

PUMA

Proposition pour l'Unification des Méthodes Agiles

SOMMAIRE DE LA COMMUNICATION

Jean-Pierre Vickoff

1.1	Prédiction ou adaptabilité	2
1.2	Complémentarité des méthodes	2
1.3	Tronc des pratiques communes	3
1.4	Les pratiques différenciatrices	3
1.5	Pour une agilité optimale	5
1.6	Piloter des enjeux et des risques	5
1.7	Relation MOA / MOE	6
1.8	Les choix « d'externalisation »	6
1.9	Stratégie de planification Agile	7
1.10	Le PAQ « Agile »	7
1.11	Arrêt d'urgence, mode opératoire	7
1.12	Variabilité du niveau de méthode	8
1.13	Forme de modélisation	8
1.14	Engagement des ressources	8
1.15	Motivation de l'équipe projet	9
1.16	Les 10 commandements du RAD	10
1.17	Rappels et conclusions	10



Directeur de projet, architecte de SI, mais toujours concepteur et développeur (SWAT oblige !), JPV s'est spécialisé, en Amérique du Nord, dans la conduite de projets stratégiques sous fortes contraintes de temps.

Participant actif de l'évolution de l'état de l'art, il est l'auteur de nombreuses communications et d'ouvrages sur les méthodes. Ses travaux portent plus particulièrement sur les approches itératives incrémentales de type RAD.

Occasionnellement il réalise des conférences ou des formations pour les entreprises, les universités et les Grandes Ecoles.

Un développement associant performance et qualité induit naturellement une conduite de projet itérative incrémentielle associée à un phasage simple. A l'origine de ces principes pragmatiques, la méthode RAD de James Martin¹. UML fut plus récemment le révélateur de cette nécessité méthodologique d'un couplage fort entre la forme de modélisation et le développement par prototypage. A ce jour, une dizaine de méthodes répondent à ces critères et se réclament du qualificatif d'Agile.

Les principales méthodes Agiles sont Adaptative Software Development (ASD), Feature Driven Development (FDD), Crystal Clear, Dynamic Software Development Method (**DSDM**), Rapid Application Development (**RAD**), Scrum, Xtreme Programming (**XP**) et Rational Unified Process (**RUP**). En terme de puissance commerciale les méthodes ne sont pas nées égales ; aussi, bien que RUP soit une méthode propriétaire, il est raisonnable de la citer et de prendre son influence en considération.

Heureusement compte tenu de leur nombre, ces méthodes sont globalement similaires et la plupart des valeurs et techniques qu'elles préconisent sont communes. Une étude des principes proposés révèle un tronc commun issu des racines du RAD. Seules des techniques complémentaires les unes aux autres ou mieux adaptées à des typologies et à des tailles de projets spécifiques les différencient.

¹ Fin des années 80 pour le RAD et milieu des années 90 pour UML

Voici (en résumé) les 4 principes de base de ces méthodes Agiles :

- 1 - Les méthodes " Agiles " privilégient la **communication et l'interaction** qui en résulte à la contractualisation des spécifications.
- 2 - Les méthodes " Agiles " favorisent la **compétence et l'implication des ressources** plutôt que le respect de processus formel et d'une vision " outillée " à l'extrême des développements.
- 3 - Les méthodes " Agiles " privilégient la **livraison de fonctionnalités** réelles à la production d'une documentation pléthorique.
- 4 - Les méthodes " Agiles " favorisent l'**acceptation du changement** et la modification des **priorités** (Time-Box, Task-Box) plutôt que le respect d'une planification figée.

1.1 Prédiction ou adaptabilité

Le paradigme des méthodes classiques est la **prédictivité**. Le paradigme des méthodes Agiles est l'**adaptabilité**. Soyons clair : aucune approche n'est réduite à une seule de ces visions. Toutes tentent de composer avec la contradiction d'une souple rigidité avec plus ou moins de finesse et avec plus ou moins de succès en fonction du contexte :

- Les méthodes prédictives tentent de réduire l'incertitude dès le début du projet par une planification très précise et très détaillée. Cette levée de risque implique que les exigences de l'application soient figées.
- Les méthodes Agiles préfèrent, partant d'une planification initiale, réévaluée régulièrement, s'adapter aux évolutions du contexte. La réévaluation servira de base à une prise de décision de type GO ou NO GO (figure 1) à chaque grand changement appliqué au projet initial.

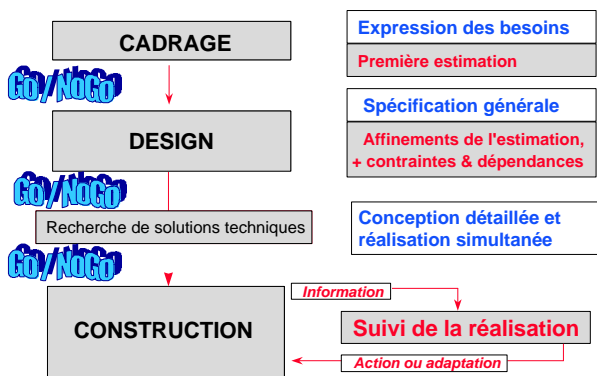


Figure 1 - Processus permanent d'évaluation / décision

Toutes les méthodes se situent concrètement à divers degrés sur une échelle les positionnant de la plus "prédictive" à la plus "adaptative". Si l'on souhaite disposer d'une vision relativement précise du champ d'application des diverses méthodes actuelles, il est nécessaire de les répertorier en fonction de ces critères (figure 2).

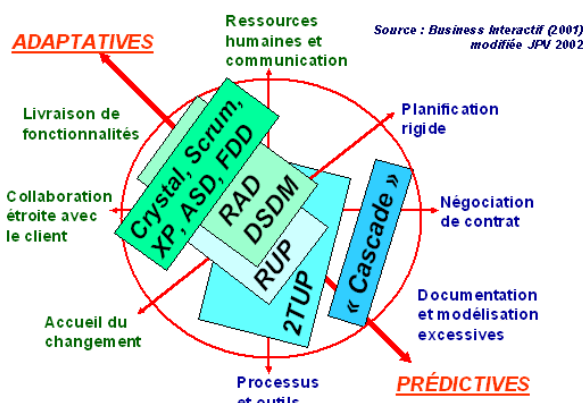


Figure 2 - Adaptativité / Prédictivité des méthodes

Les différences ainsi mises en exergue (figure3) justifient alors l'analyse fine de l'environnement du projet et du type de l'application afin de déterminer quels aspects de la méthode permettront de limiter les risques ainsi révélés et quels niveaux de service méthodologique et de qualité applicative seront mis en œuvre.

