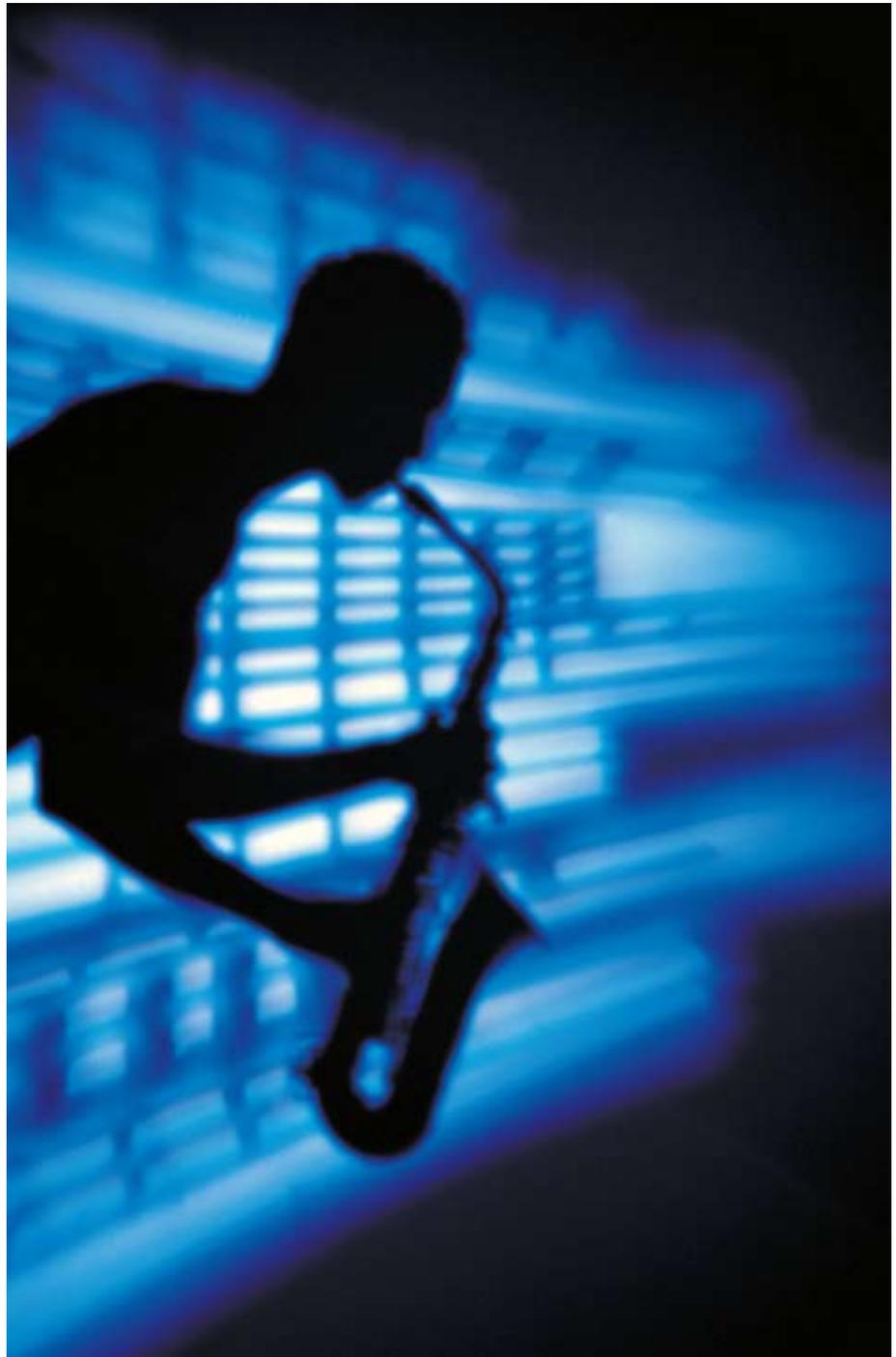




**Mai 2009**

|  |    |
|--|----|
| Les méthodes Agiles,<br>le pourquoi . . . . .  | 2  |
| L'état d'adoption des<br>méthodes Agiles. . . . .  | 4  |
| La valeur d'Agile pour une<br>organisation. . . . .  | 5  |
| Les challenges des projets<br>Agiles. . . . .  | 6  |
| Mettre en musique la<br>partition Agile : la plateforme<br>technologique Jazz. . . . .                 | 7  |
| Orchestration de la<br>collaboration des équipes<br>projet : IBM Rational<br>Team Concert. . . . .     | 8  |
| Définition de partitions :<br>IBM Rational Requirements<br>Composer. . . . .                           | 12 |
| Conclusion . . . . .   | 15 |
| Comparaison des méthodes<br>et options pour choisir<br>une méthode de<br>développement agile . . . . . | 16 |



## Optimisez vos investissements dans vos projets

# Les méthodes Agiles, le pourquoi

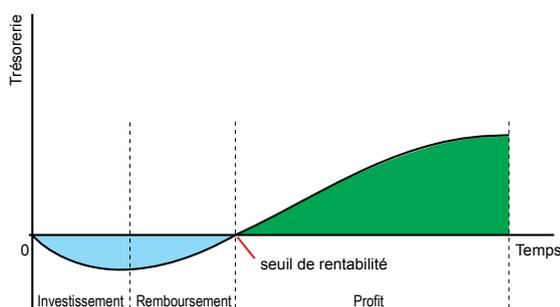
Des études<sup>1</sup> récentes montrent que seuls 34 % des projets réussissent à l'échelle mondiale, et près de la moitié n'arrivent pas à fournir toutes les fonctionnalités prévues au démarrage du projet.

Cette situation vient en grande partie du fait que les entreprises considèrent leurs organisations informatiques comme un centre de coûts plutôt qu'un centre générateur de valeur.

Pour mieux comprendre ce phénomène il faut d'abord comprendre les mécanismes financiers qui régissent la création du logiciel. Et pour cela, utilisons le modèle financier d'un projet mené avec succès et décrit par Mark Denne et Jane Cleland-Huang dans leur ouvrage « Software by the Numbers »<sup>2</sup>. Ce modèle a pour avantage d'être simple et compréhensible.

La création de tout produit ou service commence par une phase d'investissement. Une fois que le produit ou le service est disponible, la phase de retour sur investissement commence et se poursuit par une phase où ce produit génère un profit. Le logiciel n'est pas une exception. La Figure 1 présente les investissements et le retour sur investissements pour un logiciel dans le temps. Cette dépendance laisse penser que, s'il y a un moyen de raccourcir et de réduire la partie Investissement, il sera possible d'atteindre le seuil de rentabilité plus vite en profitant pleinement du logiciel produit.

Considérons que la période d'investissement est le temps nécessaire pour développer le logiciel final - ce qui reste une bonne approximation. Il est clair que,



Source: Mark Denne et Jane Cleland-Huang  
«Software by the Numbers»

**Figure 1. Projection de trésorerie d'un projet de développement mené avec succès**

pour arriver au seuil de rentabilité plus rapidement, il faut raccourcir la phase de développement.

Le montant d'investissement dépendra non seulement du temps passé dans le développement mais aussi de la complexité demandée au logiciel et de la charge de travail que cette complexité engendre.

Encore aujourd'hui, les logiciels sont souvent développés de telle manière que les utilisateurs ne voient que le résultat final ou, au mieux, reçoivent quelques livraisons de logiciels dites bêta avant la remise du logiciel au client. Cette situation signifie que les investissements sont réalisés en aveugle : il est bien possible qu'une fois livré le logiciel ne répondra pas aux attentes des utilisateurs ou pire que le logiciel ne sera jamais livré. Et cela constitue un risque majeur pour le retour sur investissement dans un tel projet.

Et si le logiciel était livré, montré et validé avec l'utilisateur, un nombre de fois important dès le début du projet et ce tout au long d'un projet de développement ?

Comment cela pourrait-il faire évoluer notre profil de trésorerie ?

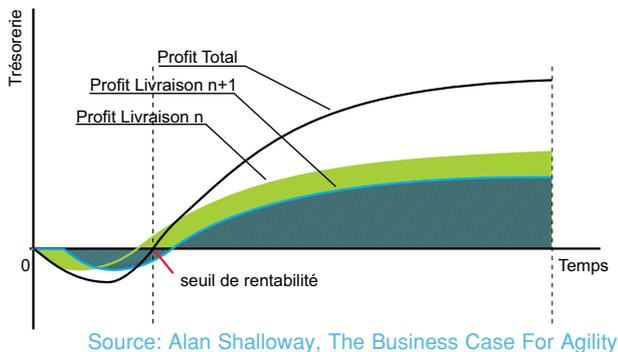
Lors d'un projet de développement d'un système basé sur le logiciel, chaque livraison d'un logiciel constituera un mini projet en soi, appelé itération, qui a pour but de réaliser une partie exécutable du système tout en intégrant les parties qui ont été développées lors des itérations antérieures. Cela n'est possible que quand ces mini projets se succèdent dans le temps. Par conséquent, le profil financier de deux itérations successives sera une somme de profils de chaque itération, Figure 2.

La Figure 2<sup>3</sup> met en évidence un constat très intéressant : le projet de développement par itérations (livraisons successives du logiciel) atteint le seuil de rentabilité plus rapidement qu'un projet classique, Figure 3. De plus, chaque livraison constitue un résultat tangible qui montre, de façon objective, l'état d'avancement du projet et qui, potentiellement, peut être utilisé par les utilisateurs. Par contraste, le projet classique n'est pas capable de donner la visibilité aux parties prenantes avant que le résultat final soit achevé. Cela augmente les risques d'échec d'un projet et peut provoquer éventuellement une perte des investissements.

