

TO BE AGILE, OR NOT TO BE ?

Auteur : Kevin NGUYEN
Septembre 2015

TheCodingMachine™
TCM://

INTRODUCTION

Les méthodes de développement des logiciels ou des applications web prennent de plus en plus d'importance. La digitalisation est un des enjeux majeurs actuels. Il faut donc préparer le SI à accueillir de nouvelles applications, de nouveaux *business models*. Il faut aussi changer de méthode pour gérer les projets de développement pour aller plus vite, plus loin et avec moins de risques.

Dans ce contexte, les méthodes agiles apparaissent comme la panacée. Naturellement, elles sont agiles, donc elles permettent le changement ! Et puis, de toutes les manières, qui voudrait de la méthode « Maladroite » ou « Gauche » ;-) ?

Bref, en ce moment, il faut être agile ou carrément se résigner à ne pas être...

Chez TheCodingMachine, on pense que chaque projet mérite un instant de réflexion pour adopter la bonne approche méthodologique ! Pour certains types de projets ou bien certains contextes clients, les méthodes agiles sont très bien adaptées. Dans d'autres situations, c'est naturellement moins le cas.

Ce livre blanc s'adresse donc à tous ceux qui aiment réfléchir à leurs pratiques, tous ceux qui aiment développer des projets avec succès !

Après avoir dressé un panorama des méthodologies actuelles, TheCodingMachine propose d'analyser les avantages et les inconvénients de chacune d'entre elles et esquisse une manière de choisir la méthodologie la mieux adaptée à votre projet.

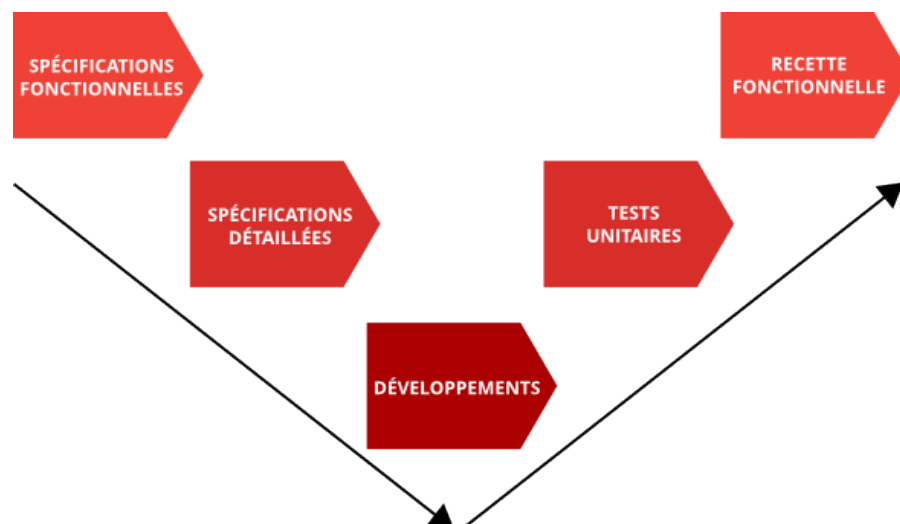
PARTIE 1 - PANORAMA DES METHODES DE GESTION DE PROJET

Cette première partie présente succinctement les deux principales méthodes de gestion de projet : cycle en V et méthodes agiles.

Note : les principes et les valeurs du *Manifeste Agile* sont annexés à ce document.

1.1 - METHODE CLASSIQUE – CYCLE EN V

C'est la méthode la plus classique et c'est certainement celle qui est la plus employée. Le terme « cycle en V » vient du fait que la branche gauche du « V » corresponde, de manière descendante, aux étapes de conception du projet tandis que la branche droite correspond, de manière ascendante, aux étapes de test du projet. Enfin, la pointe du V représente le développement. Chaque étape de conception est liée à une étape de test. Ainsi, comme sur le schéma ci-dessous, les spécifications correspondent avec les tests de validation, la conception préliminaire avec les tests d'intégration, la conception détaillée avec les tests unitaires...



