

Agile : Controverse et réflexions

Source et antériorité des publications

Dans son dossier consacré aux méthodes Agiles, 01 Informatique¹ a su remettre en dates l'évolution du pilotage de projet basé sur l'approche itérative, incrémentale et adaptative. En clair, rendre à César ce qui est à César, ou, en l'occurrence, rendre à la méthode RAD et à James Martin son fondateur, la primeur d'initiation de ce courant de pensées.

Cette mise au point est techniquement secondaire, mais historiquement importante pour la compréhension de ceux dont l'arrivée dans ce grand mouvement technologique et humaniste est récente.

Ce qui détermine le changement de paradigme en conduite de projet c'est le passage du « cascade » à l'itératif, incrémental et adaptatif.

La première méthode basée sur cette rupture est la méthode RAD de James Martin, publiée en 1991. Ensuite DSDM pour le côté anglophone, a pris la relève de cette évolution en termes de techniques.

A partir de 1991 des chefs de projets du monde entier et particulièrement aux US ont utilisé la méthode RAD. Obligatoirement les contributeurs à l'évolution des méthodes suivantes ont eu connaissance de ces travaux. Ce n'est pas la méconnaissance des recherches de Pierre et Marie Curie qui doit me donner le droit de revendiquer la découverte de la radioactivité.

Quant au livre « RAD » de James Martin, c'est un monument d'une complétude impossible à imaginer pour ceux qui n'ont pas eu la chance de pouvoir le consulter.

Les deux successeurs immédiats furent Jean-Pierre Vickoff (1994) pour le côté francophone et Jennifer Stapleton (1995) pour DSDM, la version anglaise du RAD.

Ce n'est qu'en 1996 que Ken Shwarber et Jeff Sutherland dévoilèrent les bases de *Scrum* et en 1999 que Kent Back et Ron Jeffrie proposèrent de systématiser, en les poussant à l'extrême, certaines techniques de qualité du génie logiciel.

Dans la plupart des présentations des sociétés les plus importantes que j'ai pu trouver sur le Web, ces dates sont erronées quand ce n'est pas

¹ Numéro du 23 Octobre 2008

directement trafiquées. Ce manque d'éthique en matière de communication technique est lamentable.

Beaucoup de jeunes informaticiens ne le savent pas, mais vers le milieu des années 90, les premières pages de certains sites des chefs de projets américains qui travaillaient sur leurs propres méthodes, mais qui avaient été devancés par James Martin (anglais), lui faisaient ouvertement la guerre, donc connaissaient ses travaux. Le principal reproche était (officiellement) qu'il n'était pas un chef de projet de terrain mais un universitaire appuyé par des sherpas et que son site débutait par une phrase agaçante du style « James Martin a écrit plus de livres que tout autre chercheur encore vivant. ».

Il m'a été reproché sur des blogs dédiés à *Scrum* de ne pas citer de références reliant *Scrum* à RAD pour justifier sa filiation. Ce n'est pas aux prédécesseurs de donner des références, qui, à l'évidence, n'existent pas. Refuser l'antériorité des publications sur un sujet (ce qui est la base même de la communication scientifique), n'empêche pas ces communications d'exister.

Lors d'une discussion sur un blog français l'auteur a posté « Alors, Schwaber et Sutherland ont-ils pompé sur Martin sans le citer ? La réponse n'a que peu d'importance. Tous ces travaux datent de la même période et leurs influences sont forcément multiples. » C'est faux, ces travaux ne datent pas du tout de la même époque :

- James Martin publication méthode RAD complète en 1991² ;
- *Scrum* ébauche en 1995, publication 1996 ;
- XP publication 1999.

En informatique 4 ou 8 ans c'est une éternité de projets.

Au final, c'est aux initiateurs de *Scrum* de mettre en évidence les ruptures fondamentales qui les affranchiraient de référencer leurs prédécesseurs. C'est comme cela que fonctionne la communication scientifique.

Le révisionnisme qui consiste à vouloir supprimer les références à la première méthode itérative-incrémentale et adaptative ne résistera pas à l'histoire et au temps. Il n'y a rien de bien dans notre domaine que des tentatives pareilles peuvent exister.

