant quatre phases qui sont lancement, conception, réalisation et livraison. Les développements sont guidés par des cas d’utilisation. On commence par les fonctionnalités génériques pour aller ensuite de plus en plus vers le spécifique.

Disciplined Agile Delivery (DAD)

DAD est une méthode d’aide à la décision et d’industrialisation qui a pour objectif de simplifier l’intégration des processus liés à une gestion de projet incrémentale et itérative, agile. Elle s’applique de façon particulièrement efficace en association avec une méthode de développement telle que Scrum ou Lean. Il s’agit d’une méthode hybride, destinée à faciliter l’adoption de l’agilité au sein d’une organisation de taille significative. Tous les aspects de l’intégration de l’agilité dans le développement logiciel sont pris en charge en tenant compte du contexte global du projet et surtout de l’entreprise. Lorsqu’une méthode agile doit être appliquée à un projet impliquant des équipes importantes, DAD peut être d’un grand secours pour faciliter l’adoption des différents processus.

Enterprise Unified Process (EUP)

EUP est une extension des méthodes d’industrialisation comme DAD ou comme les dérivés de Unified Process (UP), ou des méthodes de développement agile comme Scrum. EUP apporte deux autres phases à la fin des cycles des autres méthodes agiles :

* Mise en production : maintenance en condition opérationnelle des systèmes déployés.
* Retrait de production : processus de retrait de la production des systèmes déployés.

La phase de retrait de production peut être mise en œuvre pour plusieurs raisons comme le remplacement complet du système, la fin du support de la version courante, la redondance du système ou encore l’obsolescence de l’application.

**Quel que soit votre choix parmi toutes les méthodes agiles, le**[**logiciel de gestion de projet Nutcache**](https://www.nutcache.com/fr/)**sera l’outil idéal pour suivre l’avancement de votre projet.**