1.2 - METHODES AGILES

PRINCIPE DES METHODES AGILES

Les maîtres-mots des méthodes agiles sont la précocité et la flexibilité. Ils représentent tout leur intérêt. Le client est impliqué en permanence dans le projet afin que ses attentes puissent évoluer et qu'elles soient prises en compte au fur et à mesure des développements dans une approche itérative.

Le développement itératif signifie qu'on découpe le projet en différentes étapes de quelques semaines chacune (selon la complexité de l'étape). Pendant chaque étape, une version fonctionnelle du produit est développée puis soumise au client afin qu'il la valide. Ce dernier bénéficie alors d'une transparence maximale sur l'avancée du projet et peut ajouter des fonctionnalités dès qu'il le souhaite. La possibilité de visualiser une ébauche du projet permet, en effet, de mieux se rendre compte des fonctionnalités à ajouter ou supprimer. Elles sont alors modifiées au fur et à mesure de la conception du projet de manière incrémentale, ce qui signifie que le produit se perfectionne d'itération en itération jusqu'à atteindre la qualité voulue par le client.

Il n'y a pas de planning qui dresse en détail les différentes étapes du projet de manière chronologique. Seules les principales échéances sont fixées et les critères importants du projet sont décrits. Après chaque étape (ou itération) le client ainsi que le prestataire décident ensemble des fonctionnalités qui seront ensuite développées en fonction de leur priorité.

ZOOM SUR LE SCRUM

Scrum est la méthodologie de développement agile la plus populaire (ce paragraphe devrait vous permettre de vous familiariser avec le vocabulaire associé à cette méthodologie).

Le cycle de vie Scrum est composé d'itérations de quatre semaines appelées les « sprints ». Le « product backlog », décrit avec le client, représente la liste des exigences initiales.

La méthode Scrum met en avant trois valeurs essentielles :

1. **La visibilité** : les résultats doivent être totalement soumis aux clients sans aucune interprétation possible pour maximiser la transparence. Tous les membres du projet doivent s'accorder sur les résultats souhaités.
2. **L'inspection** : il s'agit de la vérification des écarts avec l'objectif fixé initialement.
3. **L'adaptation** intervient lorsqu'on remarque qu'il y a un écart important entre les attentes et les résultats. Il faut finalement entreprendre des

ajustements afin de ne pas creuser les écarts entre les attentes et les résultats mais plutôt pour les effacer.

Scrum définit des rôles précis :

1. **Le ScrumMaster** prend le rôle de directeur de projet, il dirige l'équipe afin de respecter la mise en place des pratiques régies par Scrum. Le ScrumMaster se charge aussi de la résolution des problèmes.
2. **Le product owner** : il gère le product backlog (liste des exigences) et représente les utilisateurs. Il est alors chargé d'ajouter ou de supprimer de nouvelles exigences, ainsi que de les prioriser selon leur valeur ajoutée.
3. **L'équipe** : ses membres sont plurifonctionnels car ils travaillent tous ensemble à tous les niveaux durant chaque sprint. Il n'y a par exemple pas d'architecte ni de programmeur car ils le sont tous. L'équipe est responsable de ses propres décisions et elle gère son temps comme elle le souhaite.

Le daily scrum (ou mêlée quotidienne) est une réunion de planification quotidienne de 15 minutes maximum afin de faire une synthèse sur l'évolution du projet ainsi que ses difficultés. L'équipe développe rapidement trois sujets : l'avancement de la veille, ses projets du jour pour atteindre l'objectif du sprint et les difficultés.

La revue de sprint intervient à la fin du sprint. Durant une réunion de quelques heures (quatre au maximum), l'objectif est la validation de l'incrément de produit créé pendant la phase de sprint. Les éléments du carnet de produit choisis en tout début de sprint sont énoncés puis les éléments réalisés sont présentés. Il s'agit de procéder à un bilan du sprint. Le carnet de produit est ensuite remis à jour en fonction des éléments non réalisés. Un ajustement du budget et du financement est effectué puis la nouvelle version du carnet de produit est ajustée en conséquence.