² 1991 officiellement mais j'avais lu ses premières communications au Canada bien avant.

Pour le Gartner Group qui s'est penché sur le sujet depuis ses origines, les méthodes Agiles actuelles sont simplement « RAD ou néoRAD ».

Il n'y a aucune rupture méthodologique dans XP ou *Scrum*, juste une amélioration de l'usage des techniques de qualité et de conduite de projet adaptative. Je serais d'ailleurs vraiment curieux de savoir qui avait vraiment entendu parler de *Scrum* ou de XP en France avant 2000 et avant que Jean-louis Bénard n'en fasse une étude comparative.

Le refus de s'appuyer sur les expériences d'un passé aussi proche afin d'éviter de renouveler les mêmes erreurs et de devoir reformuler les mêmes solutions, représente un formidable risque pour les projets en cours ainsi qu'une formidable perte de temps pour la profession. Et cela n'est pas du tout Agile !

Questions gênantes pour méthodes « phares »

Devant l'aspect actuellement « incomplet » des méthodes Agiles, il n'est pas interdit de se poser deux ou trois questions simples.

- 1 - Pourquoi XP ne propose-t-il pas une forme d'expression structurée des exigences permettant de décrire les divers aspects d'une application de niveau « système d'information » ? Réponse possible : le projet qui a conduit à formaliser XP étant une refonte de paie, il est probable que ces aspects n'étaient pas une préoccupation du moment.

- 2 - Pourquoi *Scrum* n'implique-t-il pas de techniques de développement ? Les certifications de *Scrum*master s'obtenant en quelques heures et sans prérequis, il serait logique et charitable de penser que les initiateurs de cette méthode étaient des chefs de projets ne développant pas, ou plus, eux-mêmes.

Au niveau du champ d'application des méthodes, il serait aussi très judicieux de considérer que si 18% seulement des exigences applicatives peuvent être fixées en début de projet, ce qui implique que les méthodes en cascade sont inadaptées pour 82% des projets, en contrepartie il n'y a certainement pas plus de 18% des applications qui peuvent faire l'objet d'un développement totalement itératif.

Les controverses actuelles sur le sujet

L'histoire est sensée nous rappeler notre passé afin que nous n'en répétions pas les erreurs. Ce principe fonctionne assez bien dans la plupart des domaines scientifiques ou industriels mais cela semble n'avoir jamais empêché les guerres ni les échecs de projets.

Au printemps 2001, Thierry Cross, le fondateur d'XP France et moi-même présentions RAD et XP dans le grand « amphi » de l'UNIL à Lausanne. Les deux méthodes RAD et XP étaient présentées comme identiques et complémentaires. Pour étayer cette évidence, je modifiais « *on the fly* » ma conférence. Comme, à l'époque XP, ne disposait pas de références européennes à présenter, durant la période de question à Thierry ce furent celles du RAD à la SG et à la Seita, entre autres, qui furent « partagées ».

A plusieurs reprises il m'a été donné d'entendre des membres éminents du domaine Agile me dire off « Nous savons très bien que nous venons du RAD, mais personne ne souhaite se le rappeler. », ou bien encore « Pour introduire des nouvelles méthodes, il est indispensable de détruire les précédentes. »

En l'occurrence LA précédente c'est la méthode RAD, la PREMIERE méthode complète et publiée. Par le changement de paradigme qu'elle a apporté en formalisant, dans le détail, la mise en œuvre de « l'itératif, incrémental, adaptatif », elle constitue la racine indiscutable des méthodes Agiles suivantes.

En ce qui concerne *Scrum*, sans réfuter son incroyable impact médiatique lié au laxisme de ses certifications, l'étude comparée de son contenu par rapport à celui de RAD ne fait apparaître aucun nouvel apport à l'exception de la systématisation des rétrospectives. Par contre, *Scrum* ne dispose que d'une seule forme de planification et est handicapé par une absence totale de pratique de génie logiciel.