<sup>1</sup>Standish Group, CHAOS Report 2004

<sup>2</sup>Mark Denne and Jane Cleland-Huang, Software by Numbers, Prentice Hall, 2004, ISBN 0-13-1407286-7

<sup>3</sup>Alan Shalloway, Net Objectives, The Business Case For Agility: The Lean Agile Connection, Agile 2007 Conference, [http://agile2007.agilealliance.org/agile2007/downloads/presentations/LeanAgileAgile2007\\_596.pdf](http://agile2007.agilealliance.org/agile2007/downloads/presentations/LeanAgileAgile2007_596.pdf)



Source: Alan Shalloway, The Business Case For Agility

**Figure 2. Projection de trésorerie pour deux itérations successives**

Il reste à trouver une réponse à la question suivante : comment réduire le montant absolu d'investissement lors d'un projet ? Certes, les règles de mathématique sont strictes, la somme de toutes les courbes qui représentent un profil financier sur chaque itération aura tendance à accroître les investissements. Par contre, l'application d'une règle 80/20 pour déterminer les vrais besoins vis-à-vis du système en développement permet de se concentrer sur l'essentiel du système et permet d'optimiser ainsi la charge de travail, de contenir les investissements et d'assurer la satisfaction de parties prenantes.

En résumé, l'atteinte rapide d'un seuil de rentabilité d'un projet nécessite :

- un nombre important de livraisons intermédiaires du logiciel qui doivent fonctionner tout au long de la durée de vie du projet ;
- une collaboration efficace afin de pouvoir bien hiérarchiser les besoins et valider le résultat ;
- une organisation adéquate qui permet l'adaptation rapide au changement dans le projet.

Que sont les méthodes Agiles et comment permettent-elles d'accélérer l'atteinte du seuil de rentabilité d'un projet de développement logiciel ?

Le développement de logiciel en utilisant les méthodes agiles (« Agile ») est une approche itérative et incrémentale menée de façon fortement collaborative avec le juste niveau de formalisme pour produire un logiciel de bonne qualité, dans le respect des délais et des coûts, en satisfaisant les besoins évolutifs de ses clients.

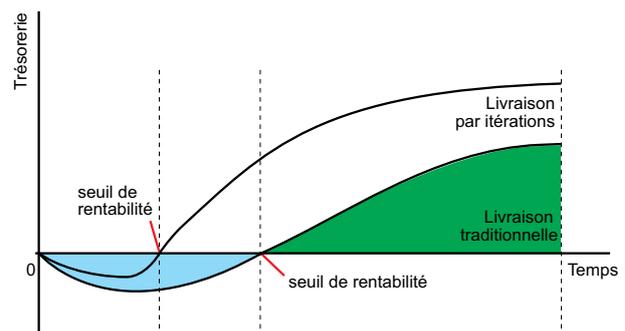
C'est cette nature itérative et collaborative d'Agile qui permet de réduire les risques liés au projet et d'atteindre le seuil de rentabilité plus rapidement.

Le développement Agile repose sur les principes suivants<sup>4</sup> :

- valoriser et motiver les individus au sein de l'équipe de développement ;
- adopter des processus adaptés « juste ce qu'il faut » ;
- mettre en œuvre une collaboration constante au sein de l'équipe ;
- impliquer en permanence les parties prenantes ;
- effectuer des livraisons fréquentes de logiciels qui fonctionnent ;
- offrir une réponse rapide aux changements ;
- effectuer des tests et validations en continu ;
- assurer transparence et visibilité.

Ces principes constituent une base philosophique et solide pour permettre le développement efficace de logiciels.

Les livraisons fréquentes permettent de valider le travail effectué ; la bureaucratie minimale renforce la collaboration et assure l'amélioration en continu.



Source: Alan Shalloway, The Business Case For Agility

**Figure 3. Projection de trésorerie pour un projet itératif mené avec succès**

Source: IBM

<sup>4</sup>Agile Manifesto, <http://agilemanifesto.org/>

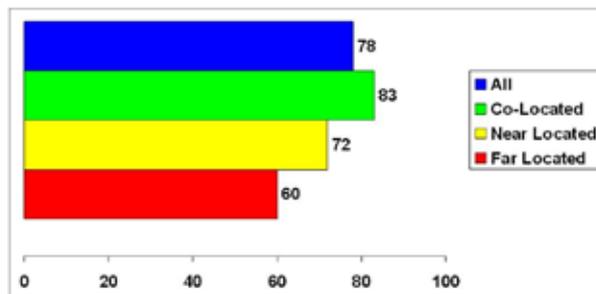
# L'état d'adoption des méthodes Agiles

Les méthodes Agiles présentent de plus en plus d'attraits pour les organisations qui s'occupent de livraison de logiciels. Les études faites par Dr. Dobb's Journal<sup>5,6</sup> en 2008 (642 réponses) révèlent que 69 % de répondants affirment que leur organisation a adopté une ou plusieurs techniques Agiles (Figure 4). 15 % de ceux qui n'ont pas adopté ces techniques Agiles indiquent vouloir les utiliser dans les six mois.

Ces résultats sont confirmés par le sondage intitulé « 3rd Annual Survey 2008: The State of Agile Development <sup>7</sup> », commandité par VersionOne Inc. et conduit en juillet-août 2008 sur la base de 3 061 répondants dans 80 pays. 5 % des répondants à ce sondage affirment que leur compagnie n'a jamais utilisé de méthodes Agiles contre 60 % qui les pratiquent depuis plus d'un an.

Alors pourquoi les organisations adoptent-elles les méthodes Agiles ? En fait, l'adoption d'Agile présente un risque très limité. Des études révèlent que le taux de réussite des projets Agiles est élevé. Sur plus de 500 projets étudiés par Dr. Dobb's Journal, le taux de réussite est supérieur à 50 %, ce qui constitue un très bon résultat pour l'industrie du logiciel (Figure 5). Ces chiffres sont confirmés par le sondage de VersionOne Inc. selon lequel 76 % de répondants affirment que plus des trois quarts des projets réussissent en utilisant les méthodes Agiles dans leur entreprise.

On reproche fréquemment à Agile de ne pas pouvoir l'utiliser à grande échelle, par exemple pour les projets distribués. Les résultats des études contestent ce point de vue. Selon Dr. Dobb's, le taux de réussite des projets distribués atteint 60 %, Figure 5. Ces



Source: Dr. Dobb's Journal

Figure 5. tenant compte de la répartition géographique

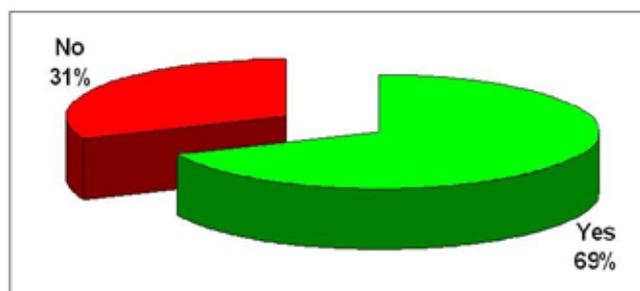
chiffres sont affirmés par l'étude « 3rd Annual Survey 2008: The State of Agile Development ». 49 % des répondants utilisent Agile sur deux sites et plus.

IBM rencontre une expérience largement positive de la conduite des projets Agiles distribués.

Par exemple, l'équipe IBM qui développe Rational Team Concert – environnement collaboratif de livraison de logiciels basé sur la plateforme technologique Jazz – est constituée de 70 développeurs et de 20 testeurs. Elle utilise les méthodes Agile - Eclipse Way et Agile Mash up et est répartie sur 8 sites : 3 au Canada, 3 aux États-Unis, 1 en France et 1 en Suisse<sup>8</sup>.

Pour conclure, en 2008, les organisations commencent à largement adopter les méthodes Agiles, qui présentent un risque relativement faible par rapport aux méthodes classiques.

Source: IBM



Source: Dr. Dobb's Journal

Figure 4. Adoption des techniques Agiles

<sup>5</sup>Scott W. Ambler, Ambysoft Inc., Agile Adoption Rate Survey, February 2008, <http://www.ambysoft.com/surveys/agileFebruary2008.html>

<sup>6</sup>Scott W. Ambler, Has Agile Peaked, Dr.Dobb's Journal, <http://www.ddj.com/architect/207600615?pgno=1>

<sup>7</sup>VersionOne Inc., 3rd Annual Survey: 2008 "The State of Agile Development" Full Data Report, [http://www.versionone.com/pdf/3rdAnnualStateOfAgile\\_FullDataReport.pdf](http://www.versionone.com/pdf/3rdAnnualStateOfAgile_FullDataReport.pdf)

<sup>8</sup>Erich Gamma, IBM, Using Team Concert in a Globally Distributed TeamBuilding Team Concert with Team Concert, Rational Software Development Conference 2008, Orlando USA

# La valeur d'Agile pour une organisation

Qu'apporte l'adoption de méthodes Agile aux organisations ? Est-ce que les méthodes Agile tiennent leurs promesses : atteindre plus vite le seuil de rentabilité en raccourcissant le temps de projets ?