PARTIE 2 - AVANTAGES ET INCONVENIENTS DE CES METHODES

Les avis divergent fortement autour de ces méthodologies. Chacune a ses partisans. Aussi, TheCodingMachine, qui propose les deux approches à ses clients, tente ici de faire une synthèse des arguments de chacun des camps.

L'ORGANISATION DES PROJETS

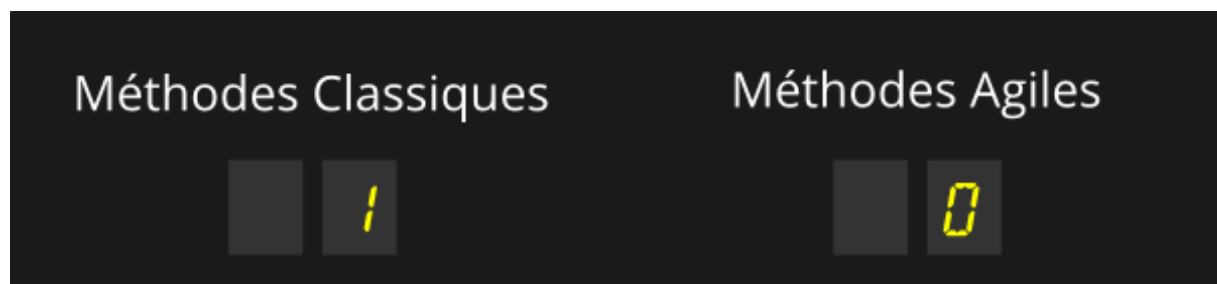
Les méthodes de gestion de projet classiques ont le gros avantage de proposer un résultat, un budget et un planning qui sont (ou au moins qui se prétendent) prévisibles. Même s'il est possible de penser qu'il s'agit souvent d'une prophétie auto-réalisatrice⁽¹⁾, l'expérience des chefs de projet et des développeurs fait que ce résultat, ce budget et la date de livraison sont, dans la plupart des cas tenus au moins partiellement. Cette capacité de rendre un projet prédictible permet d'organiser les ressources financières et humaines nécessaires à ce projet. Si l'on fait l'analogie avec les chantiers du BTP, ils sont rarement livrés dans les délais, ni même en respectant les budgets. Pourtant, il semble bel et bien impossible de partir sans plans détaillés ou chiffrages précis.

⁽¹⁾ prophétie qui modifie des comportements de telle sorte qu'ils font advenir ce que la prophétie annonce, vous pouvez regarder l'article de [Wikipédia](#) sur ce sujet ;-)

Concernant les méthodes agiles, puisque le postulat est de se laisser une certaine liberté, il est impossible d'organiser le projet autrement que par la méthodologie elle-même : distribuer les rôles de chacun, gérer les sprints et le backlog et organiser les différentes réunions.

Note : un outil permet de mesurer la productivité des développements et donc permet d'améliorer l'organisation / la prévision du projet. Cet outil s'appelle la « vélocité » qui permet de mesurer le nombre de fonctionnalités développées durant chaque sprint. Sur les projets longs, il est donc possible d'améliorer la prévisibilité du projet.

Le score du match est donc :