Une remise en cause édifiante

L'apport d'*eXtrem Programming* proposé en 1999 par Kent Beck dans le livre *XP explained* était la systématisation de certaines pratiques comme la programmation en binôme associé à la rotation des binômes, ces techniques facilitaient le TDD et la montée en compétence de l'équipe.

Ayant compris que le fossé du changement était loin d'être comblé, Kent Beck, très certainement sous la pression commerciale de *Scrum*, publia, pour tenter de faire adopter plus largement sa méthode, le livre *Planning XP*. Ce document remettait en question l'aspect extrême des pratiques XP.

L'ensemble des pratiques préconisées par XP préexistaient de manière indépendante et de mise en œuvre moins systématique en fonction du

contexte ou des ressources. En acceptant une nouvelle voie plus laxiste, c'est une très grande partie de l'apport de XP qui disparaissait.

Du même coup, ma proposition de « remplacer la phase de Construction du RAD par XP » n'avait plus vraiment de sens. Etrangement ces changements ne trouvèrent pas d'échos en France jusqu'en décembre 2008, lorsque Emmanuel Chenu³, publia un compte rendu de cet ouvrage sur son blog.

Ce livre a été conjointement écrit en 2001 par Kent Beck et Martin Fowler et préfacé par Tom DeMarco. « Cela fait trois grosses signatures pour un seul et même ouvrage. » J'ai immédiatement commandé le livre en question pour vérification, mais, par éthique, je préfère citer complètement Emmanuel Chenu :

« Les auteurs consacrent le livre à la planification et au pilotage d'un projet mené dans une démarche XP. Ils cherchent à montrer qu'en plus de pratiques techniques, XP contient les pratiques nécessaires et suffisantes pour planifier et piloter un projet.

On dit souvent que *Scrum* apporte aux pratiques techniques d'XP les pratiques de pilotage de projet et d'équipe. Ce livre tend à montrer que XP est déjà autosuffisant pour la gestion de projet. La surcouche *Scrum* ne fait que recouvrir avec son propre vocabulaire et ses rôles des pratiques préexistantes dans XP. Ceci dit *Scrum* apporte plus de formalisme et de cérémonial.

D'une manière générale, Martin et surtout Kent ont mis de l'eau dans leur vin et l'aspect extrême d'XP en est diminué. Déjà, en préface, les auteurs rappellent que XP n'est pas une destination mais un parcours. La durée de l'itération est devenue un sujet d'expérimentation. Le cycle hebdomadaire n'est plus une référence. L'essentiel est de rester en dessous du mois et de ne pas glisser !

De même, la pluridisciplinarité de chaque membre de l'équipe n'est plus encadrée par une rotation systématique des binômes sur toutes les parties de l'application⁴. La spécialisation des développeurs est même encouragée à partir du moment où elle est liée à une motivation personnelle.

³ <http://emmanuelchenu.blogspot.com>

⁴ Le *Pair Programming* généralisé comme outil de communication et d'apprentissage vient d'être tué.

Aussi, les auteurs sont plus permissifs quant à l'existence de bugs dans le produit. La correction de bugs mineurs n'est pas nécessairement une priorité pour l'équipe⁵.

Enfin, les auteurs sont moins impératifs sur la nécessité de développer les fonctionnalités de bout en bout. L'essentiel est désormais que tout ce qui est développé dans l'itération soit testé à la sortie de l'itération.

Personnellement je désapprouve le calcul d'une vélocité individuelle pour chaque développeur car je ne vois pas comment cela peut rester compatible avec un esprit d'équipe efficace⁶. Je préfère calculer la vélocité de l'équipe pour mesurer sa capacité de travail par itération et éventuellement diviser cette vélocité par le nombre de développeurs pour surveiller l'impact de l'intégration de nouveaux membres dans l'équipe.

