



Scrum, kamban... : Les méthodes Agile

Introduction

Les méthodes *Agile*, car il faut bien parler de méthodes au pluriel, sont issues de recherches universitaires US sur l'optimisation des procédés et des organisations avec un objectif « client », tout comme le « lean management ». A la différence du « lean » plutôt dédié à la production de biens matériels comme les voitures (avec des techniques spécifiques de suivi et d'amélioration de la qualité), les méthodes Agile sont dédiées - et en principe plus adaptées - à la production de biens immatériels comme le logiciel et à la gestion.

Principes

Les méthodes Agile ont pour principes de valoriser les individus et leurs interactions plus que les processus et les outils ; des logiciels opérationnels plus qu'une documentation exhaustive ; la collaboration avec les clients plus que la négociation contractuelle ; l'adaptation au changement plus que le suivi d'un plan.

Tout en reconnaissant la valeur des seconds éléments, *Agile* privilégie les premiers. Ces principes semblent mieux tenir compte de l'aspect humain et de sa capacité d'adaptation, ce qui est en soi une bonne chose, mais on constate que l'on cherche aussi par le fait à appliquer dans les grandes entreprises des modes de fonctionnement de petites structures. C'est là qu'il faut être vigilant : la mobilité des personnels et les interactions multiples entre entités de ces grandes entreprises nécessitent des interfaces suffisamment formalisées donc écrites, pour pouvoir fonctionner dans les meilleures conditions.



Objectifs

Les objectifs sont pilotés par la satisfaction du client, qui est la plus haute priorité. Pour ce faire il faut :

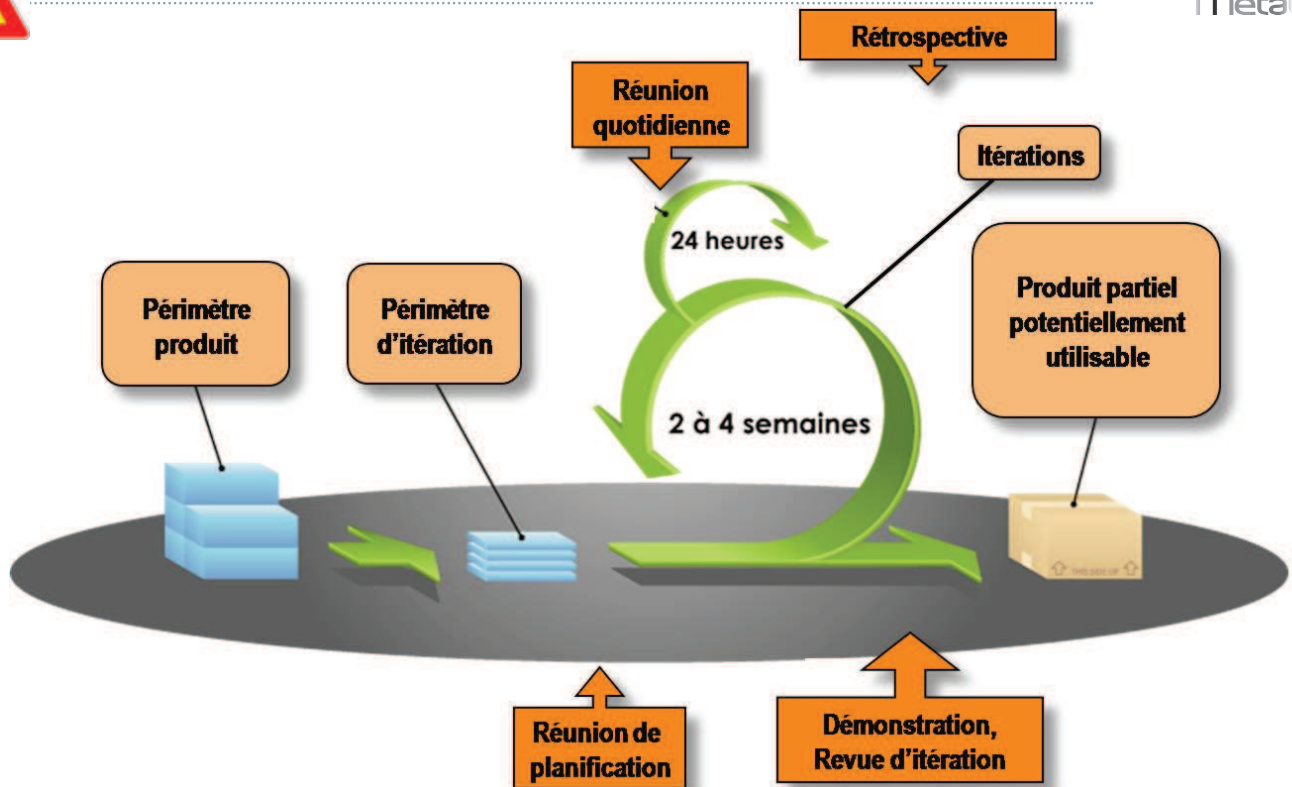
- > Livrer rapidement des fonctionnalités privilégiant la plus grande valeur ajoutée
- > Livrer selon un rythme régulier et constant avec des cycles les plus courts possibles
- > Viser constamment la minimisation de la quantité de travail inutile à tous les niveaux
- > Accueillir positivement les changements de besoins, même tard dans le projet
- > Un fonctionnement des équipes le plus possible responsabilisé et autonome avec des réunions périodiques sur l'amélioration de ses procédés
- > Privilégier la transmission d'informations en face à face à l'intérieur des équipes.