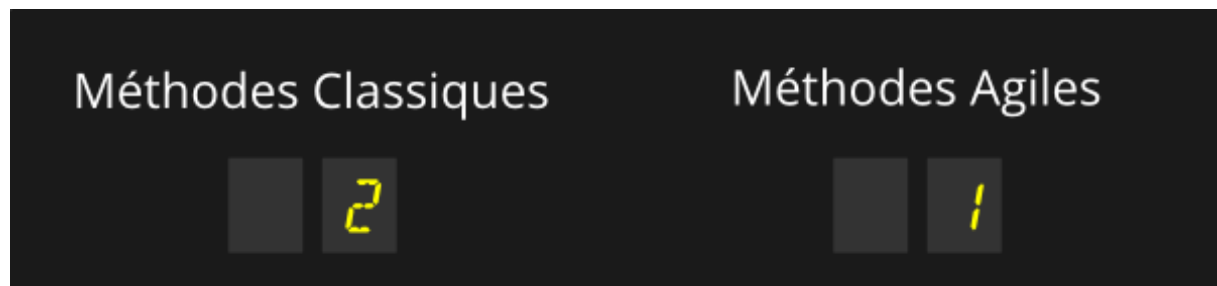
LA REUSSITE DES PROJETS

La plupart des projets gérés avec des méthodologies dites classiques rencontrent des difficultés (le fameux Chaos Report du Standish Group). Ce même rapport indique que statistiquement, les méthodes agiles réussissent mieux que les méthodes classiques : 29% de projet « failed » pour les méthodes classiques contre 9% pour les méthodes agiles.

En fait, beaucoup d'éléments sont discutables :

- ce que l'on mesure : dans le cas des méthodes classique, la réussite se juge à l'aune du respect des budgets et du délai tandis que ces deux éléments ne sont justement pas définis pour les méthodes agiles. Il semble donc qu'il soit un peu plus facile de parler de réussite de projet !
- la taille des projets : les projets qui expérimentent les méthodologies agiles sont certainement de taille plus modeste. Or, il est plus facile de réussir un petit projet qu'un gros !

Bref, pour nous égalité sur ce point, le score est donc de :



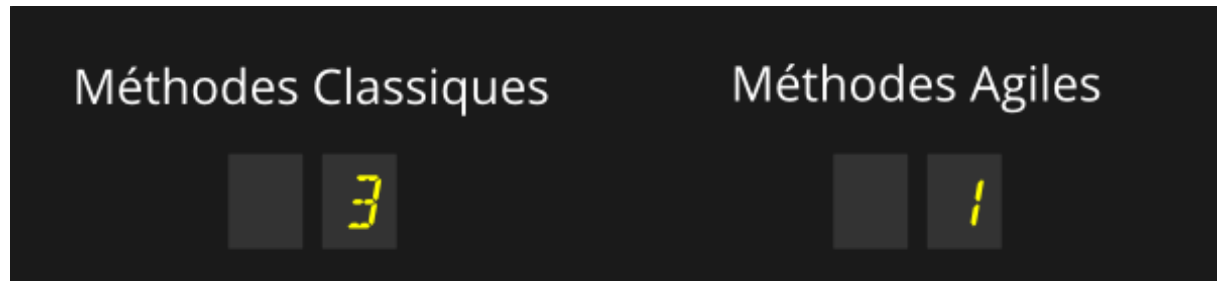
LE COUT GLOBAL DES PROJETS

Le coût côté client est souvent beaucoup plus important avec les méthodes agiles. Des ressources dédiées doivent souvent être affectées au projet. Pour certains prestataires peu scrupuleux, les méthodes agiles sont même parfois un argument commercial qui vise à accroître l'obligation de collaboration du client tout en déresponsabilisant le prestataire.

Côté prestataire, laisser la possibilité au client de changer d'avis, de revenir sur des décisions peut tout simplement empêcher le projet d'avancer et donc coûter beaucoup plus cher (là, c'est notre culture sur des projets au forfait qui parle un peu). Tandis que cette contrainte est plutôt bien gérée par les méthodes classiques très adaptées aux prestations au forfait.

Un dernier élément est la conception : il est moins coûteux pour le client d'essayer de se projeter dans l'application en passant du temps en conception que de partir directement dans les développements avec le risque de refaire.