1.2 Complémentarité des méthodes

L'étude des méthodes agiles les démontrent similaires dans leurs fondements. Seules quelques techniques complémentaires entre elles ou mieux adaptées à des typologies et à des tailles de projets spécifiques les différencient. Par contre, ces méthodes offrent des couvertures plus ou moins complètes en regard des préoccupations génériques d'un chef de projet :

- 1 - **Respect de l'urbanisation** (positionnement du projet dans le système d'information).
- 2 - **Pilotage** (gestion des ressources, planning, suivi, qualité, reporting, visibilité).
- 3 - **Ingénierie de l'application** (gestion des Exigences, conception et développement, validation des livrables).
- 4 - **Conduite du changement** (impacts organisationnels et déploiement).

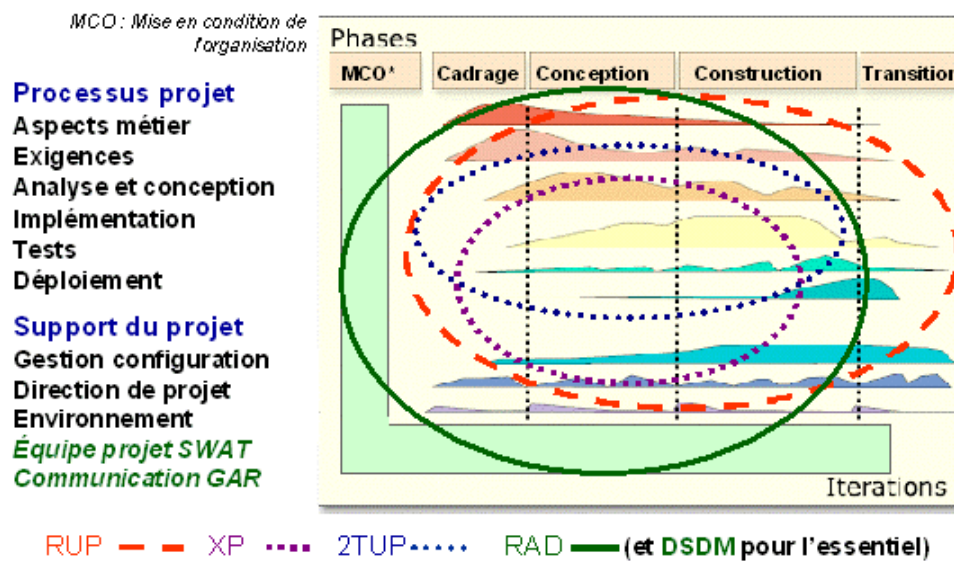


Figure 3 - Couverture de la méthode à l'ingénierie de l'application et à la conduite du projet

1.3 Tronc des pratiques communes

- Spécification et **validation permanente** des **Exigences**.
- Participation de l'**utilisateur final** aux groupes de travail.
- Interlocuteur ayant **pouvoir de décision**.
- **Autonomie** et organisation centralisée de l'équipe (motivation).
- **Niveau méthodologique variable** en fonction des enjeux du projet.
- Pilotage centré sur les **enjeux et les risques**.
- Recherche d'excellence **technique de la conception**.
- Vision graphique d'une **modélisation nécessaire et suffisante**.
- Vision d'une **documentation intégrée nécessaire et suffisante**.
- Planification stratégique globale basée sur des **itérations rapides**.
- Réalisation en **jalons** par **prototypage** actif itératif et incrémental.
- Normes et techniques raisonnables de **qualité du code** (métrique).
- Architecture à base de **composants, gestion des changements**.
- Recherche continue d'optimisation et d'amélioration des pratiques.

1.4 Les pratiques différenciatrices

Passons en revue les plus marquantes des pratiques différenciatrices.

La méthode DSDM se particularise par la **spécialisation des acteurs** du projet dans une

notion de "rôles". Ainsi, l'on trouvera dans les réunions DSDM, des sponsors exécutifs, des ambassadeurs, des utilisateurs visionnaires, des utilisateurs conseillers, sans oublier l'animateur - facilitateur et des rapporteurs.

Le Groupe d'Animation et de Rapport représente d'ailleurs un sérieux apport du RAD à la communication de projet et à la formalisation des exigences applicatives.

La méthode SCRUM affirme sa différence dans des pratiques de courtes **réunions quotidiennes**. Ces temps de travail commun ont pour objectifs d'améliorer la motivation des participants, de synchroniser les tâches, de débloquer les situations difficiles et d'accroître le partage de la connaissance.

Pour FDD, la particularité nommée **Mission focused** réside dans une forte orientation vers un but immédiat mesurable. C'est en fait l'ambition globale d'une itération qui se trouve ainsi renforcée.

Cet aspect se retrouve aussi dans le RAD sous la forme des objectifs de Focus ou dans Scrum dans la notion de *Sprint*.

Comme son nom l'indique, FDD préconise le **Features Driven Development**. Cette technique se caractérise par des notions de *Feature* et de *Features set* (fonctionnalités et groupe de fonctionnalités). La priorité est donnée aux fonctionnalités porteuses de valeur.

Le RAD propose des techniques proches : livraison en fonctionnalité réduite ou livraison par thème.

