

# L'agilité essentielle

Compte rendu de la rencontre « autour d'un verre » du lundi 6 octobre 2008  
animée par Jean-Pierre Vickoff

Rapporté par Martine Otter

Jean-Pierre Vickoff, adhérent d'ADELI, a présenté dans nos colonnes, dès juillet 2002, la méthode PUMA (Proposition pour l'Unification des Méthodes Agiles), dont il a développé différents aspects dans les Lettres 68 et 69. Dans le présent numéro il nous propose une nouvelle version « allégée » de cette méthodologie. La rencontre du 6 octobre fut l'occasion de revenir globalement sur le thème de l'agilité.

## L'agilité

Jean-Pierre Vickoff introduit son exposé en nous présentant sa vision de l'agilité : « faire parler les clients ». Un conférencier agile, explique-t-il en forme de boutade, aurait du commencer par nous faire nous exprimer sur nos centres d'intérêt, en dresser la liste sur un tableau mural et nous demander de les prioriser.

Il en profite pour nous présenter les outils pratiques de l'animateur agile que sont les jeux de carte et les murs de la pièce. Dans l'impossibilité de taguer les murs de la Villa Maillot nous nous trouvons contraints d'en rester à un mode de communication plus classique.

## Bref historique de l'agilité

Même si certains font remonter l'agilité aux années 60, le concept n'est apparu chez Barry W. Boehm qu'en 1986 avec l'invention d'un nouveau modèle de développement itératif et incrémental. Les méthodes structurées des années 70, fondées sur un modèle en cascade, commençaient à paraître trop lourdes, productrices de trop de documents et de trop d'étapes, sources du fameux effet tunnel.

Le temps de réalisation des documents requis par ces méthodologies était devenu inadéquat aux nouvelles contraintes de délai. Le concept fut repris par James Martin qui proposa la méthodologie RAD (Rapid Application Development) en 1991, suivant un principe de validation permanente des utilisateurs.

Jean-Pierre Vickoff nous indique qu'il a commencé à mener des développements selon le principe itératif-adaptatif dès 1989 au Canada, en s'appuyant sur les premières communications de James Martin. C'est à partir de 1994 qu'il s'est totalement consacré à l'évolution des méthodes Agiles.

L'eXtreme Programming a été inventée par Kent Beck, Ward Cunningham et Ron Jeffries dans le cadre d'un projet de calcul des rémunérations chez Chrysler et publiée officiellement en 1999 avec le livre « eXtreme Programming Explained » de Kent Beck.

En 2001, plusieurs chefs de projet américains se sont réunis pour structurer ces démarches et ont publié l'« agile manifesto » qui présentait 4 valeurs se déclinant en 12 principes<sup>1</sup> et marque le début de l'explosion du mouvement agile.

Jean-Pierre Vickoff distingue ainsi trois époques successives dans l'évolution des systèmes d'information :

- les années « matériel », marquées par la recherche de la puissance de calcul ;
- les années « logiciel », que l'on pourrait également appeler les « années Microsoft » ;
- les années « agile », entamées par le rachat de Rational par IBM.

## De quoi s'agit-il ?

Le premier effet de l'« agilité » est le raccourcissement du cycle de développement : on plonge tout de suite dans la réalisation. Les phases d'expression des besoins et de planification sont raccourcies.

La programmation extrême trouve ses limites dans l'impasse faite sur l'expression des exigences, du point de vue non seulement technique mais également de la communication. Les grandes compagnies sont habituées à des processus longs de validation de documents. Le niveau de complexité des grandes structures où beaucoup d'intervenants ont leur mot à dire, nécessite une communication transverse qu'une méthode totalement extrême ne permet plus.

L'eXtreme Programming, XP en abrégé, répond au besoin d'ingénierie du logiciel. Scrum<sup>2</sup>, apparue dès 1995, apporte en plus la couche pilotage de projets. Le couplage de l'eXtreme programming avec Scrum permet de maîtriser le développement, mais il manque encore à cette combinaison une expression solide des exigences, nécessaire dans les projets complexes. La faiblesse actuelle du

<sup>1</sup> Sur les valeurs et principes de l'Extreme Programming, on pourra se reporter à l'article « Méthode Agile » de Wikipedia

<sup>2</sup> Le terme SCRUM, ou « mêlée » en anglais, est issu du vocabulaire du rugby.

mouvement agile est bien ce manque d'expression des exigences.

La méthode PUMA, créée par Jean-Pierre Vickoff, se présente comme un processus organisant les méthodes agiles. La tentative d'unification, d'abord proposée par son auteur, n'a pas rencontré le succès escompté. C'est ainsi que le U comme unification de Puma est devenu en 2007 U comme Urbanisation, avec l'ambition de couvrir l'ensemble des trois volets du génie logiciel, du management de projet et de l'expression des exigences.

L'Entreprise agile avec ses six grandes étapes imbriquées est apparue encore trop complexe aux DSI pariant sur l'agilité. Face à cette demande de simplicité, PUMA s'est donc allégé pour revenir à l'essentiel avec un minimum de pratiques. Le résultat est la méthode PUMA ESSENTIEL que Jean-Pierre Vickoff nous présente dans cette même Lettre.