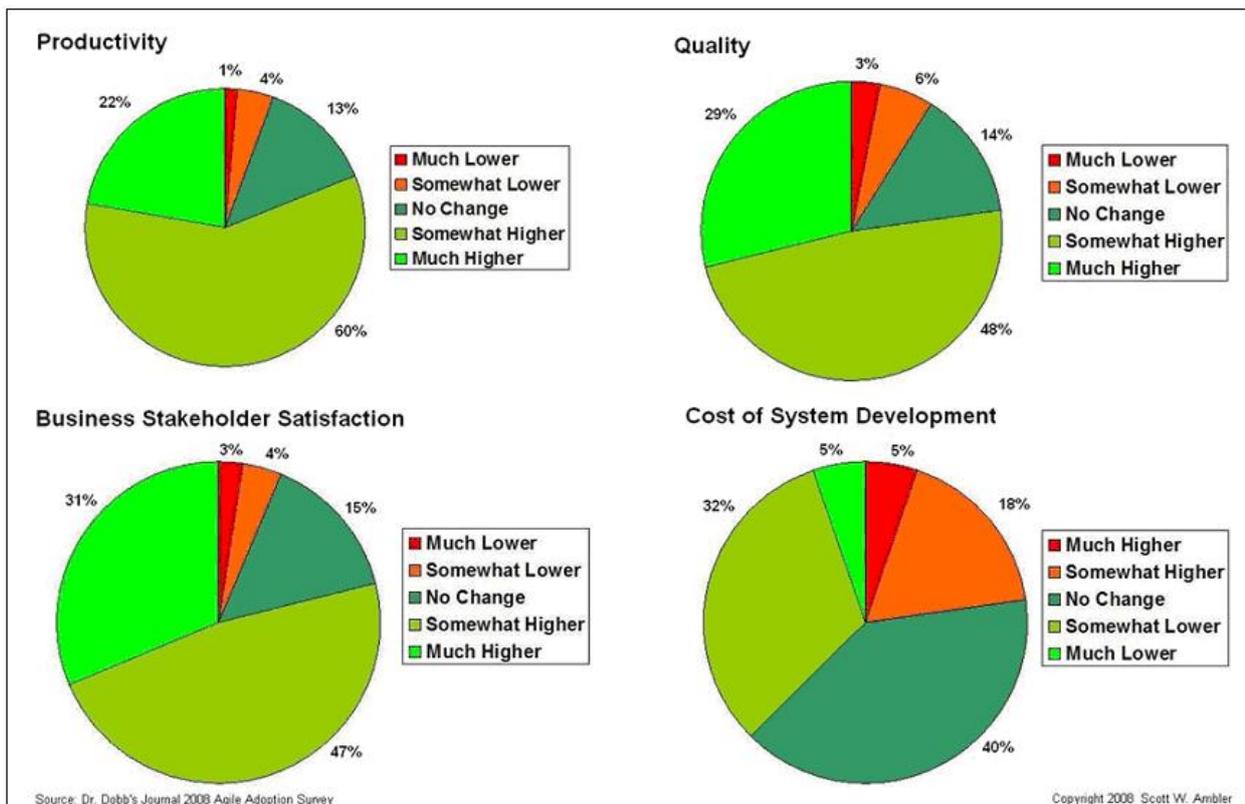
Dr. Dobb's Journal a évalué l'efficacité des équipes en pratique en évoquant les aspects de productivité, de qualité, de satisfaction des parties prenantes et de coût de développement (Figure 6). Les chiffres confirment l'hypothèse posée au début de cet article : les projets qui suivent les méthodes Agiles améliorent considérablement la satisfaction des parties prenantes et délivrent une meilleure productivité et une qualité supérieure.

L'étude<sup>5</sup> montre que les projets conduits avec les méthodes Agile permettent d'accélérer le temps de mise sur le marché (64,9 %) et, par conséquent, de diminuer le temps d'atteinte du seuil de rentabilité et de réduire les risques liés aux projets (64,6 %).

Pourquoi les projets Agiles obtiennent-ils d'aussi bonnes caractéristiques ?

Les itérations courtes permettent d'atteindre rapidement le résultat en termes de fonctionnalité exécutable. Grâce au caractère collaboratif des projets Agiles, les résultats délivrés sont validés en permanence avec les parties prenantes, ce qui permet de corriger tout de suite une éventuelle incompréhension. L'intégration continue assure la stabilité du système logiciel et les tests en continu la qualité. Lorsque le projet échoue, il échoue vite et n'occasionne donc pas d'investissements importants par rapport aux projets classiques, ce qui, en quelque sorte, constitue une réussite.

Source: IBM



Source: Dr. Dobb's Journal

Figure 6. Efficacité des équipes Agiles en pratique

# Les challenges des projets Agiles

Les méthodes Agiles promettent aux organisations qui les adoptent une grande efficacité et la réalisation plus rapide d'une valeur métier. Mais l'adoption des méthodes Agiles peut se heurter à des obstacles.

En 2008, SearchSoftwareQuality.com a effectué un sondage intitulé Agile Trends Survey 2008. Ce sondage a montré que les préoccupations majeures dans les projets Agiles sont<sup>9</sup>:

- communication (52 % de répondants),
- documentation (48 %),
- résistance aux changements (24 %),
- intégration des outils (24 %).

De plus, selon l'étude « 3rd Annual Survey 2008: The State of Agile Development », les autres obstacles qui impactent le succès de l'adoption des méthodes Agiles incluent :

- la capacité à changer la culture de l'organisation,
- le manque de personnel qui possède de l'expérience en développement Agile,
- la collaboration avec les parties prenantes,
- le soutien de la part du management.

Par la nature des projets Agiles, la collaboration et la communication sont des composants du succès des projets sur lesquels les équipes peuvent agir. Leur importance croît de façon exponentielle avec la réalisation de projets Agiles géographiquement distribués. Par ailleurs, Gartner, dans son étude « Agile Success Factors »<sup>10</sup>, recommande de mettre l'accent sur les techniques qui permettent d'assurer la collaboration au sein de l'équipe quand les projets prennent de l'importance en termes de taille et de distribution.

Idéalement, pour optimiser l'efficacité de l'équipe, la solution en termes de collaboration et de communication doit comprendre des capacités de planification des itérations et des capacités de gestion des tâches (avec un reporting non-intrusif et réalisé en temps réel par rapport à l'état du projet et aux activités de développement). Elle doit aussi offrir une infrastructure nécessaire au travail en équipe, comme la gestion de configurations et de construction de logiciel ainsi que la gestion des anomalies. De plus, cette solution doit être applicable aux équipes travaillant en un même lieu comme aux équipes distribuées afin d'optimiser les investissements sur une telle solution.

Un autre point important est de maintenir une collaboration et une communication efficaces avec les parties prenantes. En effet, l'étude « Agile Trends Survey » mentionne le processus de découverte et de collecte des exigences comme étant une difficulté majeure pour plus de 30 % de répondants<sup>11</sup>.

Très naturellement, les équipes projet et les représentants du client vont avoir tendance à communiquer avec des vocabulaires différents. Afin de réduire le syndrome : « oui, ce qui est livré est bien, *mais...* », caractéristique d'une compréhension incomplète des besoins des parties prenantes, il faut mettre en place un mécanisme qui permettra de capturer les besoins à travers des techniques visuelles et textuelles efficaces, sans contraindre la liberté de choisir celle qui est la plus adaptée. Ce mécanisme permettra aussi de partager la connaissance de ces besoins de façon instantanée avec tous les membres de l'équipe de développement.

Source: IBM

<sup>9</sup>Colleen Frye, SearchSoftwareQuality.com, Agile Trends 2008, Agile practitioners face challenges, but see process improvements, [http://searchsoftwarequality.techtarget.com/news/article/0,289142,sid92\\_gci1318820,00.html](http://searchsoftwarequality.techtarget.com/news/article/0,289142,sid92_gci1318820,00.html)

<sup>10</sup>David Norton, Matthew Hotle, Gartner, Agile Success Factors, 15 May 2007, G00148114

<sup>11</sup>Jack Vaughan, SearchSoftwareQuality.com, Agile Trends 2008, Teams turn to use cases, user stories to ease requirements gathering challenges, [http://searchsoftwarequality.techtarget.com/news/article/0,289142,sid92\\_gci1318606,00.html](http://searchsoftwarequality.techtarget.com/news/article/0,289142,sid92_gci1318606,00.html)

# Mettre en musique la partition Agile : la plateforme technologique Jazz

En reconnaissant leur valeur, IBM utilise les méthodes Agiles pour développer un grand nombre de ses propres produits comme IBM Lotus Sametime, IBM WebSphere Service Registry and Repository, IBM Rational Team Concert, IBM Rational AppScan et autres.

En 2008, IBM a annoncé la disponibilité de sa nouvelle plateforme technologique Jazz qui devient une fondation pour les outils de la marque IBM Rational de gestion du cycle de vie des applications collaboratives. Pourquoi une telle plateforme ?

IBM participe et contribue aux efforts communautaires de développement de logiciels libres comme Eclipse. Aujourd'hui, Eclipse est devenu, de facto, un environnement standard de développement Java. Parmi les facteurs de son succès, notons : intégration et test en continu, focus sur les points d'étape, planification incrémentale basée sur le retour fourni par la communauté et les exigences définies par le conseil, rétrospectives, processus « Endgame », etc. Chaque fabrication (build) d'Eclipse est considérée comme candidat à la livraison : logiciel qui doit marcher. Mais ce sont les propriétés d'utilisation des méthodes Agiles ! En effet, l'équipe IBM de développement d'Eclipse utilise une méthode Agile (EclipseWay) et l'équipe utilise Eclipse pour développer Eclipse.

Eclipse est focalisé sur la productivité d'un individu, pas sur celle de l'équipe... De plus, la méthode EclipseWay s'appuie sur la coopération des individus qui travaillent à l'unisson.

Les méthodes Agiles sont axées sur la collaboration des personnes au sein de l'équipe responsable d'un

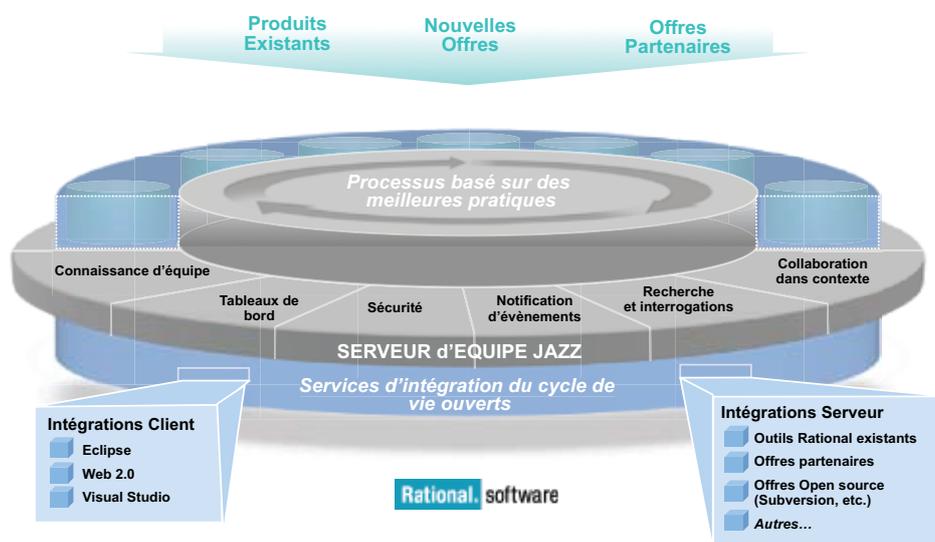
produit logiciel. L'efficacité d'une telle collaboration dépend de l'accessibilité instantanée à une information à jour liée à l'état d'avancement du projet. Cette information, correcte et objective, permet à son tour de prendre les bonnes décisions au bon moment. L'intégration étroite des mécanismes de gestion de fabrication, de gestion de configurations logicielles et de gestion des tâches ou des anomalies permet d'augmenter la productivité, la transparence et la prévisibilité de projets.