Enfin, j'ai apprécié le désaccord entre Kent et Martin portant sur la priorité des fonctionnalités à développer. Kent accorde une plus grande priorité à celles qui apportent le plus de valeur au client alors que Martin accorde les plus hautes priorités à celles qui comportent le plus de risques. C'est le pilotage par retour sur investissement contre le pilotage par les risques. »

Voilà, merci monsieur Chenu, vous m'avez apporté beaucoup avec cette synthèse, car j'ai beau lire énormément sur le sujet, je n'avais pas étudié cet ouvrage, le titre m'ayant amené à penser que cela traitait uniquement de planification.

Ce qui me dérange, car je considère XP comme la méthode ayant apporté le plus aux années 2000, c'est l'abandon des aspects extrêmes. Si l'on tire un trait sur le *Pair programming* généralisé comme outil de communication et d'apprentissage ainsi que sur les aspects qualité (TDD et *Refactoring* généralisés), que va-t-il rester ? Le *Programming* ?

⁵ Cela ne remettrait pas en cause la généralisation du TDD et du Refactoring ces idées ?

⁶ De même, la méthode RAD préconisait les récompenses au niveau équipe et utilisait la métaphore du foot pour expliquer que même si l'un des joueurs était un dieu du stade, le match perdu l'était pour toute l'équipe.

Scrum, le fond des problèmes

Depuis la fin des années 80, j'ai intégralement consacré ma carrière et mes temps libres à la promotion d'une conduite de projet itérative incrémentale adaptative et en particulier à la première de ces méthodes : le RAD de James Martin.

Aujourd'hui, je suis totalement sidéré d'observer comment, des sectaires⁷, sans culture de leur métier pour certains et sans éthiques pour d'autres, tentent de s'affranchir de cette antériorité en dénigrant sans complexe ce qui est l'origine même des pratiques qu'ils idolâtrèrent.

Encore récemment, quelques fanatiques me reprochaient d'affirmer et de prouver que les méthodes Agiles actuelles sont incomplètes même en couplant plusieurs (généralement *Scrum* et XP) et que, de plus, leurs couvertures respectives se chevauchent. Pire, frisant « l'iconoclastie » je prétendais que mes clients se plaignaient de ne pas disposer de structuration Agile des exigences pas plus que de formes agiles de maîtrise des communications en environnement organisationnel complexe. Il faudrait suivre à tout jamais « *Scrum in the book* ».

Amusant cet aspect « ne touchez plus à rien » pour les jeunes défenseurs inconditionnels d'une Agilité qu'ils auraient souhaitée totalement révolutionnaire et totalement nouvelle. Il est vrai que cette virginité permettait de promettre une martingale incontournable du succès. La deuxième génération de méthodes itératives aurait eu comme avantage de ne rien devoir à personne et, du même coup, se serait pour un temps affranchie du passif de la pléthore d'échecs organisationnels aussi sûrement programmés que non évités dans la plupart des grandes organisations ayant refusé pour de multiples « bonnes » raisons la véritable évolution Agile. La suite se présente comme encore plus triste et pénalisante pour le monde Agile. Les critiques les plus dures viennent des US et elles sont cinglantes.

The Decline and Fall of Agile

Une des controverses majeures a pour origine l'article « *The Decline and Fall of Agile* » sur le blog de James Shore d'un expert indiscuté du domaine auteur de l'ouvrage *The Art of Agile Development*. Ce praticien confirme globalement :

⁷ "Agile is starting to sound a lot more like religious rhetoric than an engineering practice"

- Que prises indépendamment, les méthodes agiles actuelles sont incomplètes.
- Que *Scrum* est en train de faire déconsidérer globalement le mouvement Agile⁸.
- Qu'une approche semi-itérative (un retour au RAD quoi ...) qui permet de définir un minimum de besoins et d'architecture⁹ est mieux adaptée à la plupart des développements.

Il s'en est suivi un dialogue qui a permis à plus d'une centaine d'autres praticiens de venir exposer leurs problèmes. En résumé « L'engouement pour *Scrum* a fait que nombre d'équipes insuffisamment préparées se retrouvent dans des situations inextricables de dette technique. Ils ont naïvement cru que l'on pouvait tirer bénéfice d'une gestion de projet agile sans en payer le prix. »

Ceci ne m'a pas appris grand-chose puisque ce sont ces mêmes réflexions qui sont à la base de mes travaux sur PUMA. Mais sincèrement, je n'aurais pas pu imaginer mieux pour inciter certains fanatiques à se questionner un minimum.