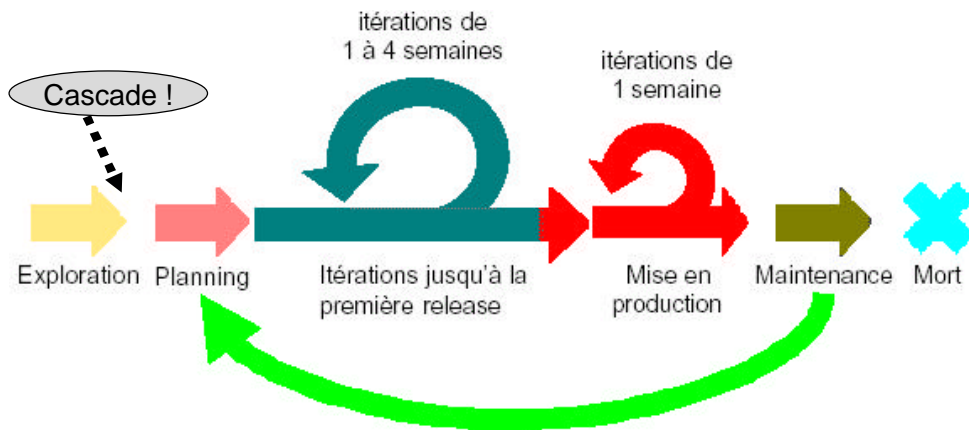


Figure 4 - Le cycle projet de XP (les autres méthodes Agiles épousent un cycle identique)

La méthode XP est très axée sur la partie Construction de l'application. Une de ses originalités réside dans l'approche de planification qui se matérialise sous la forme d'un jeu intitulé **planning game** et qui implique simultanément les utilisateurs et les développeurs. On notera aussi des techniques particulières liées à la production du code comme la **programmation en binôme**, l'**appropriation collective du code**, le **refactoring** et l'**intégration continue**.

La méthode RAD préconise dans ce sens des revues de code personnelles, puis collectives et l'intégration avant chaque Focus. Par contre, le RAD limite la programmation en binôme aux parties les plus stratégiques ou les plus complexes de l'application.

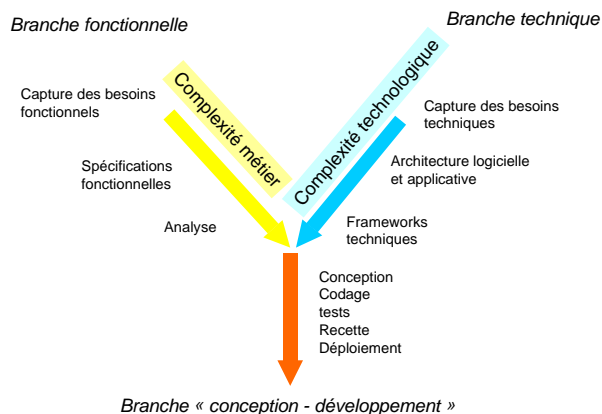


Figure 5 - Le cycle en Y de 2TUP

La méthode 2TUP² préconise un **cycle de vie en Y** qui dissocie et parallélise la résolution des questions fonctionnelles et techniques. Le cycle de vie de 2TUP s'apparente à un cycle en

² Un mélange de cycle en V et d'UDSP selon Miguel Moquillon ou un mélange de conduite de projet et de modélisation selon Jean-Pierre Vickoff.

cascade mais introduit une forme itérative interne à certaines tâches. Il n'est pas certain que ce cycle s'apparente réellement à une approche Agile. Par contre, 2TUP préconise des formes de recherche de qualité et de performance intéressantes telles que les services réutilisables et la conception générique (**Framework** et **Design pattern**) proches des mécanismes architecturaux de RUP.

RUP³ se caractérise par une approche globale nommée "**Vue 4+1**". Les 5 composants⁴ de cette vue sont : la vue des Cas d'utilisation, la vue Logique, la vue d'Implémentation, la vue du Processus, la vue du Déploiement. RUP offre aussi, à l'identique du RAD, un **Processus guide** formel et adaptable comme guide d'activité. Dans le cas de RUP, il est malheureusement propriétaire et orienté outil.

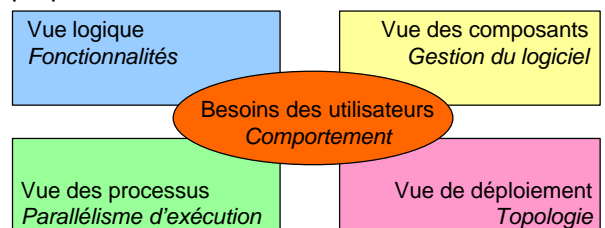


Figure 6 - La vue 4+1 de RUP

³ Fondement du développement objet **USDP** (Unified Software Development Processus) a été développé par les auteurs d'UML pour spécifier les principes du modèle d'analyse et de conception. Ses principales implémentations sont **2TUP** de Valtech et **RUP** de Rational.

⁴ Ph. Kruchten proposa en 1995 5 différentes perspectives, indépendantes et complémentaires : la **vue ("4+1")** pour définir un modèle d'architecture qui servira de base à UML..

La méthode RAD (*cycle figure 7*) préconise un groupe d'animation et de rapport (**GAR**), ainsi qu'une équipe de développement particulière, le **SWAT**. Cette équipe est autonome, spécialement formée, concrètement motivée et outillée. Elle se compose essentiellement d'un profil unique de concepteurs-développeurs formés à des spécialités techniques complémentaires. Le SWAT travaille avec les utilisateurs dans une salle permanente spécialement équipée formant un plateau de communication (**salle RAD**).

Le RAD recommande aussi la **variabilité de la taille et de la maturité⁵ des groupes de travail** en fonction des phases du projet afin d'optimiser l'engagement des ressources et de préserver leur intérêt par un travail adapté à leurs préoccupations. L'organisation performante des réunions est basée sur un mode opératoire des entretiens (**sessions en 3 étapes**) et sur des techniques de **validation permanente**.

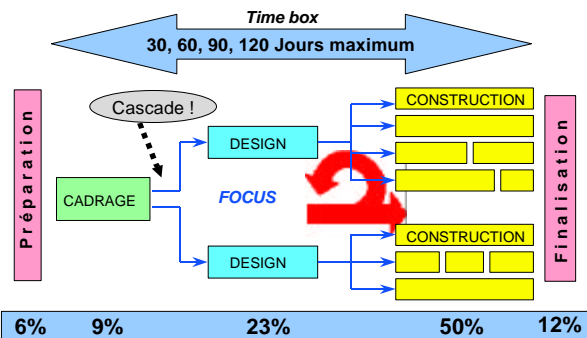


Figure 7 - Cycle RAD : parallélisation et sérialisation

1.5 Pour une agilité optimale

Si une tentative d'unification des méthodes Agiles devait se heurter à de trop grandes difficultés compte tenu de leurs origines respectives (USA), l'émergence d'une méthode unificatrice de l'ensemble des bonnes pratiques qui les caractérisent devrait cependant s'avérer possible.