Malheureusement, les mécanismes comme les wikis et les outils disparates ne sont pas suffisants pour intégrer facilement un nouveau membre dans l'équipe, pour réaliser une collaboration efficace et pour atteindre les niveaux de productivité requis.

La résolution de ces problématiques constitue une des raisons de la création de la plateforme Jazz. Centrée sur la notion d'équipe, la plateforme Jazz permet de réunir les personnes, le processus nécessaire en fonction de l'étape du projet et les outils, d'effectuer le partage de l'information de façon fiable et sécurisée, de mettre un place le mécanisme de collaboration dans le contexte et de réaliser la collecte non intrusive de l'information liée à l'état du projet, dans un développement collaboratif de logiciel de haute qualité, Figure 7.

La plateforme Jazz est une infrastructure technique qui met à la disposition des services d'intégration des outils sur le cycle de vie du développement.

Source: IBM



Source: IBM

Figure 7. Plateforme Jazz

# Orchestration de la collaboration des équipes projet : IBM Rational Team Concert

IBM Rational Team Concert est une solution de gestion du cycle de vie des applications (Application Lifecycle Management) le premier de la ligne de produits basé sur la plateforme technologique Jazz.

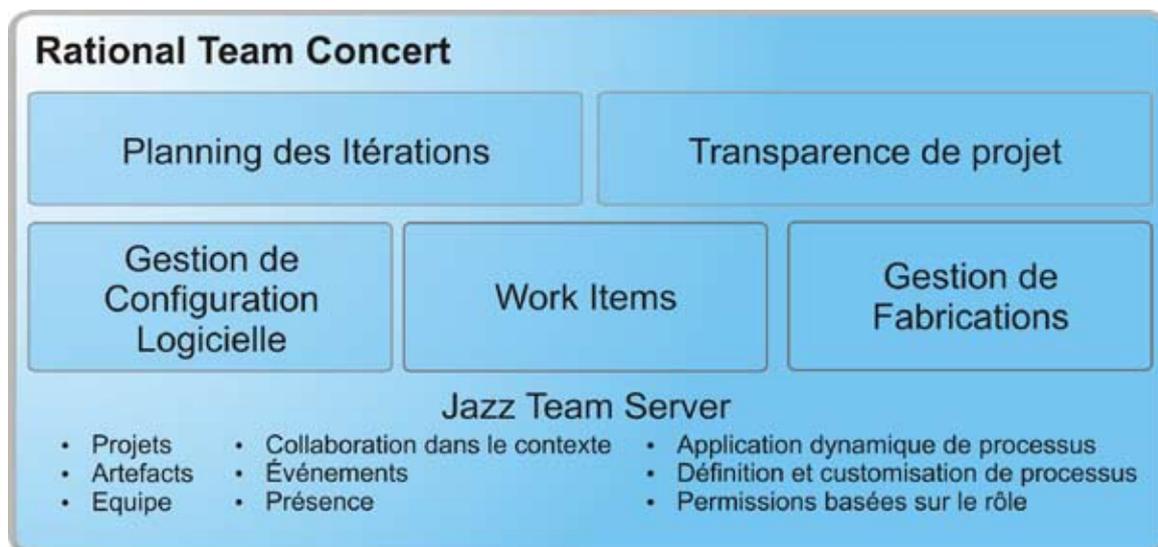
Il permet d'améliorer la collaboration au sein de l'équipe, d'encourager les meilleures pratiques du développement Agile, d'équilibrer la gouvernance et l'autorité et de réduire le coût total de possession.

L'environnement de développement collaboratif Rational Team Concert permet la planification et le suivi de projets de développement et inclut des fonctionnalités intégrées de gestion des configurations, de gestion des demandes de changement et de fabrications, et la collaboration dans le contexte, Figure 8. En parallèle, l'environnement offre la possibilité de personnaliser les processus et les rapports selon la méthode utilisée : EclipseWay, Scrum, XP...

Enfin, pour donner la liberté aux utilisateurs de travailler dans l'environnement de développement de leur choix, Rational Team Concert s'intègre avec Eclipse et Visual Studio .NET 2008.

Les tâches, un « Work Item » dans le vocabulaire de Rational Team Concert, sont une unité de planification et d'exécution de travail. Le Work Item couvre l'ensemble des activités projet : le travail planifié, les demandes de changement, de corrections, d'amélioration, etc. Ils sont liés aux modifications effectuées dans le référentiel de projet et aux fabrications réalisées. Les Work Items permettent de suivre l'exécution de tâches, de connaître le travail réalisé et celui planifié pour les itérations ultérieures et de garder ces traces, Figure 9. Le statut et le nombre de Work Items peuvent être utilisés comme des indicateurs de santé du projet.

Pour collaborer efficacement au sein de l'équipe, Rational Team Concert utilise des fonctionnalités de messagerie instantanée qui permettent de véhiculer des informations et des discussions liées au projet entre les membres de l'équipe, à travers la plateforme Jazz. Chaque élément d'information est géré par Rational Team Concert - un plan d'itération, un Work Item, un élément de référentiel, etc. - et peut être partagé entre les membres de l'équipe. Il devient accessible instantanément de telle manière que le



Source: IBM

Figure 8. IBM Rational Team Concert



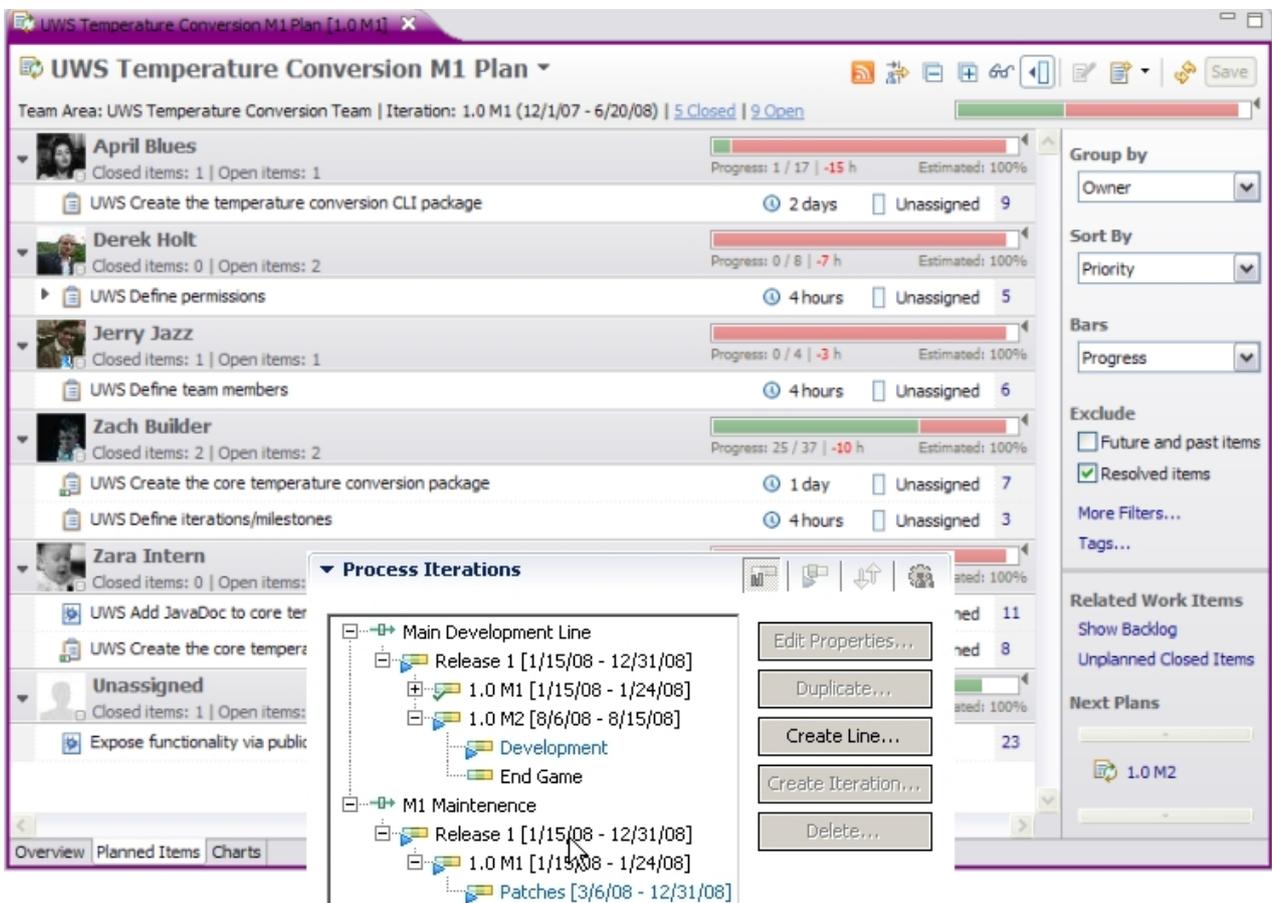
destinataire est capable d'appréhender sans effort ce que l'expéditeur voulait exprimer dans le contexte de son travail, Figure 10.

Afin d'atteindre une meilleure transparence et une meilleure visibilité de projets, Rational Team Concert propose un client Web riche basé sur la technologie Web 2.0. La plupart des fonctionnalités de Team Concert sont accessibles pour les utilisateurs au travers du Web.

IBM Rational Team Concert collecte de l'information de façon non intrusive et la met à disposition pour construire des rapports et des tableaux de bord personnalisés, accessibles au travers du Web. Les rapports et les tableaux de bord permettent d'évaluer la santé des projets et de prendre des décisions à partir d'informations objectives, Figure 11. Bien entendu, l'information publiée dans les rapports et sur les tableaux de bord est mise à jour en temps réel dans le client Web riche, sans effort supplémentaire pour l'équipe.