Nouvel avantage pour les méthodes classiques :



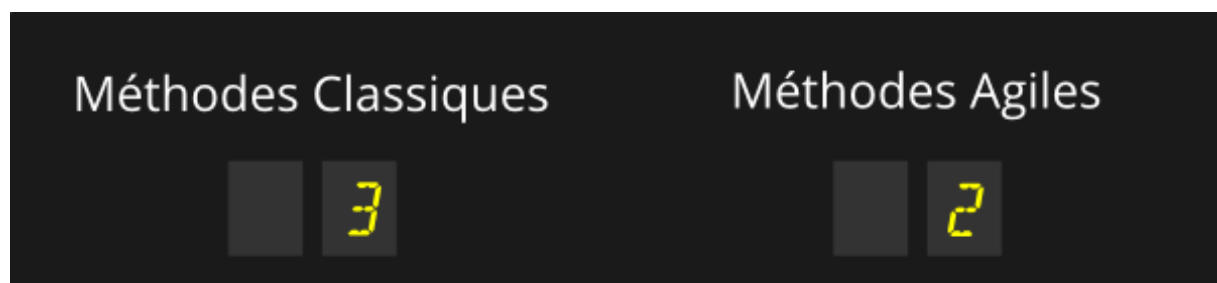
LA GESTION DU RISQUE

Les avantages en termes de gestion du risque pour les méthodes agiles sont :

- Les différents risques sont détectés très rapidement grâce aux nombreuses interactions avec le client. Ils sont donc rapidement sécurisés.
- La transparence entre le client et le projet permet d'éviter les incompréhensions et les incohérences fonctionnelles car si elles existent, elles sont traitées très tôt lors du projet et sont donc aisément corrigibles. De plus, la visibilité donnée au client lui permet de vérifier si les enjeux sont bien compris tout au long du projet. C'est un atout dans le cadre de projets complexes car les attentes et les besoins du client évoluent pendant le développement du projet.

Au final, le fait que les phases soient découpées en itérations permet de modifier facilement les premières versions développées alors qu'avec la méthode cycle en V, chaque caractéristique est développée de bout en bout, c'est donc plus difficile de la modifier.

Les méthodes agiles permettent de mieux gérer les risques :



LA SATISFACTION FINALE DES UTILISATEURS

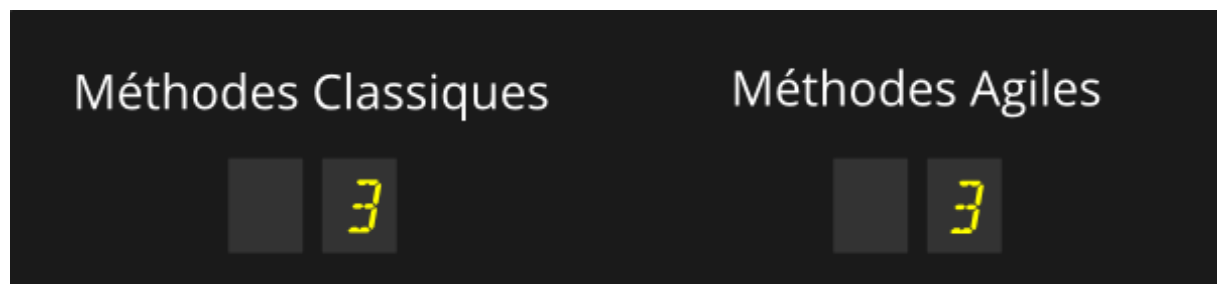
Après avoir établi les différents besoins et écrit les spécifications détaillées, on remarque souvent l'apparition d'un décalage entre ce qui était prévu et la réalité concrète du projet (surtout dans le cas de projets complexes). Les spécifications fixées peuvent être incomplètes ou irréalisables, d'autant plus que le client peut voir évoluer son besoin et donc souhaiter rajouter des fonctionnalités.

Pour les projets s'appuyant sur des méthodes classiques, la prise en compte de ces changements n'est en général possible qu'après négociation. Des intérêts opposés s'affrontent alors. Le client cherche à intégrer au périmètre ce qu'il considère comme « une conséquence logique » de ses besoins ou un défaut d'analyse initiale du prestataire. Tandis que le prestataire cherche à exclure tous les éléments ne faisant pas partie du périmètre contractuel.