La responsabilisation des équipes est une bonne chose, mais on constate que les contraintes de régularité sont issues du taylorisme avec les risques psychosociaux que cela comporte, si les équipes sont sous-dimensionnées, ce qui est souvent la tendance actuelle. La dimension des équipes a aussi un très gros impact sur la capacité d'adaptation au changement et par voie de conséquence sur les RPS, surtout en cas de répétition excessive et mal contrôlée de telles situations.

Procédés généraux

Les idées à l'origine des procédés listés ci-dessous viennent d'experts Agile différents, d'où la complexité de leur mise en œuvre commune.

- > Faire de la conception et développement itératif, qui permet de mieux s'adapter au changement
- > Développement piloté par le test : le développeur travaille alternativement sur le développement des tests unitaires (automatisation) et le développement proprement dit
- > Intégration continue des livraisons itératives
- > Réunion « rétrospectives » d'amélioration :
 - Qu'est-ce qui a bien marché ?
 - Que devrions-nous faire différemment ou arrêter de faire ?
 - Qu'avons-nous appris ?
 - Que devons-nous approfondir (pour mieux comprendre et améliorer, notion d'itération sur les points sur lesquels on ne peut statuer dans la réunion courante)
- > Techniques diverses (plusieurs sources) de planning et d'estimation. Un exemple : écrire chaque besoin élémentaire sous forme « d'histoires ». Si l'estimation a été trop courte, l'histoire est coupée



en deux, si elle a été trop longue, on l'associe à une autre histoire...

On s'aperçoit ici que la pression risque de s'appliquer essentiellement sur le développeur qui prend en charge aussi le développement des tests unitaires, selon comment s'applique l'exigence des livraisons itératives.

SCRUM

Scrum (mêlée en anglais) est une approche légère et hautement itérative de type *Agile* qui est mise en œuvre à Alcatel-Lucent.

Ce n'est pas une méthode de développement complète, c'est plutôt un ensemble de pratiques de gestion de projet qui peuvent être utilisées conjointement avec diverses méthodologies logiciel. Les auteurs la définissent comme un cadre de travail à suivre par les équipes dans l'organisation de leur travail de développement logiciel.

Les principales caractéristiques sont :

Le **Scrum Master** responsable de la gestion des pratiques *scrum* dans une équipe

Le **Product Backlog** liste de toutes les fonctionnalités affaires (vue du client) et techniques, listées par

ordre de priorité

Le développement est fait par des équipes de 5 à 9 membres travaillant sur les différentes fonctionnalités du *Backlog*

Toute l'équipe travaille sur une itération d'un groupe de fonctionnalités pendant le temps imparti à cette activité appelée *sprint*

L'équipe crée son propre backlog ou découpe en sous-tâches des fonctionnalités à réaliser dans le *sprint*. Pendant le *sprint*, l'équipe tient tous les jours une réunion permettant de donner l'état d'avancement de la journée écoulée, identifier les obstacles et vérifier ce qui doit être fait le jour suivant. Tout sujet important ou bloquant est laissé à d'autres réunions.

Aucune personne extérieure ne peut changer l'étendue et la nature du travail à exécuter pendant un *sprint*.

Chaque *sprint* se termine par une réunion « *rétrospective* » (voir plus haut).

Cette méthode met bien en évidence l'exigence accrue de contrôle et le travail supplémentaire que cela représente : cette méthode ne peut en aucun cas justifier des effectifs en réduction sans risque d'explosion des RPS !

KANBAN

Cette méthode *Agile* a surtout été utilisée aux US chez Alcatel-Lucent.

Kanban est une méthode *Agile* de gestion de processus développement logiciel autour du concept d'imposer des limites sur les tâches en cours (*WIP* ou *Work In Progress*).

Les projets utilisent un tableau présentant l'avancement des tâches et leurs limites. Il montre aussi la capacité attendue de l'équipe à exécuter les tâches dans le planning global. Cette méthode a déjà montré des limites en particulier sur l'aspect collaboration à travers l'organisation. Elle ne fonctionne pas par itérations, elle est plus proche du *lean manufacturing*.

Conclusion

La tendance actuelle est à la réduction des coûts et, bien souvent, à celle des effectifs. C'est bien là que se situe le problème : les méthodes peuvent induire des optimisations de par leur application mais pas a priori. Il faut donc s'attendre à une montée forte des RPS, ce que nous constatons dans les chiffres récents sur Alcatel Vélizy.