IBM Rational Team Concert met à disposition l'infrastructure nécessaire à l'équipe de projet : les modules de gestion de configuration et de fabrication de logiciel comprennent tous les composants nécessaires qui garantissent la pérennité de la propriété intellectuelle créée ou modifiée lors du projet, sa gestion ainsi que la fiabilité de fabrication. IBM Rational Team Concert propose à ses utilisateurs :

- la gestion du développement et du travail en parallèle,
- la gestion des environnements et des versions de logiciel,
- l'automatisation de la fabrication entre les environnements,
- les reports de corrections et les arbitrages des conflits de fusion.



Source: IBM

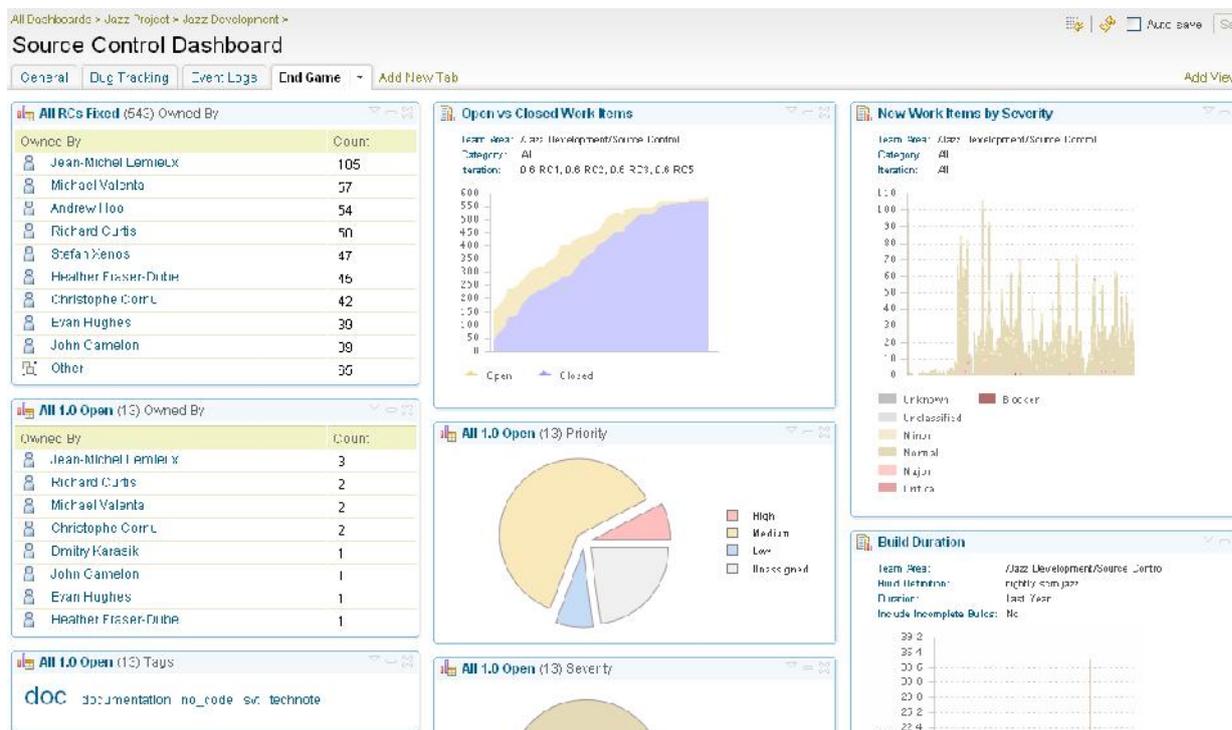
Figure 9. Planning des itérations

# Orchestration de la collaboration des équipes projet : IBM Rational Team Concert



Source: IBM

Figure 10. Collaboration dans le contexte



Source: IBM

Figure 11. Tableau de bord



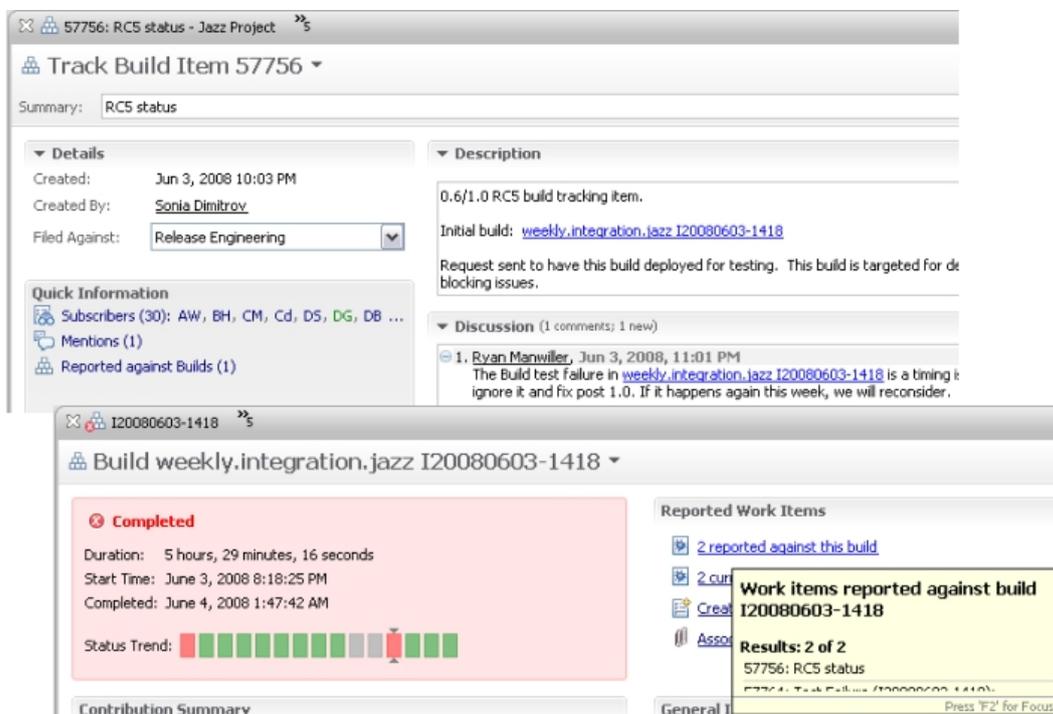
Concernant la gestion de la fabrication, Figure 12, Rational Team Concert comporte un module de construction et d'intégration continue avec pour objectif de permettre aux équipes de développement d'adopter des méthodes Agiles, pour les projets nécessitant des livrables fréquents. IBM Rational Team Concert capture l'ensemble des résultats d'une fabrication afin d'en garder l'historique, qui peut être utilisé pour une analyse ou une comparaison.

Des systèmes d'alertes et de notification sont également disponibles pour prévenir les utilisateurs, par exemple, de la disponibilité d'une nouvelle fabrication à récupérer.

Pour quantifier les bénéfices potentiels suite à l'utilisation d'IBM Rational Team Concert, IBM a demandé à Alinean, Inc., une compagnie indépendante spécialiste dans le domaine du calcul de ROI, d'effectuer une analyse de retour sur l'investissement lié à son utilisation par des équipes de développement. Le calculateur de retour sur

investissement utilise les métriques financières et les modèles de coûts définis selon l'industrie. Les métriques et le modèle sont développés par Alinean de façon indépendante. Selon Alinean<sup>12</sup>, Rational Team Concert permet potentiellement :

- d'améliorer le taux d'utilisation de l'équipe de 15 % à 50 % grâce à des rapports en temps réel, via une meilleure visibilité sur le projet et via la suppression de barrières de communication à travers l'équipe ;
- d'augmenter la qualité du logiciel d'environ 30 % grâce à la promotion de meilleures pratiques et d'une meilleure connaissance du processus par IBM Rational Team Concert ;
- d'améliorer de 15 % la productivité liée à la gestion de projet. Cela comprend la gestion du projet, la coordination des ressources, la maintenance des rapports, la gestion de la planification ;



Source: IBM

Figure 12. Gestion de la fabrication

<sup>12</sup>Alinean Inc., Business Value Analyst for Rational Team Concert (RTC) Product Savings Document, IBM, Alinean Inc., 2008

# Orchestration de la collaboration des équipes projet : IBM Rational Team Concert

- de diminuer de 40 % les coûts de la collaboration entre les membres distants de l'équipe ;
- de réduire d'environ 11 % le nombre de demandes de changements grâce à une meilleure visibilité sur le projet, une collaboration et une communication améliorées ;
- de diminuer d'environ 15 % les problèmes liés à la fabrication.
- efficacité accrue des équipes de développement distribuées grâce à un accès rapide et fiable aux artefacts de développement, quelle que soit leur localisation ;
- diminution du coût de possession du logiciel.

Les sondages menés par IBM auprès de ses clients confirme les bénéfices liés à l'utilisation de IBM Rational Team Concert.

Par exemple, dans les deux semaines qui ont suivi l'implémentation de IBM Rational Team Concert, en plus de l'augmentation de la productivité immédiate et d'une meilleure collaboration, le fournisseur de services métiers et informatiques au Canada, CGI Group, a constaté une amélioration instantanée de sa capacité à prévoir l'issue des projets en maintenant la visibilité et la transparence des projets, sans effort.

L'expérience des équipes de développement d'un produit IBM, qui utilisent des méthodes Agiles – IBM Rational AppScan – est un autre exemple démontrant que l'utilisation de IBM Rational Team Concert dans l'environnement Microsoft Visual Studio .NET 2008 apporte des bénéfices tels que :

- meilleur support de développement en parallèle et meilleur support de fabrication simultanée des différentes livraisons ;

Mais l'exemple le plus important est peut-être celui de l'équipe de développement de IBM Rational Team Concert, qui est répartie sur huit sites dans le monde, constituée d'environ 100 membres et qui utilise la méthode EclipseWay et IBM Rational Team Concert depuis fin 2006 pour développer Jazz et IBM Rational Team Concert. Aujourd'hui, ce projet Agile est constitué d'environ 66 000 fichiers avec un volume total du code de 10 Go, de plus de 40 000 Work Items, de 40 fabrications par jour et d'une fabrication d'intégration par semaine.