Ainsi, les méthodes agiles, puisqu'elles proposent de procéder par étape, avec à la fin de chacune de ces étapes une validation du client, permettent d'éviter des déceptions ou un retour en arrière. C'est l'avantage d'une conception dynamique de la solution.

Il est cependant possible de donner une certaine souplesse aux projets en méthodes classiques en acceptant une contingence sur le projet. Ce budget prévu d'avance permet de se donner la latitude de changer certaines fonctionnalités tout en évitant des changements trop nombreux.

Malgré cela, le point est pour les méthodes agiles :

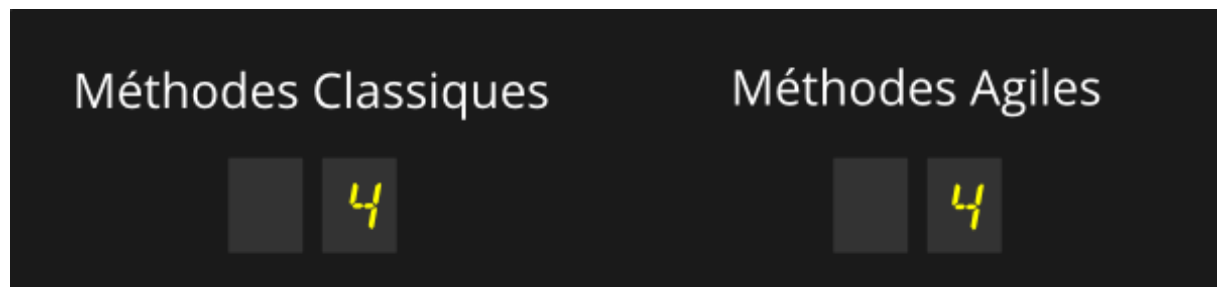


LA QUALITE DE L'APPLICATION

Paradoxalement, en fixant un cadre, les méthodes classiques peuvent ne pas laisser suffisamment de temps aux développeurs pour pouvoir produire du code de très bonne qualité. Les projets aux forfaits qui appliquent une méthodologie classique nécessitent souvent de courir derrière une deadline qui empêche de revenir sur des éléments déjà développés.

Une critique que l'on peut adresser aux méthodes agiles est le manque de documentation. Cela ne pose pas tellement de problème au moment du développement mais peut en poser plus tard lorsque la solution est exploitée, s'il faut reprendre le code.

Ici, égalité :



PARTIE 3 COMMENT CHOISIR ?

En renforçant la collaboration entre le client et le prestataire, en permettant de s'adapter simplement aux changements ou encore en favorisant un développement itératif, les méthodes agiles s'affirment de plus en plus comme une alternative crédible aux méthodologies classiques. Cependant, ces méthodes ne sont pas non plus un gage absolu de succès.

Voici donc quelques conseils pour choisir l'une ou l'autre de ces méthodes.

REFLECHISSEZ A VOS BESOINS

Est-ce que votre projet est une refonte ? Est-ce que la finalité du projet est bien définie ? La principale contrainte sur votre projet est-elle le budget ou le délai ? Est-ce que la satisfaction finale de vos utilisateurs est cruciale pour la réussite du projet ? Avez-vous des ressources en interne à affecter au projet ?

Posez-vous toutes les questions possibles et mettez-les en regard avec les avantages et les inconvénients de chacune (cf. partie 2). Cela vous donnera des premières pistes pour choisir.

Et puis réfléchissez à votre besoin. Même si c'est un peu réducteur, si vous ne savez pas parfaitement ce que vous cherchez, partir sur une méthodologie agile semble être une bonne option. Inversement, si votre besoin est très clairement défini, une méthodologie classique est certainement plus indiquée.

ET REFLECHISSEZ A VOS MODES DE FONCTIONNEMENT

Les méthodes agiles remettent en cause les principaux éléments contractuels classiques :

- périmètre fonctionnel,
- prix forfaitaire,
- planning détaillé.