Une fois les pratiques communes et les pratiques différenciatrices isolées, il est aisé d'imaginer ce que devrait être la méthode optimale en fonctions d'un type particulier de projet (*figure 8*).

La méthode Agile unifiée se composerait de l'ensemble ou d'une sélection des pratiques communes auxquelles il conviendrait d'ajouter la ou les pratiques spécifiques judicieuses en fonction du contexte. L'ensemble de ces aspects s'inscrivant obligatoirement dans un niveau variable de service méthodologique.

⁵ La mise en œuvre de ces distinctions est de la responsabilité de l'animateur - facilitateur.

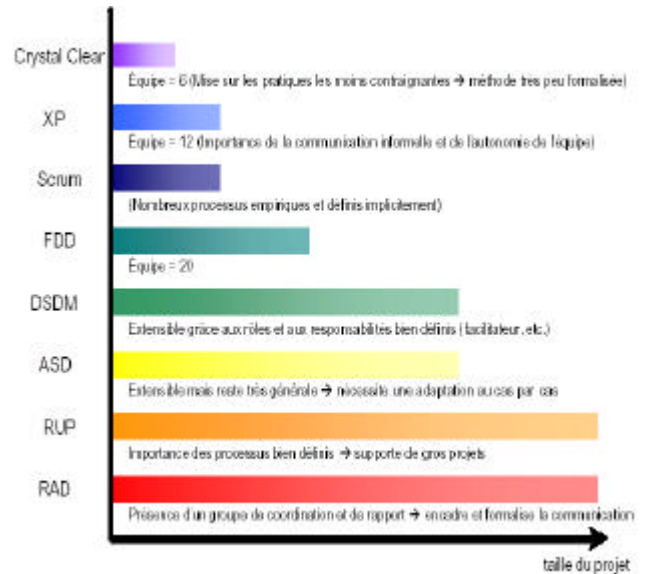


Figure 8 - Taille du projet / adaptation de la méthode

1.6 Piloter des enjeux et des risques

Le pilote d'un projet de développement est confronté en général à la maîtrise de 3 processus distincts et concourants :

- Piloter un projet **rentable**.
- Construire une application **adaptée**.
- Organiser un changement **attendu**.

Chacun de ces aspects s'inscrit dans une problématique d'enjeux et de risques. Un processus sécurisé de conduite de projet couvrira donc l'ensemble de ces aspects (*figure 9*). La pathologie du projet en difficulté met globalement en évidence l'imbrication de multiples causes de défaillance. En revanche, les effets se limitent généralement à quatre types d'insuffisances :

- l'adéquation fonctionnelle,
- la qualité technique,
- les conditions économiques de réalisation,
- la préparation au changement.

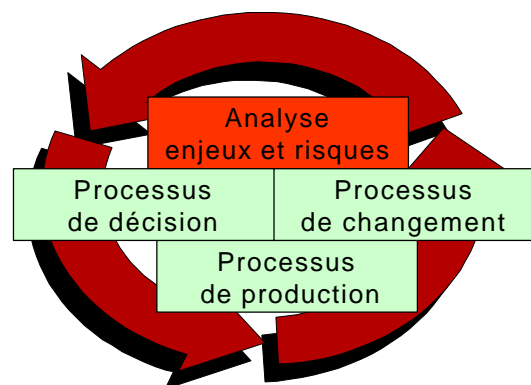


Figure 9 - Pilotage des processus par les risques

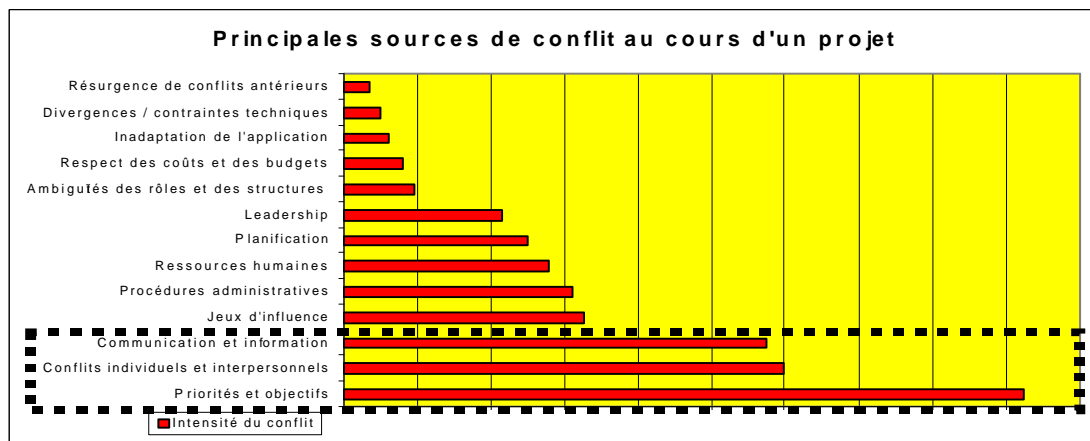


Figure 10 - Risques dans les projets (source Hervé Courtot)

Lorsqu'une, ou plusieurs de ces insuffisances se manifestent, le projet est en difficulté. Pour intervenir efficacement et recentrer un projet déviant, il faut nécessairement disposer d'une conduite de projet efficiente en regard des risques qui se sont matérialisés. Hervé Courtot, expert en ce domaine, détermine les 3 risques actuels les plus importants comme étant liés à la communication, aux rapports humains et à l'arbitrage des priorités (figure 10).

Une méthode agile unifiée devra donc s'attacher en priorité à réduire ces risques chroniques. L'apport du RAD dans son principe **animation facilitation neutre** est prépondérant pour traiter préventivement ces trois cas.

1.7 Relation MOA / MOE

Dans les applications actuelles le **métier** s'adapte à la compression du temps. Les solutions sont liées aux **technologies** et évoluent en cours de projet. La complexité des solutions actuelles impose alors une MOA sachant spécifier ses besoins actuels comme futurs et proposer une dynamique applicative. L'engagement planifié et si possible permanent des utilisateurs devient indispensable.