IBM a mis à disposition sur son site Web <http://www-01.ibm.com/software/rational/rtc/roi/> un calculateur de retour sur investissement simplifié qui permet d'évaluer les bénéfices d'utilisation de IBM Rational Team Concert dans une entreprise. Pour une étude approfondie et personnalisée de retour sur investissement sur l'utilisation d'IBM Rational Team Concert, veuillez-vous adresser à votre représentant commercial IBM Rational.

En résumé, la solution collaborative IBM Rational Team Concert propose à ses utilisateurs l'ensemble des outils nécessaires à une collaboration efficace et permet de mener les projets Agiles avec succès.

Source: IBM

## Définition de partitions : IBM Rational Requirements Composer

Le modèle financier d'un projet de développement de logiciel décrit au début de cet article suggère que les montants d'investissement soient liés au périmètre que le logiciel doit réaliser afin de répondre aux besoins exprimés par le métier. Une des questions qui se pose à l'équipe de développement est de savoir comment concentrer les efforts sur les besoins essentiels qui permettront de garantir la satisfaction des parties prenantes à l'issue du projet. Une des suggestions proposées par cet article est d'utiliser la règle 80/20.

Il existe un certain nombre de techniques très efficaces qui permettent de retrouver les vraies exigences pour un logiciel en développement. Malheureusement, jusqu'à aujourd'hui, il n'existait aucun moyen de transcrire cette bonne compréhension des exigences et de les partager à la fois avec les parties prenantes et l'équipe de projet. Cela est un obstacle majeur dans la conduite des projets Agiles. Par ailleurs, le rapport « Agile Trends 2008 » mentionne que la partie du cycle de vie des applications qui pose le plus de problèmes est le processus de collecte des exigences (30,7 % de répondants).

# Définition de partitions : IBM Rational Requirements Composer

Pour remédier à ce problème, IBM a annoncé en 2008 la disponibilité d'un outil basé sur la plateforme Jazz – IBM Rational Requirements Composer, Figure 13 – qui a pour but d'aider les équipes de développement à surmonter les challenges de définition des exigences.

Au sein du cycle de vie des exigences, IBM Rational Requirements Composer (Figure 13) favorise une collaboration en temps réel entre les parties prenantes et l'équipe de projet en fournissant une variété de techniques visuelles et textuelles pour collecter, analyser et élaborer les besoins : les glossaires métiers (Figure 14), les objectifs métiers, les processus métiers (Figure 15) et également les diagrammes de cas d'utilisation (Figure 16) ou aussi les possibilités d'écrire l'interface des utilisateurs cible sous forme de scénario et de maquette (Figure 17) et de composer le tout au sein de documents.

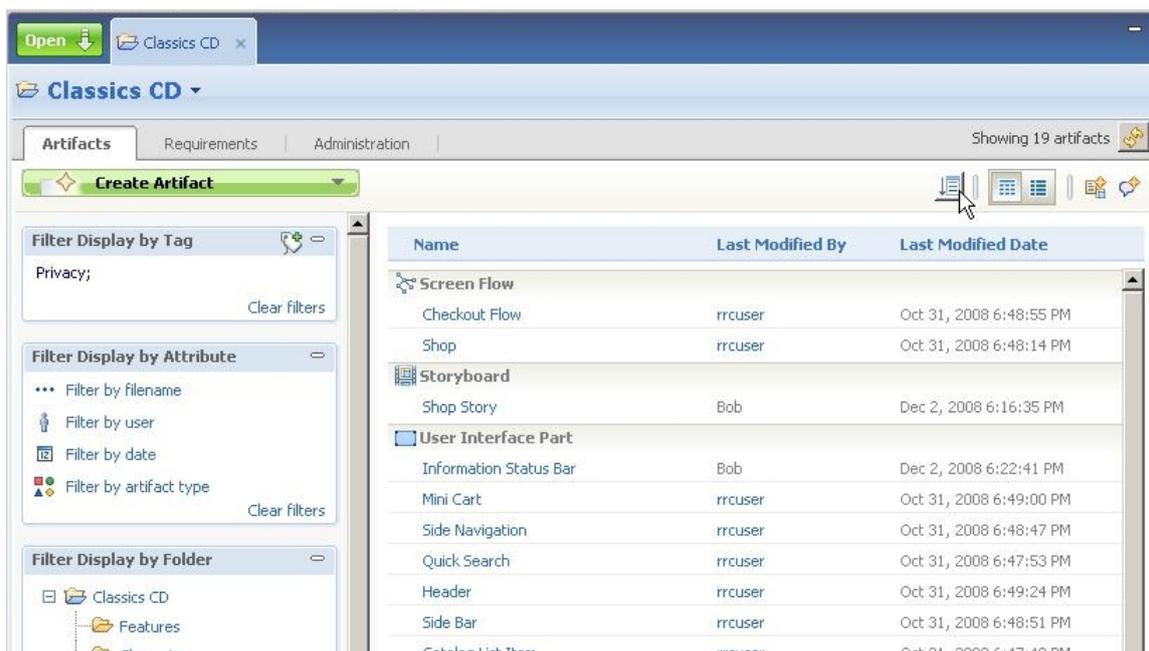
Bien entendu, l'équipe de développement va être amenée à toujours utiliser les techniques de découverte des exigences ; les capacités riches et graphiques d'IBM Rational Requirements Composer permettront de capturer ces exigences et de partager efficacement la connaissance acquise du système en développement dans l'équipe ainsi qu'avec les parties prenantes.

Une fois les exigences identifiées, IBM Rational Requirements Composer permettra de rentrer dans le cycle de leur gestion en s'interfaçant avec les outils de gestion des exigences comme IBM Rational RequisitePro.

Ce qui peut fonctionner pour de petits projets d'une équipe partageant le même bureau ne fonctionne plus pour des projets plus importants, complexes ou distribués. Grâce à son socle Jazz, IBM Rational Requirements Composer permet de passer :

- de situations initiales où les informations sur les besoins peuvent se trouver éparpillées et donc non accessibles à tous (à cause de référentiels, mails, formats, présentations, documents, feuilles de calcul différents)
- à un réseau d'information intégré, permettant des collaborations en temps réel entre l'ensemble des parties prenantes tout en conservant les liens et la traçabilité entre les différentes informations.

La collecte et la définition efficace des exigences avec IBM Rational Requirements Composer permettent aux équipes de se concentrer sur ce qui est réellement important pour les parties prenantes au niveau du système en développement en appliquant réellement la règle 80/20 afin d'obtenir une meilleure satisfaction des parties prenantes.



Source: IBM

Figure 13. IBM Rational Requirements Composer

# Définition de partitions : IBM Rational Requirements Composer



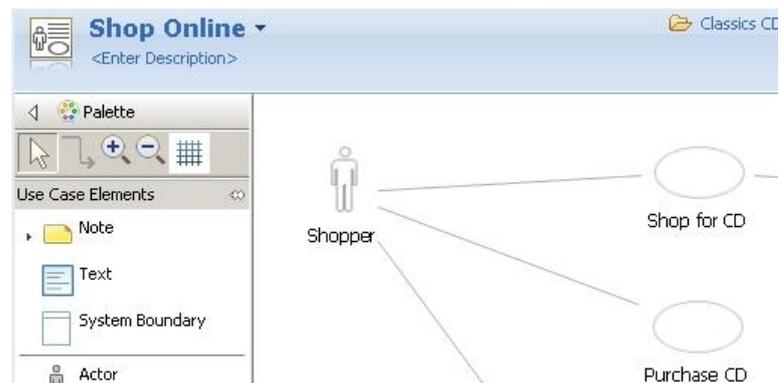
Source: IBM

Figure 14. Définition de glossaire métier



Source: IBM

Figure 15. Capture de processus métier



Source: IBM

Figure 16. Diagramme de cas d'utilisation

# Définition de partitions : IBM Rational Requirements Composer



Figure 17. Scénarios et maquette de l'interface utilisateur

## Conclusion

Les méthodes Agiles se sont affirmées dans l'industrie comme des méthodes de développement qui réduisent les risques, produisent le résultat escompté et permettent de livrer des systèmes logiciels de qualité dans les meilleurs délais. Economiquement, les méthodes Agiles sont plus efficaces en termes de rentabilité et de retour sur investissement.

Néanmoins, il reste certains challenges qui doivent être résolus pour rendre les équipes Agiles plus performantes et efficaces et pour augmenter leur taux de réussite et confirmer les méthodes Agiles comme les méthodes dominantes du développement de logiciel.

IBM Rational, grâce à ses solutions comme Rational Team Concert et Rational Requirements Composer basées sur la plateforme technologique Jazz, permet d'automatiser le développement Agile, de le rendre plus transparent, plus efficace et encore plus prévisible, d'augmenter le taux de réussite des projets et d'aider l'équipe à se concentrer sur les sujets vraiment importants pour les parties prenantes.

Michel Speranski  
Rational Market Manager  
IBM France

© Copyright IBM Corporation 2009  
Tous droits réservés

Rational Team Concert, Rational Requirements Composer, IBM, le logo IBM et Rational sont des marques déposées d'International Business Machines Corporation aux Etats-Unis et/ou dans certains autres pays.

Les autres noms de sociétés, de produits ou de services peuvent appartenir à des tiers.

Les informations contenues dans la présente documentation sont fournies à des fins d'information uniquement. Même si tout a été mis en œuvre pour vérifier l'intégrité et l'exactitude des informations contenues dans la présente documentation, ces dernières sont fournies "en l'état", sans aucune garantie, explicite ou implicite. De plus, ces informations sont basées sur les plans et la stratégie de produits actuels d'IBM, lesquels sont sujets à modification par IBM sans préavis. IBM ne peut être tenu pour responsable de tout dommage émanant de l'utilisation de, ou sinon associée à la présente documentation ou toute autre documentation. Aucun élément présent dans cette documentation n'a pour objet, ni n'aura pour effet, de créer une quelconque garantie ou représentation de la part d'IBM (ou de ses fournisseurs ou concédants de licence) ou de modifier les conditions du contrat de licence en vigueur régissant l'utilisation des logiciels IBM.

# Comparaison des méthodes et options pour choisir une méthode de développement agile

Cette étude présente une comparaison des méthodes actuelles de développement agile et *lean* les plus appréciées. Ces informations ont pour objectif de permettre aux dirigeants de prendre une décision éclairée et d'avancer dans leur choix.