Puma ESSENTIEL se compose ainsi de quatre moteurs, à chacun desquels sont associées quatre pratiques :

1. un moteur de communication pour faciliter l'engagement et le consensus.
2. un moteur de solution pour structurer l'expression de l'exigence et de la solution.
3. un moteur de projet, de type Scrum, pour piloter l'évolution de la performance de l'engagement ;
4. un moteur de réalisation, de type XP, pour assurer la qualité fonctionnel et technique de la solution.

Jean-Pierre Vickoff nous apporte quelques éclairages sur ces pratiques qui sont décrites dans l'article PUMA ESSENTIEL. Il nous précise qu'il est quelquefois nécessaire de combiner des pratiques agiles avec d'autres pratiques plus classiques. Par exemple, pour l'estimation des charges, on utilise un jeu de cartes pour collecter les réponses de l'équipe, dont on fait ensuite la moyenne. Ceci permet d'obtenir un engagement fort des développeurs mais doit être recoupé avec d'autres techniques d'évaluations, par exemple le décompte des cas d'utilisation.

En théorie, le reporting classique n'est plus nécessaire, puisque tout est visualisé sous forme d'un affichage mural : nombre de fonctionnalités prévues et nombre de fonctionnalités restantes, on sait toujours où l'on en est, la mise à jour se fait en direct sur le mur. C'est le client qui est responsable de la priorisation des fonctionnalités. S'il abandonne une fonctionnalité déjà entamée, c'est lui qui supporte le temps consommé sur la fonction abandonnée. En contrepartie cela l'autorise à changer d'avis même en cours de développement.

## Limites et difficultés

---

Il est clair que ces pratiques sont difficiles à faire rentrer dans le cadre juridique contraignant des marchés publics.

Une véritable révolution culturelle est à l'œuvre lorsque les bonnes pratiques de l'agilité telles que la programmation en binôme (pair programming) viennent se heurter à l'individualisme des développeurs : imaginez deux développeurs utilisant un même PC avec deux écrans et deux claviers. De même, les techniques dites de refactoring permanent, ou amélioration continue du code, et d'intégration continue des développements viennent bouleverser les habitudes acquises dans les cycles de développement en cascade.

Un saut culturel que certaines entreprises et certains individus ne sont pas prêts à franchir. Pourtant qualité du code et réduction du nombre de bogues sont bien au rendez-vous, avec un gain de qualité minimum de 30%...au prix d'une augmentation de coût de 10% à 20%. Là encore, il peut être difficile de faire comprendre aux acheteurs la différence de prix entre deux compétiteurs lorsque les méthodes de développement utilisées ne font pas partie des exigences de l'appel d'offres.

Toutes les structures ne sont pas compatibles avec l'agilité.

## Les conditions du succès

---

Le développement agile se fait sous la pression permanente des utilisateurs. Cela nécessite que les concepteurs-développeurs soient hyper motivés. La pression doit être compensée par le plaisir de produire du code propre, et tout le monde n'est pas forcément motivé.

Des pays comme le Canada et les États-unis ont vécu de véritables big-bangs agiles, mais en France, pour vaincre les résistances au changement, il faut y aller progressivement, en commençant par des petits projets.

Un petit projet après l'autre, pour démontrer le gain de productivité.

Une condition de réussite du déploiement des méthodes agiles est le retour à la simplicité, avec des techniques de communication par affichage mural, en attendant, précise Jean-Pierre Vickoff, la mise à disposition d'écrans muraux.

Parmi les critiques adressées aux méthodes agiles on peut citer l'absence de modélisation. Cela est faux, même si la modélisation UML peut prendre la forme de bouts de carton représentant les classes, affichés sur un mur avec des épingles. Ce type de modélisation interactive, effectuée à plusieurs, est impossible à réaliser sur un écran avec les outils actuellement utilisés.

En théorie il n'y a pas de chef de projet dans un projet agile, mais un animateur facilitateur responsable vis-à-vis de la direction. Les entreprises ne sont pas convaincues par ce mode d'organisation, qui sort des définitions de fonction et grilles de salaire. Dans nos cultures fortement hiérarchisées, il faut savoir qui est responsable. Même si l'on ne parle plus de chef de projet, il n'est pas question de confier des responsabilités sans évaluation des compétences...

### **Petit vocabulaire agile**

Il faut s'y faire, le français est incontournable dans les méthodes agiles. Il présenterait même l'avantage d'offrir un facteur de rupture.

Le « sprint » dont la définition de base est « Course de vitesse sur une courte distance » est l'itération qui produit une nouvelle version de l'appliquatif.

Le « backlog » est un journal qui contient la liste des fonctionnalités du produit à réaliser.

## **De multiples questions**

---

### **L'agilité pour quels projets ?**

Chaque cas est un cas particulier. Entre le cycle en cascade et l'itératif, il y a place pour du semi itératif. Un petit projet peut être entièrement conduit en mode itératif, un gros projet nécessitera une étape préalable.

La part plus ou moins grande des phases préalables d'exploration et de planification dépend de l'entreprise et du contexte projet.