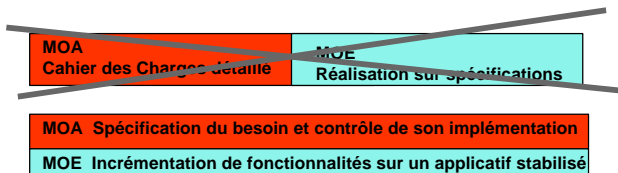


Figure 11 - La relation MOA / MOE actualisée

Pour sa part, la MOE doit se considérer comme une force de services de réalisation et de proposition technologique. De ce constat il découle que contrairement à certains discours plus que douteux, la relation maîtrise d'œuvre / maîtrise d'ouvrage n'est pas caduque ! Ce qui n'est plus adapté, c'est l'engagement séquentiel de ces deux parties (figure 11). Il est même

possible d'affirmer que cette relation n'a même jamais été aussi indispensable. Surtout si l'on prend en compte la nécessité d'externaliser l'obtention de parties de qu'il faudra contractualiser. Ce modèle répond à une logique universelle⁶ et est d'ailleurs recommandé par le CIGREF. Méthodes Agiles ou non, la définition des responsabilités doit être formelle.

Pour mieux comprendre les distinguos introduits par ces concepts, en voici une concrétisation :

- La Maîtrise d'Ouvrage a pour champ de responsabilité l'évolution des métiers de l'organisation. La MOA raisonne en termes de systèmes d'information, de processus et d'applications. La MOA dirige les projets qu'elle commande.
- La Maîtrise d'Œuvre a pour champ de responsabilité la fourniture de la solution. La MOE raisonne en termes de systèmes informatiques, de conception et de réalisation. La MOE assure le pilotage et d'ingénierie de la solution.

1.8 Les choix « d'externalisation »

Compte tenu de la complexité croissante des applications, les possibilités et nécessités d'externalisation de la solution deviennent stratégiquement indispensables. Le modèle que le RAD propose au chef de projet se présente en 7 couches externalisables en fonction du projet :

- 1 - développement de la partie GÉNÉRIQUE
- 2 - développement de la partie SPÉCIFIQUE
- 3 - stratégie de MAINTENANCE
- 4 - solution de DÉPLOIEMENT
- 5 - stratégie d'ASSISTANCE aux utilisateurs
- 6 - solution d'EXPLOITATION
- 7 - solution de LOGISTIQUE

⁶ Exemple aux USA : Owner / Chef Information Officer.

1.9 Stratégie de planification Agile

Il existe globalement 4 types de contraintes pour un projet. Le RAD propose des techniques de conduite de projet (*figure 12 colonne de droite*) adaptées à chacune de ces contraintes. Malheureusement, ces 4 contraintes sont contradictoires et vouloir les gérer à l'optimum pose un sérieux problème de combinatoire. L'outil Évaluateur, gratuit sur le site www.RAD.fr, se propose de vous aider à maîtriser cette complexité grâce à un assistant de planification stratégique de projet.


	La plus stratégique (délais)	Time Boxing
	La moins risquée (ressources)	SWAT
	La plus fiable (visibilité, qualité)	Focus
	La moins coûteuse (budget)	Target Costing

Figure 12 - Les 4 formes de contraintes du projet RAD

1.10 Le PAQ « Agile »

Pour le RAD, l'Agilité n'est pas contradictoire avec le principe de l'assurance qualité. Par contre, la qualité doit être rentable, aussi le PAQ sera par principe « minimaliste » (mais normalisé CMM). Le principe sous-jacent à sa constitution doit prendre en compte la réflexion suivante : « tout ce qui n'est pas directement la production d'une fonctionnalité est une action parasite aussi utile qu'elle puisse être ». Voici les points d'un PAQ générique pouvant être formalisés dans le PAQ spécifique du projet :

- Assurance qualité et niveau de service
- Organisation, communication, pilotage
- Gestion de la qualité du logiciel
- Méthode de développement logiciel
- Gestion des exigences
- Gestion du plan de tests
- Gestion des divergences
- Gestion quantitative des charges
- Gestion de la planification
- Gestion de la documentation
- Gestion " pragmatique " des risques
- Gestion de la configuration et des versions

A partir de cette liste, le chef de projet choisira les pratiques qui lui conviennent en terme de levé des risques pressentis. Pour résumer, ce PAQ de base contient le minimum requis pour un pilotage et un développement sécurisé.

A l'extrême de la simplification, l'assurance qualité peut se limiter à ces aspects :

- 1 - Plan de communication et d'engagement.
- 2 - Gestion des exigences, tests et divergences.
- 3 - Référentiel d'application et mémoire du projet.
- 4 - Planification, évaluation, suivi du processus.

Cette simplicité en 4 points peut se traduire textuellement ainsi : (1) Les bons interlocuteurs (maîtrise d'ouvrage et utilisateurs) se donnent les moyens (2) d'exprimer les besoins et de valider la réalisation sous une forme conservée et réutilisable (3). Le pilote du projet et les techniciens de la maîtrise d'œuvre garantissent alors les conditions de fabrication (4).

1.11 Arrêt d'urgence, mode opératoire

Toute méthode de conduite de projet devrait inclure un mode opératoire pour les arrêts d'urgence. Pourtant, l'étude des plus anciennes méthodes prédictives, semble mener à un constat d'optimisme : les auteurs n'ont jamais connu l'échec et leurs disciples doivent suivre cette voie.

Sur ce point la force du RAD se situe dans la présence d'un animateur-facilitateur. Cette ressource, de préférence externe, doit être neutre en regard des autres intervenants. Elle répond à une autorité supérieure à tous les participants du projet. Ainsi, lorsque le contexte stratégique, économique ou technique d'un projet évolue, ou si les conditions de réalisation se dégradent, l'animateur reporte le problème au dirigeant qui l'a mandaté. Ce dernier peut alors prendre dans les meilleurs délais et avec des informations objectives les décisions qui s'imposent.