Si ces éléments ne peuvent pas être remis en cause dans la culture de votre entreprise (ou plus prosaïquement par votre service juridique), engager un projet en mode agile risque d'être difficile. Sinon, apprêtez-vous à faire de nombreuses réunions d'évangélisation !

ET, ENFIN, COMMENT BIEN EMPLOYER CES METHODES ?

Des clients, en choisissant une méthodologie de gestion de projets agile ont parfois l'impression de ne pas savoir ce qui va être acheté au final. Si cela fait partie de vos préoccupations, vous pouvez aussi opter pour des alternatives qui combinent les deux !

Note : il est possible de « forfaire » de l'agile (Xebia propose un super contrat type [ici](#)). Cependant, ce contrat porte plus sur la méthodologie que sur les éléments tangibles du contrat (budget, planning) alors que ce sont ces éléments qui posent souvent problème.

CYCLE EN V EN CYCLE COURT (2 MOIS PAR EXEMPLE)

Une solution réside dans la conception de cycles en V relativement courts, ainsi que dans la réalisation de versions. Ainsi, les retours au sujet de la version 1 seront bénéfiques pour les versions ultérieures.

DEMARRER AVEC UN FORFAIT CLASSIQUE ET PUIS DERIVER VERS DE L'AGILE

Un projet d'abord réalisé grâce à la méthode du cycle en V peut s'appuyer sur les méthodes agiles une fois que la première version est en exploitation.

Cette combinaison permet au client de bénéficier du contrôle des coûts et du périmètre d'intervention du cycle en V durant la phase de lancement, puis de la conception flexible des méthodes agiles une fois le projet exploité. Cette approche apporte de nombreux avantages, le client participe activement à toutes les phases et peut voir intégrer ses évolutions très rapidement.

UN EXEMPLE DE PROJET QUI COMBINE CLASSIQUE ET AGILE AU SEIN DE THECODINGMACHINE :

Le projet Webikeo

La société Webikeo organise des conférences en ligne dans le monde entier. TheCodingMachine a totalement repensé et refondu le site <http://www.webikeo.fr> et travaille toujours pour Webikeo dans le cadre d'une méthodologie qui est passée d'un mode classique à un mode agile.

Au début du projet, Webikeo avait clairement défini ses besoins. Nous sommes donc partis sur une méthodologie classique (spécification, développements et tests). Le cycle en V a permis de mettre en production le site très rapidement au bout de trois mois.

Dans un deuxième temps, lors de l'exploitation de la solution, Webikeo qui avait accordé sa confiance à TheCodingMachine a souhaité mettre en place un modèle de conception plus dynamique. Nous avons donc proposé de fonctionner sous un modèle agile pour répondre aux nouveaux besoins. Les nouvelles fonctionnalités sont à présent intégrées très rapidement. Webikeo bénéficie donc d'une visibilité à court terme sur ses fonctionnalités et peut décider de les laisser en état, ou les modifier, ou encore les supprimer sans perdre de temps. Les coûts sont également contrôlés par Webikeo.

CONCLUSION

La meilleure méthodologie n'assurera pas nécessairement le succès du projet. Un peu comme le meilleur agenda ne vous garantit pas d'arriver à l'heure à vos rendez-vous !

En revanche, les méthodes agiles déploient de nombreux facteurs qui favorisent le succès des projets. Pour en citer quelques-uns :

- la capacité à livrer rapidement une première version pour que le client se rende compte du produit final ;
- l'implication nécessaire du client dans toutes les phases du projet ;
- la capacité à négocier de manière responsable entre client et prestataire afin de gérer le changement.

De manière indirecte, le succès des méthodes agiles, c'est aussi mettre la technologie au centre du projet. Nous devons avouer que cela nous plaît beaucoup chez TheCodingMachine.