Dans *Voyage in the Agile Memplex: in the world of agile development, context is key*, Philippe Kruchten s'inquiète lui aussi d'une « agilitation

⁸ "Rescuing Scrum teams keeps me in business"

⁹ "When in fact if you are truly agile, you will use waterfall (design up front) when necessary"

scrumiste » exagérée. Comme le précise Claude Aubry sur son blog : « Bref on le sent un peu remonté contre le dogmatisme des zéloteurs de l'agilité, les *agilistas*. » Philippe Kruchten craint le « retour de flamme » d'une agilité appliquée sans discernement et sans étude du contexte. Il stigmatise :

- le jargon utilisé, à travers les *Agilese* ;
- le suivisme aveugle, à travers l'*Agilism* ;
- l'intolérance des convertis, à travers l'*Agilitis* ;
- les études pseudo-scientifiques hors contextes d'*Agilologie*.

Selon Claude Aubry « A le lire, on a l'impression qu'aux US et au Canada, l'agilité est le courant dominant et qu'il est difficile d'y résister, voire de le critiquer : l'agilité serait la pensée unique. C'est loin d'être le cas en France. »

Martin Fowler dans un article de février 2009 en fait aussi état et présente le problème ainsi « Certains veulent une méthodologie Agile et choisissent Scrum. Ils adoptent les pratiques Scrum et même les principes. Au bout d'un certain temps l'avancement du projet ralenti car le code source est en pagaille du fait que Scrum omet délibérément les pratiques techniques » (détail sur le blog de Denis Dolfus).

Il n'est pas possible de citer tout le monde, mais il est évident que ces errements, qui causent un grand tort au mouvement Agile dans son ensemble, commencent à en énerver plus d'un et pas des moindres.

Note : tout cela me rappelle Merise et les certitudes inébranlables des « merisiens » avant qu'ils ne disparaissent sans laisser d'adresse.

Pour des outils adaptés à la culture européenne

Il faut que les experts Agiles français cessent d'être les simples perroquets de leurs maîtres américains en répétant inlassablement, persuadés d'avoir découvert le secret du laser à couper le beurre, l'appel aux trois ou quatre techniques qu'ils ont retenues « *Planning poker, Backlog, Burndown chart. Planning poker, Backlog, Burndown chart ...* »

Et, pourquoi pas : « Elle est fraîche ma méthode, elle est fraîche ... ».

D'autant que pour avoir pratiqué de nombreuses années des deux côtés de l'Atlantique, je peux vous assurer que rien ne s'approprie aussi aisément. D'ailleurs, mes visites dans de nombreuses DSI me laissent à penser qu'une jolie catastrophe, qui alimentera les futures statistiques d'échecs de projets du *Standish group* ou du *Gardner group*, est en train de se mettre plus ou moins agilement en place.

Les directions informatiques ne doivent pas être dupes et il serait logique de considérer *PUMA Essentiel* qui est la seule approche européenne et naturellement francophone. Simple et complète, elle permet d'obtenir une Agilité raisonnablement pensée et parfaitement en adéquation avec notre culture.

L'Agilité doit passer le cap du simple assemblage de techniques permettant l'encadrement d'un projet de développement et s'étendre à l'ensemble des préoccupations de l'organisation face à son système d'information, ses processus et ses ressources humaines.

De la méthode Agile, oui, mais la plus efficace. Surtout si elle est européenne et que nous pratiquons en Europe !

Mes premières expériences

L'aventure des développements *Agiles, Lean, Rapides*, etc. commence pour moi au Canada en 1988 avec la découverte des travaux de Boehm sur le développement itératif-incrémental. Dès 1989, après les premières communications de James Martin sur l'aspect adaptatif et le phasage court, je réalise à Hydro-Québec une application avec la méthode RAD, en utilisant le premier SGBDR sous Windows et en impliquant des utilisateurs enthousiastes. Le projet est une phénoménale démonstration d'efficacité mais me coûte mon job.