Dans un environnement classique de projet, les acteurs impliqués sont parties prenantes de la problématique et la dissimule dans l'ambition souvent louable d'y porter remède. Le retour sur investissement, les plannings, les budgets sont donc « améliorés » afin de lisser la problématique. Cette situation conduit à des blocages dans la remontée d'informations objectives au-delà d'un premier niveau hiérarchique et aboutit finalement à une rupture dans la chaîne de décision et de commandement. A ce niveau d'altération de la réalité, toute tentative de recentrage met généralement en évidence des dysfonctionnements dans l'organisation. La notion de *Courage*, une des quatre valeurs fondamentales de *l'eXtreme Programming*, prend alors toute sa signification et certains commencent à regretter qu'une autre de ces valeurs, le *FeedBack*, n'ait pas été prise en considération plus tôt.

1.12 Variabilité du niveau de méthode

Une solution méthodologique efficace dans un environnement donné peut ne pas être adaptée à un environnement de taille, de complexité ou de nature différente. Adapter méthode et qualité de la production à la dimension économique et au cycle de vie des applications est devenu un mal nécessaire. Voici, dans l'esprit du RAD et de l'Agilité, les éléments constitutifs du niveau de service méthodologique d'un projet avec lesquels il faut composer :

Niveau de service « pilotage »

- Planification et suivi de l'avancement.
- Planification et suivi des risques.
- Planification et suivi de la sécurité.
- Formalisation des activités de pilotage.
- Formalisation Exigences / Divergences
- Formalisation de la gestion de configuration.
- Formalisation de la gestion des tests.

Niveau de service « application »

- Qualité technique de l'application.
- Qualité fonctionnelle de l'application.
- Qualité de la documentation technique.
- Qualité de la documentation d'utilisation.

On notera la séparation faite entre le pilotage du projet et l'ingénierie de l'application.

Pour exemple : on ne documente pas à l'identique une application dédiée à quelques cadres permanents et une application prévue pour des centaines d'utilisateurs en fort turn-over.

1.13 Forme de modélisation

En fonction du contexte, la forme⁷ et le niveau de modélisation devraient pouvoir être choisis. A l'évidence, la plupart des méthodes récentes sont fortement influencées par l'Objet et plus particulièrement par UML. Par contre, la nécessité d'imposer sans discernement une modélisation UML complète⁸ est de plus en plus controversée.

Un courant de pensées simplificateur, l'Agile Modeling, considère cette théorisation à outrance comme inadaptée aux projets sous fortes contraintes de temps ou d'argent.

Le pragmatisme impose au pilote de projet de conserver à l'esprit que **l'objectif est une application** et non un modèle. De par mon expérience, je vois se profiler avec UML des

excès identiques à ceux auxquels avait abouti Merise voici quelques années. Le plus inquiétant est le fanatisme des tenants de l'objet. Aucune critique des pratiques de leur secte n'est entendue.

L'Objet propose depuis de nombreuses années des techniques de conception qui ne sont pas liées à une forme de modélisation particulière. Citons parmi les plus connues ou utiles la dissimulation, la modularité, l'abstraction, l'encapsulation, la cohésion, le couplage, la hiérarchisation, l'héritage, le polymorphisme, l'algorithmique de base, la structuration (des données et des traitements).

Le but ultime d'une conception efficace est de faciliter le développement et la maintenance ultérieure mais aussi la conduite de projet dans sa rencontre avec le nerf de la guerre : le retour sur investissement. Parmi d'autres, le principe de « conception en vue de modifications » a été imaginé pour ces raisons. Il se base essentiellement sur la technique de « dissimulation d'information ». Pour la mettre en œuvre il est nécessaire de respecter les étapes suivantes :

- Définition exhaustive des fonctionnalités.
- Identification des règles de gestion instables.
- Encapsulation des parties modifiables.
- Définition formelle des interfaces d'appel.
- Programmation modulaire.

Les gourous du développement qui ne comprirent pas tout de suite l'intérêt de cette pratique furent contraints rapidement de faire amende honorable. Fred Brooks « La critique de cette technique est une des rares erreurs de la première édition de mon livre, *The Mythical Man-Month* ». Boehm « C'est une des rares techniques théoriques qui ait fait la preuve de son utilité pratique ». Quelle que soit l'expression de modélisation choisie, mettre en œuvre systématiquement de bonnes pratiques de conception serait déjà un grand pas. Selon des statistiques de Microsoft (*54 Règles pour un grand logiciel*) 97.5 % des informaticiens ne connaîtraient pas ou ne pratiqueraient pas ces techniques basiques de conception.

1.14 Engagement des ressources

La nouvelle nature des applications requiert l'expression d'un besoin dont la complexité s'accroît régulièrement. Dans le même temps, les technologies chargées d'apporter des solutions subissent, elles aussi, des évolutions majeures. De plus, la contrainte principale est de plus en plus souvent liée à la compression du temps. **De ces pressions découle le Mode Projet.**

⁷ Merise avec ses niveaux d'abstraction et de préoccupation, les Flux associés à l'entité-relation, l'Objet, ...

⁸ Pas moins de 9 types de diagrammes .

Cette organisation concentre dans un même espace géographique et temporel l'ensemble des intervenants fonctionnels et techniques engagés dans une unique mission. Les projets menés suivant cette approche sont des succès. Au regard de ce constat, on est en droit de s'étonner de la continuité, voire du développement d'une gestion matricielle des ressources. Les fervents de cette approche, en général de purs gestionnaires, espèrent pouvoir optimiser l'engagement de leurs ressources sur de multiples projets simultanés. Théorie parfaite en ce qui concerne des petits travaux de maintenance, mais battue en brèche par une réalité différente dans les projets de développement : **le crime de dispersion ne paie pas !**

Les causes sont profondément humaines : perte de reconnaissance et de responsabilité⁹ dans l'aboutissement du projet. En pratique, il m'a été donné d'observer la totale absence de participation d'individus alors qu'ils étaient, théoriquement, affectés entre 5 et 15% de leur disponibilité ! D'autre fois la pléthore de ressources engagées aboutissait à des résultats semblables : « Lorsque tout le monde semble s'occuper de tout, en général personne ne s'occupe de rien ».

Une équipe RAD reste de dimension raisonnable (4 personnes à l'idéal, 8 au maximum), mais l'ensemble de l'équipe est engagée en bloc au début du projet (Cadrage).