La méthode doit s'adapter à la taille du projet. Décomposer les grands projets en petits projets ne simplifie pas le problème, car se pose alors la question des interfaces entre tous ces petits projets.

### **Compatibilité avec CMMI et les méthodes de management de projet ?**

CMMI et méthodes agiles sont-ils compatibles ou bien appartiennent-ils définitivement à des mondes différents ? Les méthodes agiles peuvent prétendre couvrir le niveau 2 de CMM avec les pratiques de gestion des exigences, de reporting (mesures et analyse), surveillance et contrôle du projet, planification, assurance qualité, gestion des fournisseurs ou encore de gestion de configuration. L'atteinte du niveau 3 est plus problématique<sup>1</sup> car elle suppose une approche processus : le processus défini doit être respecté. Dans une approche agile on est amené à modifier

---

<sup>1</sup> Sur ce sujet, est paru en novembre 2008 une note du SEI tendant à démystifier l'opposition entre les deux approches, souvent faite par des personnes ne maîtrisant aucune des 2 !  
Technical Note  
CMU/SEI-2008-TN-003  
CMMI® or Agile: Why Not Embrace Both !

constamment le processus défini, le respect du processus pouvant être contre productif. On se moque un peu des processus car on vise le résultat.

Si l'on parle de maturité dans l'approche agile, c'est plus de celle des personnes et des organisations que de celle des processus. On constate, en France, que les tentatives et échecs de déploiement de méthodes agiles sont plutôt corrélés avec la maturité de l'organisation.

### **Et les métiers ?**

On parle aujourd'hui de concepteurs-développeurs. Les méthodes agiles, on l'a vu, suppriment la fonction de chef de projet, l'équipe fonctionnant sur un mode auto géré. Cela pose clairement le problème de reconnaissance par le titre.

D'autres méthodologies, telle Praxeme, continuent à se développer, sur des bases non agiles. Ces méthodologies répondent aux besoins de développement de grands applicatifs, pour lesquels les phases de modélisation amont sont importantes, la solution étant alors de générer du code directement à partir des modèles.

Pour faire face à la différenciation concurrentielle, il y aurait donc deux grandes voies pour le développement des applicatifs :

- modéliser et générer du code ;
- renoncer à la modélisation, trop lourde, produire et mettre en œuvre le plus vite possible.

### **Que se passe-t-il une fois en maintenance ?**

Une fois l'appliquatif passé en production, peut-on encore rester agile ? On pourrait envisager la maintenance comme une simple itération supplémentaire, mais ceux qui ont fait le développement sont-ils encore là pour la maintenance ?

Si l'on veut anticiper, il faut pouvoir intégrer dans l'équipe de développement quelqu'un qui puisse ensuite être motivé par la maintenance. Si le code est bon, c'est plus facile de le maintenir. Les développements agiles sont avarés en documentation : pour réaliser des modifications importantes, par exemple en cas d'évolution réglementaire, on sera contraints de plonger dans le code ou de faire appel à des outils d'analyse d'impact.

### **Quels outils ?**

Les méthodes agiles ne recommandent aucun langage ni aucun outil particulier. Même s'il existe des frameworks de test, les outils sont secondaires, l'agilité est d'abord un état d'esprit, une façon de penser.

## Méthode agile et certification ?

Existe-t-il une certification pour les développeurs agiles ? Même s'il existe bien un processus de certification des Scrum master développé par la Scrum alliance ([www.scrumalliance.org](http://www.scrumalliance.org)), cette question fait hurler les puristes ! Jean-Pierre Vickoff précise que cela n'est pas utile, l'appartenance à la famille des développeurs agiles se fait par cooptation. La programmation en binôme permet de lisser et d'équilibrer les niveaux. La montée en compétences des développeurs est permanente, ce qui diminue le risque humain. Chacun code ou teste à son tour, les binômes sont en rotation permanente, mêlant débutants et confirmés, intravertis et extravertis, suivant différentes combinaisons.

## Management et entreprise agile ?

Le concept d'entreprise agile va beaucoup plus loin que celui de projet agile, en déployant le management total par la qualité de service. L'Entreprise Agile est « orientée service », c'est une organisation au sein de laquelle les hommes, les processus et le SI inter-réagissent dynamiquement en fonction des évolutions « métier ». Cela est possible, Jean-Pierre Vickoff a pu le constater dans les années passées au Canada, mais, en France, il constate que le concept de reengineering fait peur. Dans un pays comme le Canada, il y a moins de différences hiérarchiques, un plus grand respect des techniciens, l'échelle des salaires est plus resserrée.

## Conclusion

---

Un seul grand principe, ne pas réinventer la roue : prenons le meilleur de chaque méthode selon le projet et prenons ce dont nous avons besoin, en construisant notre méthode à la carte. Il vaut mieux rechercher la simplicité plutôt que la structuration et se centrer sur les risques.

« Faire évoluer l'état de l'art en comblant les failles rencontrées sur le terrain », ce message résume l'approche pragmatique de Jean-Pierre Vickoff. ▲

*vickoff@noos.fr*  
*martine.otter@adeli.org*