## Principales conclusions

- Le bénéfice des pratiques agiles et lean pour la productivité, la qualité et la flexibilité a été démontré de façon empirique ces cinq dernières années.
- Il n'existe pas une seule méthode agile, mais un ensemble de méthodes ayant en commun la collaboration, de petites équipes et la primauté des personnes sur les outils et les processus.
- Les méthodes agiles diffèrent selon la couverture et les pratiques en phase de développement. Certaines méthodes sont axées sur la gestion, d'autres sur le développeur, d'autres enfin sur les deux.

## Recommandations

- Bien comprendre les valeurs, les principes et les pratiques de chaque méthode et ce que chacune peut apporter pour optimiser la productivité, la qualité et la réactivité.
- Être rigoureux lors de l'évaluation et de l'adoption des méthodes agiles, comme pour toute nouvelle technologie.
- Adopter un développement agile dans le cadre d'une initiative diversifiée pour intégrer les valeurs et les principes agiles dans les sociétés de services informatiques et l'activité. L'adoption franche de processus agiles implique des changements très importants au niveau des développeurs et de la gestion.

## ANALYSE

### CE QUE VOUS DEVEZ SAVOIR

- Les entreprises qui comptent adopter des méthodes agiles et lean ont le choix parmi un grand nombre de méthodes. Le secret d'une adoption réussie réside dans la connaissance des méthodes, de leur couverture, de leur impact sur l'entreprise, de leurs risques et de leurs avantages.
- Vu la publicité qui entoure les méthodes agiles, un échec peut très vite arriver sur la place publique. Les responsables du développement doivent peser les chances de réussite et les risques liés à l'adoption d'une méthode agile et ne doivent pas se précipiter pour choisir.

- L'adoption d'une méthode agile doit être analysée dans le cadre plus large de l'introduction des pratiques agiles dans l'entreprise. Il s'agit là d'un changement stratégique et non tactique.

## Méthodes et couverture

Il n'existe pas une seule méthode agile, mais un ensemble de méthodes (au moins 14 sont largement reconnues) avec des valeurs et des principes communs. Les thèmes communs des méthodes agiles sont la collaboration, des équipes dotées des moyens nécessaires, des clients actifs et des cycles de mise en production courts.

Cette étude porte essentiellement sur les méthodes Extreme Programming (XP), Scrum, Dynamic System Development Method (DSDM), FDD (Feature-Driven Development) et Lean Software Development. Il s'agit des méthodes agiles les plus appréciées et les plus largement adoptées actuellement. Microsoft Solutions Framework (MSF), Rational Unified Process (RUP) d'IBM et les dérivés agiles tels que Essential Unified Process et Agile Unified Process feront l'objet d'une autre étude.

- **Scrum** : terme emprunté au rugby, Scrum désigne une méthode de développement agile axée sur la planification et l'organisation des projets. Elle vient en complément des méthodes de développement agiles axées sur les disciplines du développeur dans le contexte de la gestion Scrum. Scrum permet d'organiser les équipes afin de les rendre plus productives et de leur donner les moyens de produire des logiciels de meilleure qualité avec une valeur métier. Cette approche permet aux équipes de réagir rapidement aux nouveaux besoins des clients, car le développement se fait généralement en cycles de 30 jours ou moins.
- **XP** : il s'agit d'une méthode de développement de logiciel lean et agile qui repose sur un ensemble de valeurs, de principes et de pratiques. En combinant la collaboration étroite des équipes et l'enveloppement direct, XP vise à optimiser la productivité des développeurs en se concentrant sur le code. XP accorde plus d'importance aux personnes qu'aux processus et à la technologie, mais encourage tout de même la discipline de projet. Étant donné que XP privilégie les

# Comparaison des méthodes et options pour choisir une méthode de développement agile

développeurs, elle doit être prise en charge par une structure de gestion de projets efficace. XP et Scrum sont couramment associées.

- **DSDM** : cette méthode a été développée en Grande-Bretagne au milieu des années 1990. Elle est contrôlée par le consortium à but non lucratif DSDM et en est à la version 4.2. DSDM s'étend de l'autorisation à la maintenance après l'analyse de rentabilité, ce qui lui donne la plus grande portée de toutes les méthodes agiles. Comme Scrum, DSDM est axée sur les aspects liés à la gestion de projet de la livraison itérative de produits. Bien que donnant des conseils sur le test, le développement, la modélisation et le prototypage, elle a également la meilleure pratique d'utilisation de DSDM avec XP, là où XP fournit l'agilité au niveau du développeur.
- **FDD** : cette méthode est tournée vers la partie normative de l'échelle de méthode agile, avec XP à l'autre bout. FDD réutilise des éléments des méthodes itératives plus formelles. Elle adopte une approche verticale de l'analyse et de la planification et privilégie la modélisation comme activité déterminante. Cette méthode est associée aux principales pratiques agiles telles que de petites équipes et des cycles de livraison efficaces pour le client. En partageant l'espace des problèmes et des solutions par le biais de cas d'utilisation et de modèles, FDD peut augmenter ou diminuer la taille des équipes selon les besoins. FDD est particulièrement appréciée pour les projets de grande envergure avec des équipes d'au moins 50 personnes.
- **Lean** : cette stratégie de développement de logiciel s'appuie sur les principes lean (littéralement « allégé ») expérimentés dans le secteur de la fabrication. Semblable aux

techniques agiles, elle est axée sur les personnes, la constitution de savoir et d'expertise. Elle porte tous ses efforts sur l'ajout de valeur métier et sur la suppression des gaspillages du processus de développement, ce qui l'allège. Lean n'est pas une méthode, mais une structure de gestion et d'optimisation. Elle n'est donc pas entourée d'objets, de rôles ni d'activités. Les principes lean peuvent être appliqués au développement, mais également au fonctionnement, au support technique, à l'architecture ou à tout rôle.

Le Tableau 1 présente les méthodes agiles les plus appréciées et les plus largement adoptées.

## Comparaison des méthodes

Il existe peu de données empiriques fiables concernant les méthodes agiles. Pour la plupart, il s'agit de données non corroborées ou produites par des personnes et des organisations ayant directement intérêt à favoriser ou à critiquer les méthodes agiles. Les sources des données du Tableau 2 sont variées et proviennent notamment d'études de cas, de communautés agiles et d'échanges entre clients. Les sources clairement partiales ont été supprimées de même que les observations statistiques extrêmes, c'est-à-dire les valeurs qui ne peuvent pas être répétées dans la plupart des organisations.

La taille des projets agiles varie avec chaque méthode. XP utilise des histoires utilisateur (user stories), Scrum utilise des index de produits, FDD des fonctions et DSDM une ligne de code ou des points de fonction. Lean référence le stock, mais laisse la mesure quantitative indéterminée.

Nous pouvons donc comparer les méthodes de projets dont les tailles ont été normalisées par l'utilisation de points de fonction. Les points de fonction (PF) sont présentés ci-après par gammes de projets classiques et classés par tailles de projet (petit, moyen, grand, etc.) :

Tableau 1. Couverture de la méthode par phase

| Catégorie  | Scrum | XP | DSDM | FDD | Lean |
|--|-------|----|------|-----|------|
| Pratiques de gestion   | S     | O  | S    | G   | S    |
| Pratiques du développeur   | O     | S  | G    | S   | G    |
| Analyse préprojet  | O     | O  | S    | O   | O    |
| Pratiques de modélisation  | O     | O  | S    | S   | O    |
| Pratiques de maintenance   | G     | G  | G    | O   | G    |
| S = Champ d'application, O = Hors champ et G = Conseil d'ordre général |       |    |      |     |      |

Source : Gartner (octobre 2007)

# Comparaison des méthodes et options pour choisir une méthode de développement agile

Tableau 2. Comparaison des méthodes agiles

| Catégorie                                       | Scrum                     | XP                        | DSDM                      | FDD                        | Lean              |
|---|---------------------------|---------------------------|---------------------------|----------------------------|-------------------|
| Taille habituelle d'un projet [voir Remarque 1] | 10 à 2 000 PF             | 10 à 800 PF               | 250 à 1 000 PF            | 500 à 7 000 PF             | Peu importe       |
| Taille de l'équipe                              | 5 à 12                    | 2 à 12                    | 2 à 6                     | 2 à 8                      | 6 à 12            |
| Évolutivité du projet                           | Plusieurs équipes (2 à 6) | Plusieurs équipes (2 à 4) | Plusieurs équipes (2 à 5) | Plusieurs équipes (2 à 10) | Plusieurs équipes |
| Productivité du développeur                     | 30 à 40 PF/SM             | 30 à 40 PF/SM             | 15 à 20 PF/SM             | 12 à 15 PF/SM              | 8 à 20 PF/SM      |
| Part de marché agile [voir Remarque 2]          | 20 à 25 %                 | 30 à 35 %                 | 3 à 8 %                   | 5 à 10 %                   | 5 à 10 %          |
| Risque de l'adoption                            | Modéré                    | Élevé                     | Modéré                    | Élevé                      | Modéré            |

Source : Gartner (octobre 2007)

- Petite amélioration : 1 à 50 PF ;
- Nouveau développement, petit : 51 à 500 PF ;
- Moyen : 501 à 3 000 PF ;
- Grand : 3 001 à 7 000 PF ;
- Très grand : 7 001 à 15 000 PF ;
- Extra : 15 001 PF et plus.

La productivité soulève le même problème de normalisation pour la comparaison des méthodes. XP utilise la vélocité du projet, Scrum résout les problèmes de produit et de backlog et FDD utilise la fonction par itération. Lean recommande la productivité en termes d'activité (pourcentage d'analyses de rentabilité mises en œuvre). L'utilisation de points de fonction permet de normaliser la productivité des points de fonction par mois-homme. Un mois-homme correspond à 165 heures (7,5 heures multiplié par 22 jours).

La productivité des projets agiles augmente plus régulièrement que dans les méthodes traditionnelles employant des équipes de taille similaire. Cela est très important, car lorsqu'une équipe conventionnelle dépasse les 10 individus, la productivité chute de façon spectaculaire. Il faut donc maintenir d'assez petites équipes et les multiplier dans la structure d'un programme lorsque la taille du produit augmente.