1.15 Motivation de l'équipe projet

Les méthodes Agiles s'efforcent de travailler avec les spécificités des ressources humaines plutôt que contre leur nature. La meilleure garantie de succès d'un projet reste l'engagement d'une équipe motivée. La capacité de travailler en groupe et de communiquer avec l'utilisateur est primordiale.

Le RAD recommande un forme d'équipe particulière le SWAT (*Special Weapons and Tactics*) qui devient dans l'engagement d'un projet *Skill With Advanced Tools*.

Les membres de l'équipe ont un profil unique¹⁰ de concepteur-développeur mais sont surentraînés dans une spécialité et formés aux techniques d'entretiens RAD. Basant leur mission sur la communication avec l'utilisateur, ils se complètent et s'appuient.

Afin de motiver exceptionnellement une équipe, la méthode RAD préconise de :

- Définir précisément le défi à relever.
- Formaliser la récompense (en fonction de l'atteinte des jalons définis).
- Enrôler l'équipe en donnant le choix de l'engagement à chacun individuellement.
- Organiser l'autonomie de l'équipe et termes de moyens et d'organisation du travail.
- Comptabiliser (simplement) les actions d'assistance entre les membres du SWAT.
- Personnaliser les travaux en fonction des désirs et des possibilités des ressources.
- Publier régulièrement l'avancement des résultats (sans langue de bois).
- Surveiller le tarissement de l'enthousiasme et récompenser les efforts par divers avantages.
- Fêter l'accomplissement des jalons significatifs à partir d'un budget dédié (pots, restaurants, sorties).
- Favoriser l'émergence de signes d'identité de l'équipe (pas très bien compris en France).

Le retour sur investissement de ces pratiques "à l'américaine" est immédiatement évident.

Sur le plan technique, comme le recommande la structure d'un SWAT, favorisez la formation de généralistes-experts¹¹. Répertoirez ces expertises puis affichez un tableau des ressources à consulter en cas de questions pointues. Cela permettra de développer une culture d'entraide entre les membres de l'équipe.

Afin que les plus compétents ne soient pas pénalisés par le nombre des interventions de soutien, il faut mesurer l'échange de compétences et de services. Ceci permet de gagner un temps précieux en recherche. Ceci permettra aussi d'éliminer les craintes du « demandeur » qui pourraient découler de réactions négatives de l'expert si ce dernier n'était pas dédommagé de son temps investi dans le transfert de compétences.

Ce dernier point acquis, **il faut focaliser la plus grosse partie des récompenses sur les résultats d'équipe et non individuels**, car le SWAT doit rester une synergie et une complémentarité de spécialistes engagés dans un succès d'équipe.

¹⁰ L'introduction d'un élément fonctionnel dans un SWAT est l'idéal dans les développements modernes exigeants (High-Tech, High-Touch).

¹¹ généraliste de la conception et du développement mais expert dans un domaine complémentaire

1.16 Les 10 commandements du RAD

Bien que la chance ait sa part dans un projet, comme dans la plupart des activités humaines, respecter des règles de bonne conduite (de projet) permet de s'assurer qu'elle ne sera pas gaspillé par négligence.

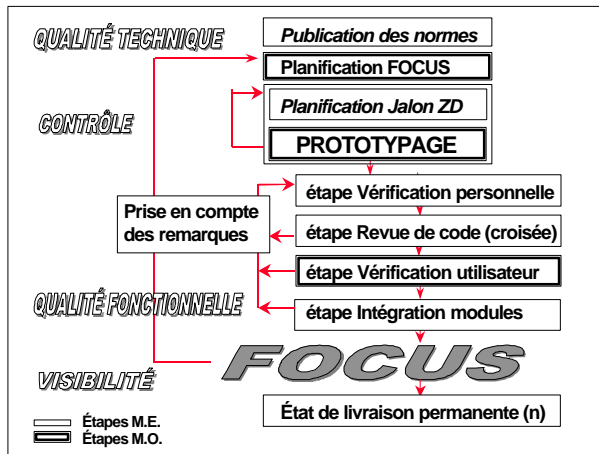


Figure 13 - Techniques de maîtrise de la réalisation

Voici les 10 commandements du RAD :

- 1 - Exigez un environnement organisationnel et technologique efficient.
- 2 - Pratiquez la validation permanente avec l'utilisateur réel.
- 3 - Planifiez de manière réaliste au début puis réévaluez régulièrement (GO/NOGO).
- 4 - Respectez le phasage et les pré-requis de poursuite.
- 5 - Respectez la dimension temporelle (optimum 90 Jours).
- 6 - Validez immédiatement la faisabilité opérationnelle (cycle en Y de 2TUP).
- 7 - Composez une équipe techniquement et fonctionnellement adaptée au défi.
- 8 - Appliquez les concepts fondamentaux de la conception Objet.
- 9 - Appliquez les techniques de maîtrise de la réalisation : Jalons ZD, revue de code (figure 13).
- 10 - Assurez le respect des livrables et des Focus de visibilité.

1.17 Rappels et conclusions

Comme le met en évidence le tableau « facteurs de succès et causes d'échecs » (annexe 2), dans la plupart des projets les échecs relèvent rarement de la technique à elle seule. De

dangereux écueils apparaissent dilués dans les méandres des relations humaines et des impératifs économiques.

Cette constatation est particulièrement évidente lorsque le projet se situe sur le terrain de la nouvelle économie, de l'innovation, du trans-domaines et parfois même du trans-organisations. La réussite du projet nécessite alors d'associer simultanément deux aspects généralement contradictoires : la **créativité** indispensable aux ambitions d'innovation différenciatrice et la **rigueur** de contraintes économiques ou stratégiques. Le pilotage par **les enjeux, les risques et les contraintes** s'affirme alors comme une nécessité vitale.

De ce constat émerge une certitude : les nouveaux projets font évoluer en profondeur le cadre de travail des DSI, des chefs de projet et des maîtres d'ouvrage. Il leur faut se doter d'une structure de développement formelle en terme d'actions et de responsabilités.

Apparaît ensuite la nécessité d'une **méthode de développement semi-itérative** sécurisée par un processus fiable et adaptable.