ANNEXE – FONDEMENTS DES METHODES AGILES

LES ORIGINES DES METHODES AGILES

Le mouvement agile est créé en 2001 aux Etats-Unis par dix-sept experts en développement de logiciels. Ces derniers avaient remarqué un taux d'échec de projets considérable et avaient déjà réfléchi chacun de leur côté à de nouvelles méthodes de gestion de projets.

LES PRINCIPES ET LES VALEURS DU MANIFESTE

Les douze principes du *Manifeste* pour le développement agile de logiciels :

1. La livraison très tôt de versions opérationnelles du projet afin d'atteindre une transparence maximale.
2. Le changement des décisions du client doit toujours être accepté, quel que soit le moment dans le cycle de vie du projet.
3. La livraison tous les mois maximum d'une version opérationnelle du projet au client.
4. Les développeurs doivent coopérer, tous les jours, pendant le projet.
5. Le volontariat est la base de l'équipe et non plus le chef de projet. Les membres d'équipe choisissent eux-mêmes leurs futurs travaux au sein du projet.
6. Les membres de l'équipe doivent être motivés car il n'y a pas de chef de projet. Les membres de l'équipe doivent donc se gérer eux-mêmes.
7. L'oral est privilégié par rapport à l'écrit. Il faut communiquer le plus possible en face à face en interne afin de faciliter la rapidité de compréhension.
8. Le rythme de travail du projet doit être soutenu tout en étant soutenable.
9. Le rythme de travail doit toujours être constant et doit être déterminé en amont par l'ensemble de l'équipe. Travailler en heures supplémentaires n'apporte, donc, pas de valeur ajoutée.
10. Le maintien d'un code évolutif, performant et décrit est essentiel. La production d'un code jetable ou non exploitable est donc à bannir.
11. L'itération doit se faire et doit être présentée au client de la manière la plus simple possible car la clarté simplifie l'évolutivité tandis que la complexité rend les évolutions plus difficiles.
12. L'équipe se remet en question, assez régulièrement, et trouve une manière de gagner en efficacité (même si l'équipe est déjà efficace). Cela permet de s'organiser à propos des nouvelles conditions voulues par le client.

Le *Manifeste* fait état de quatre valeurs essentielles à toutes méthodes agiles :

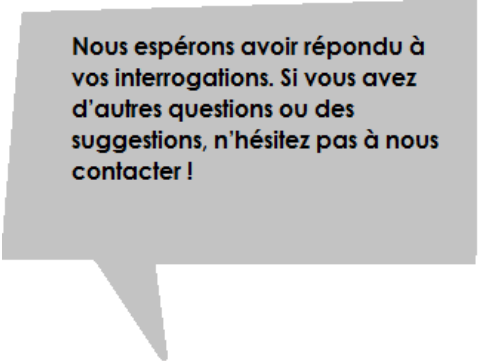
1. Les individus et leurs interactions passent avant les processus et les outils.
2. Les fonctionnalités opérationnelles passent avant la documentation.
3. La collaboration avec le client est prioritaire par rapport à la contractualisation des relations.
4. L'acceptation du changement est essentielle plutôt que la conformité aux plans.

A PROPOS DE THECODINGMACHINE

TheCodingMachine accompagne ses clients sur des missions de conseil technologique et sur des projets de développement d'applications Web.

Nous sommes spécialisés dans le développement de sites Internet, d'extranets (évidemment), d'intranets, d'applications Web métiers en PHP et en JavaScript.

Fondée en 2005, TheCodingMachine a piloté plus de 300 projets. Nous travaillons aussi bien pour des grands comptes privés et publics, pour des PME-PMI que pour des startups. Nous avons investi dès notre création dans la R&D, ce qui nous permet par exemple d'être à la pointe des technologies web (PHP, Node.JS, AngularJS, streaming video etc.).



Nous espérons avoir répondu à vos interrogations. Si vous avez d'autres questions ou des suggestions, n'hésitez pas à nous contacter !

www.thecodingmachine.com

Tél : 01 71 18 39 73

contact@thecodingmachine.com

4, rue de la Michodière – 75002 PARIS