Je réitère la démonstration chez Bell Mobilité Cellulaire et ensuite pour les laboratoires Abbott. Devant les succès obtenus, je commence à publier sur le sujet. Je participe simultanément à plusieurs opérations *Lean*, d'amélioration continue et de réingénierie des processus avant automatisation. En 1991, James Martin présente le premier livre dédié au RAD.

Début 1994, de retour en France, je publie un rapport sur la méthode qui sera acheté par les grands comptes. Je lance alors le premier projet RAD à la Seita et consacre ensuite ma carrière et mon temps libre à promouvoir ce type de méthode. En 2000 je préconise d'intégrer XP à la Construction RAD. En 2001 je propose *PUMA*. Et voilà, c'était parti !

Bibliographie principale

- Ambler (S)*, Agile Modeling : Effective Practices, *Wiley*, 2002.
- Badot (O)*, Théorie de l'entreprise agile, *L'Harmattan*, 1998.
- Beck (K)*, Extreme Programming, *Campus Press*, 2002.
- Beck (K), Fowler (M.)*, Planning Extreme Programming, *A. Wesley*, 2001.
- Beck (K)*, Test Driven Development, *Pearson Education*, 2003.
- Cohn (M)*, User Stories Applied, *Addison-Wesley*, mars 2004.
- Debrauwer (L)*, Design Patterns, *Editions ENI*, 2007.
- Delbaldo (E)*, CMMi light, *Editions AFNOR*, 2008.
- Evans (E)*, Domain-Driven Design., *Addison Wesley*, 2002.
- Fowler (M)*, Patterns Enterprise Application Architecture, *A. Wesley*, 2002.
- Greenfield (J), Short (K), Cook (S), Kent (S)*, Software Factories, *Wiley*, 2004.
- Hammer (M), Champy (J)*, Le Reengineering, *Dunod*, 2000.
- Kerievsky (J)*, Refactoring to Patterns, *Addison Wesley*, 2004.
- Cross (T)*, Maîtriser les projets avec XP, *Cépaduès Éditions*, 2004.
- Cohn (M)*, Agile Estimating And Planning, *Prentice Hall*, 2004.
- Larman (C)*, Agile and Iterative Development, *A. Wesley*, 2004.
- Madoz (J-P)*, L'amélioration continue, *Afnor*, 2005.
- Martin (J.)*, Rapid Application Development, *Macmillan* 1991.
- Meszáros (G)*, Unit Test Patterns: Refactoring Test Code, . *Wesley*, 2007.
- Newkirk (JW), Vorontsov (A)*, Test-Driven Developmen, *Microsoft*, 2004.
- Petit-Etienne (M) - Peyraud (Y.)*, Reengineering, *Éd. d'organisation*, 1996.
- Schwaber (K)*, Agile Project Management with SCRUM, *Microsoft*, 2004.
- Stapleton (M)*, Dynamic Systems Development Method, *A. Wesley*, 1997.
- Vickoff (J-P.)*, RAD, 1994 MGI et 1996 MacMilan.
- Vickoff (J-P.)*, RAD CMM UML, *Gartner Group*, 1999.
- Vickoff (J-P)*, Systèmes d'Information et processus Agiles, *Hermès*, 2003.
- Williams (L), Kessler (R)*, Pair Programming Illuminated, *A. Wesley*, 1990

Liste des mots « clés »

C

Claude Aubri · 9

D

Denis Dolfus · 10
DSDM · 2

E

Emmanuel Chenu · 6
éthique · 3, 6

H

humaniste · 2

J

James Martin · 2, 3, 7
Jean-Pierre Vickoff · 2
Jeff Sutherland · 2
Jennifer Stapleton · 2

K

Ken Shwarber · 2
Kent Beck · 5, 6

M

Martin Fowler · 6

P

Philippe Kruchten · 9
PUMA Essentiel · 10

R

Ron Jeffrie · 2

S

Scrum · 2, 3, 4, 5, 6, 8, 9

V

vélocité · 6