Pour être économiquement viables, l'ensemble de ces aspects s'inscrivent obligatoirement dans un **niveau variable de service méthodologique et de qualité requise**.

De plus, l'innovation, lorsqu'elle atteint un certain degré, affecte fondamentalement le cœur du métier. Le projet fait alors office de révélateur quant à la nécessité d'une **remise à plat des processus** que d'ailleurs l'Objet impose naturellement. A partir de ce constat, c'est souvent l'organisation complète qui se doit d'évoluer. Une telle amplitude de changement de mode de travail, lorsque ce n'est pas directement de culture, est encore exceptionnelle et représente pour l'encadrement « le temps de tous les dangers ». Les freins à l'évolution sont donc puissants.

La mise en œuvre de la méthode, dans ses aspects de communication élargie, doit en tenir compte.

Au delà de la maîtrise des projets, l'essentiel dans l'usage d'une méthode itérative et incrémentielle s'appuyant sur l'utilisateur, c'est le **dynamisme** qu'elle induit dans l'organisation parallèlement à l'acceptation du changement qu'elle favorise.

Jean-Pierre Vickoff

ANNEXE 1 - Principaux facteurs de succès et causes d'échecs des projets actuels

Aspects	Facteurs de succès	Causes d'échecs
Politique	Le directeur du projet et/ou le responsable du groupe d'animation et de rapport répond à une autorité supérieure (à la direction générale dans un projet stratégique)	Le directeur de projet ne dispose pas d'un mandat suffisant pour arbitrer les divers intervenants, limiter leurs luttes d'influence, leurs ambitions ou visions partisans.
Pilotage	Piloter le projet <i>dynamiquement</i> essentiellement par les enjeux et les risques .	Piloter le projet <i>administrativement</i> essentiellement par les budgets et les ressources .
Organisation	Engager un mode projet centralisant sur un plateau unique tous les intervenants aussi bien sur le plan organisationnel que géographique. Formaliser dans le cadre de courtes missions de réelles délégations de responsabilité verticales et latérales.	Utiliser des ressources à temps partiel et/ou non dépendantes directement de la direction de projet. Ne pas disposer de l'agenda électronique de l'ensemble des intervenants. Négliger la couverture des participants en terme de responsabilité hiérarchique.
Financier	Justifier le projet par un plan d'investissement global mais simple , dont chaque élément est une base réaliste et acceptée.	Créer et maintenir un modèle financier dont le niveau de détail interdit ensuite sa remise en question et dont la complexité fera ensuite douter de sa pertinence .
Planification	Intégrer dès le début du projet l'ensemble des sous-projets et des contraintes dans un planning réaliste qui sera ensuite suivi jalon par jalon avec un outil professionnel léger.	Négliger un contrôle approfondi sur l'avancement des diverses parties sous traitées ou réalisées dans des sites éloignés de la direction du projet
Innovation	Viser l'utilisation de <i>technologies émergentes</i> dont la stabilisation est prévue pour la date du déploiement afin d'obtenir l'optimum d'efficacité stratégique	Se limiter aux technologies ou puissance de machine <i>disponible à la date du cahier des charges</i> ou de la réalisation pour des raisons administratives ou contractuelles
Communication	Employer un groupe d'animation et de rapport disposant de moyens modernes pour dynamiser la communication ainsi que formaliser et centraliser l'information.	Ne pas distinguer les groupes de travail de ceux de validation . Laisser les intervenants organiser en permanence des réunions <i>brainstorming</i> , non préparées ou non structurées et à participation variable.
Méthode	Cycle de vie itératif incrémentiel : réaliser des livraisons de résultats partiels sous la forme de plusieurs Focus de validation, suivis d'un prototype final, puis d'un site pilote.	Cycle de vie cascade classique : avant même de lever les risques organisationnels ou techniques, viser directement un système totalement finalisé dans la vision d'un déploiement total et parfait.

ANNEXE 2 : Analytique cartésien (prédicatif) / Pragmatique empiriste (adaptatif)

Comme le met en exergue le tableau suivant, les approches prédictives et adaptatives se réclament de postulats épistémologiques fondamentalement différents.

Méthodes	Dites classiques ou « Lourdes »	Dites nouvelles ou « Agiles »
Paradigme fondateur	Prédictivité	Adaptabilité
Fondement	Analytique cartésien	Pragmatique empiriste
Cycle projet	En cascade (sans rétroaction)	Incrémentiel et Itératif (adaptatif)
Forme de levée du risque	Descriptive et documentaire	Recherche - action - expérimentation
Raisonnement	Discursif (prémisses-conclusions)	Systémique et heuristique
Vision sous-jacente	Isoler pour structurer une partie d'univers figé	Exécuter pour comprendre la dynamique des interactions
Pensée	Réductionnisme et hypothèses mécanistes	Vision holistique des phénomènes (ressources humaines, communication, environnement, ...)
Philosophie d'analyse	Considère la nature des interactions	Considère les effets des interactions
Structuration méthode	Sur la base de niveaux isolants d'abstractions et de préoccupations figés	Sur la base d'un phasage simple et souple prenant en compte les contraintes du projet
Axe de recherche	L' analyse de la structure	L' aboutissement des actions
Limites et possibilités	Réduction de systèmes simples par l'analyse	Appréhension de systèmes complexes par leur
Conduit à des systèmes	A forte entropie	A forte rétroactivité , « cybernétique »
Aboutissement	Recherche l' exhaustivité de la solution	Accepte un « rendement satisfaisant »
Philosophie d'action	Conduit à une action totalement détaillée et programmée	Conduit à une action par objectifs et flexible
Aboutit concrètement	A la reconduction de structure existante	A l' amélioration des éléments de performance
Validation	Comparaison théorique à base de jeux d'essais en fin de parcours	Confrontation permanente du modèle avec la réalité (prototype)

Bibliographie

Livre blanc : *Méthode Agiles état des lieux*, www.BusinessInterActif.fr

[Paulk], *The Capability Model : Guidelines*, SEI, 1995

[P. Roques et F. Vallée], *UML en action*, Edition Eyrolles

[Philippe Kruchten], *Rational Unified Process*, Edition Eyrolles

[Vickoff (J-P)], *Piloter les projets informatiques de la nouvelle économie*, Editions d'Organisation, 2000