Pour les méthodes agiles, l'évolutivité est un point controversé. Ces méthodes ont reçu le plus d'éloges pour les petits projets et les projets de taille moyenne.

Cela ne veut pas dire pour autant qu'elles ne sont pas adaptées aux grandes applications, car plusieurs équipes peuvent livrer de plus grands projets. Les méthodes actuelles connaissent cependant une limite quant au nombre d'équipes qu'il est possible de déployer. Il arrive que la collaboration entre équipes échoue ou prenne un temps considérable. En outre, certains grands projets ne peuvent pas être partagés et exigent des approches plus strictes, bien qu'il soit toujours possible de les alléger en utilisant le processus just-enough (juste assez).

Il est difficile d'estimer la part de marché agile vu les informations limitées et souvent contradictoires associées au secteur. Scrum et XP sont sans aucun doute les principales méthodes agiles. Cependant, il n'existe pas de données permettant de savoir si les organisations utilisent l'ensemble des pratiques ou sélectionnent celles qu'elles préfèrent. Les organisations utilisent souvent plusieurs méthodes ; par exemple, Scrum et XP sont souvent utilisés ensemble.

L'utilisation de DSDM est limitée en Amérique du Nord, mais cela traduit son aspect régional (principalement au Royaume-Uni et en Europe). L'intérêt pour DSDM est de plus en plus grand, car elle s'avère être une alternative à RUP et MSF parvenue à maturité, stable et viable. FDD commence à être utilisée dans la communauté de modélisation agile, mais est essentiellement perçue comme solution pour de plus grands projets, même si la communauté parle beaucoup de l'utilisation de FDD dans les domaines XP/Scrum traditionnels.

# Comparaison des méthodes et options pour choisir une méthode de développement agile

Les principes et les pratiques lean sont partis de la fabrication avant d'arriver dans les sociétés de services informatiques. De nombreuses personnes utilisent ces pratiques dans les sociétés de services informatiques et, au-delà, dans le métier. Vu son approche holistique et sa couverture dans le secteur informatique (développement, support technique et opérations), la pratique lean semble bien placée pour dominer en matière de fourniture du processus just-enough au cours des cinq prochaines années.

Dans ce contexte, le risque est double. Le premier risque est l'échec complet. Dans ce scénario, la méthode ne sera pas adoptée par l'organisation et il n'y aura aucune réussite commerciale tangible. La mauvaise interprétation du principe et des pratiques, la résistance aux changements organisationnels et aux opinions internes peuvent empêcher la méthode de gagner en attractivité. Le deuxième risque est l'échec partiel. Dans ce cas, les objectifs professionnels sont atteints, mais l'adoption de la méthode échoue. Cet échec peut être latent, car la direction se félicite alors que les développeurs se retrouvent avec un processus qui ne peut pas être répété.

- **Faible** : la méthode peut être adoptée avec peu ou pas d'impact. Le risque d'échec du projet et/ou de faible adoption de la méthode est inférieur à 15 %.
- **Modéré** : la méthode peut être adoptée avec des changements organisationnels appropriés, mais n'a pas d'impact significatif sur la culture d'entreprise. Le risque d'échec du projet et/ou de faible adoption de la méthode est inférieur à 25 %.
- **Élevé** : la méthode peut être adoptée avec des changements organisationnels appropriés et aura un impact significatif sur la culture d'entreprise. Le risque d'échec du projet et/ou de faible adoption de la méthode est inférieur à 35 %.

## Choix de la méthode

L'approche de la méthode agile a permis la réussite de projets individuels lorsqu'il y avait un très grand nombre de partisans d'agile, mais a abouti à une situation de confusion, de frustration et d'échec lorsque ce n'était pas le cas. Pour limiter le risque d'échec, les organisations doivent être méthodiques lors du choix et de l'adoption des pratiques agiles et lean.

1. Analyse de rentabilité : disposer d'une analyse de rentabilité clairement définie pour l'adoption de la nouvelle méthode. Décrire les avantages attendus au niveau de la productivité, de la qualité, de la diminution du cycle et de la satisfaction des clients.

2. Compréhension : clairement comprendre les valeurs, les principes et les pratiques de chaque méthode en cours d'évaluation.
3. Impact : pour chaque valeur, chaque principe et chaque pratique de la méthode, évaluer comment la méthode cadre avec votre organisation en tenant compte des éléments suivants.
  - Structure et fonction organisationnelles : prévoir de petites équipes multifonctionnelles et dotées de moyens. Évaluer les effets des équipes sur la qualité, la gestion et l'architecture du projet.
  - Portefeuille de projets : quel pourcentage du portefeuille actuel convient au développement agile ? Tenir compte des langages actuels et de l'architecture et envisager l'interne plutôt que les packages.
  - Personnes et rôles : les individus ont-ils les compétences techniques et « humaines » ? Quel est l'effet du développement agile sur les rôles actuels et quels sont les besoins en formation et en conseil ?
  - Relations externes et avec des tiers : en quoi affectent-elles la collaboration étendue et les équipes virtuelles ? Comment affectent-elles les processus et les outils de communication ?
  - Niveau contractuel : quel est l'impact potentiel sur les contrats de développement en matière de possibilités de planification, de reporting, de commercialisation et de maintenance de contrats plus souples, à risques partagés ou qui reposent sur des incitations ?
  - Interaction avec les clients : quel est l'effet d'une plus grande interaction avec l'activité tout au long du cycle de vie ? Et sur l'activité en tant que membre de l'équipe ? Quel est le niveau d'engagement attendu ?
  - Domaine de solutions : quels sont les effets potentiels sur la conformité interne et externe et sur les questions réglementaires ? Quelle est l'échelle de la solution requise et quelles sont les possibilités de partage en petites équipes ?
  - Culture d'entreprise : l'entreprise acceptera-t-elle les nouvelles méthodes de façon disciplinée, mais sans procédures normatives détaillées ?
  - Gestion de projet : les principes et les pratiques agiles vont-ils s'adapter au cadre de gestion ? Quels seront les effets sur le reporting et le contrôle du projet ?

# Comparaison des méthodes et options pour choisir une méthode de développement agile

4. Poids : assigner un poids à chaque méthode en additionnant la probabilité de succès de chaque pratique. Par exemple, XP a 12 pratiques ; quelle est la probabilité d'adoption (évaluée de 1 à 5) de chacune d'elles ? Valider les pratiques notées 5 pour qu'elles soient intégrées sans faute. Chercher plusieurs moyens de supprimer les obstacles des pratiques notées 1 ou 2.
  1. Adoption impossible
  2. Adoption vouée à l'échec
  3. Adoption possible, mais avec des changements fondamentaux
  4. Adoption possible avec un impact minimal
  5. Adoption réussie
  5. Choix de la méthode : en fonction des pratiques individuelles, du poids et de la note globale de chaque méthode (addition des notes des pratiques), sélectionner les méthodes pour un programme pilote. Accepter que tout choix implique des compromis. Adopter une méthode peut amener à combiner des pratiques de différentes méthodes ou à adopter une approche graduelle pour passer à des pratiques évoluées.
  6. Pilote : utiliser la méthode pilote dans un cadre d'évaluation défini. Accorder au pilote la formation, le conseil et les outils nécessaires. Sélectionner un projet qui représente bien votre domaine. Idéalement, le pilote doit durer entre 4 et 6 mois (ou moins) et avoir une bonne visibilité.
  7. Évaluer : comparer les résultats du pilote avec les définitions des critères de réussite prédéfinis dans le cadre du pilote.
  8. Communiquer : faire connaître votre réussite à la profession, aux cadres informatiques et aux développeurs : devenir porte-drapeau.
  9. Sur mesure : comprendre les pratiques qui fonctionnent et celles qui ne fonctionnent pas. Ajuster la méthode en fonction des rétroactions du pilote. Au besoin, repousser l'adoption de pratiques plus élaborées jusqu'à ce que l'organisation ait de l'expérience.
  10. Surveillance et contrôle : créer un programme de métriques agiles pour surveiller et soutenir l'initiative agile et pour garantir des améliorations permanentes.
  11. Déploiement : prévoir un déploiement en grandeur nature. Développer une stratégie d'introduction échelonnée pour les projets adaptés au développement agile.
  12. Amélioration : définir un modèle de maturité agile interne pour améliorer le processus. Définir des jalons pour l'adoption des principes agiles au niveau de l'équipe, du programme, du domaine et de l'entreprise. Utiliser chaque itération de projet comme une possibilité d'amélioration en analysant ce qui a fonctionné et ce qui n'a pas fonctionné.

## Remarque 1

### Taille du projet

La taille de projet exprimée en PF concerne une gamme de projets classique. Les projets peuvent être bien plus grands lors de l'intégration dans l'évolutivité de l'équipe. Par exemple, XP est généralement utilisée dans les projets petits à moyens (800 PF), mais le projet peut compter jusqu'à 11 520 PF pendant un certain mois (11 520 PF = 12 personnes multiplié par 4 équipes multiplié par 40 PF/SM).

## Remarque 2

### Part de marché agile

La part de marché agile ne fait pas référence à tout le marché, mais à un sous-ensemble du marché des méthodes, c'est-à-dire aux organisations qui décrivent l'utilisation des méthodes agiles.

Source: Gartner Recherche G00148081,  
David Norton, Matthew Hotle,  
30 octobre 2007

Optimisez vos investissements dans vos projets is published by IBM Corporation. Editorial supplied by IBM Corporation is independent of Gartner analysis. All Gartner research is © 2009 by Gartner, Inc. and/or its Affiliates. All rights reserved. All Gartner materials are used with Gartner's permission and in no way does the use or publication of Gartner research indicate Gartner's endorsement of IBM Corporation's products and/or strategies. Reproduction and distribution of this publication in any form without prior written permission is forbidden. The information contained herein has been obtained from sources believed to be reliable. Gartner disclaims all warranties as to the accuracy, completeness or adequacy of such information. Gartner shall have no liability for errors, omissions or inadequacies in the information contained herein or for interpretations thereof. The reader assumes sole responsibility for the selection of these materials to achieve its intended results. The opinions expressed herein are subject to change without notice.