

CONSERVATOIRE NATIONAL DES ARTS ET METIERS  
PARIS

**RAPPORT**  
présenté en vue d'obtenir  
**L'EXAMEN PROBATOIRE**

en

INFORMATIQUE, RESEAUX, SYSTEMES ET MULTIMEDIA

par

Julien Silverston

**Le point sur le RAD**

Soutenu le 29 mai 2007

Devant le jury composé de :

Président :

Professeur POLLET



## Table des matières

1.Introduction.....	5
2.La méthode RAD.....	6
2.1.Histoire de la méthode RAD.....	6
a)Méthode Waterfall.....	6
b) La méthode RAD : une synthèse.....	7
2.2.Objectifs.....	8
2.3.Principes.....	8
a)Les hommes.....	9
b) Les outils.....	11
c)Un management.....	12
d)Une méthodologie.....	12
2.4.Cycle de vie RAD.....	16
a)Initialisation.....	16
b) Expression des besoins.....	17
c) Conception.....	17
d) Construction.....	18
e) Mise en œuvre.....	19
3.La place de la méthode RAD dans la gestion de projet.....	20
3.1.RAD, une méthode AGILE ?.....	20
3.2.RAD, une méthode souple.....	21
a) RAD – UML – MERISE.....	21
b) RAD – Extrem Programming (XP).....	22
c) RAD – la programmation Objet.....	22
3.3.Etat des lieux de RAD.....	23
a) Avantages et inconvénients de la méthode .....	23
b) Quels projets pour le RAD ?.....	24
c) Souci de performance des systèmes RAD.....	25
d) La méthode RAD, incompréhensions.....	25
e) Des fonctionnalités réduites.....	25
f)Le RAD nécessite un changement de culture.....	25
4. Conclusion.....	27
5. Annexe 1 – Outils RAD.....	28
6. Bibliographie.....	29
7. Glossaire.....	30

## Index des illustrations

Illustration 1: Cycle de vie de la méthode Waterfall.....	6
Illustration 2: Cycle de vie RAD.....	7
Illustration 3: Principes RAD.....	9
Illustration 4: Equipe RAD.....	10
Illustration 5: Cadre de la méthodologie RAD.....	13
Illustration 6: JAD.....	14
Illustration 7: Mécanisme d'une Timebox.....	15
Illustration 8: Déroulement d'un projet RD.....	19
Illustration 9: Position de RAD face aux méthodes Agiles.....	20

# 1.Introduction

L'industrie du logiciel est en perpétuelle évolution, elle doit répondre toujours plus vite à des exigences de plus en plus importantes. La puissance des équipements et les moyens de communication actuels permettent de présenter, de vendre, de se faire connaître et éventuellement d'en tirer des bénéfices.

De cette façon, l'utilisation des nouvelles technologies de l'information et de la communication (NTIC) peut permettre aux entreprises d'obtenir de nouvelles parts de marché ou bien de se distinguer de la concurrence. Pour cette raison les NTIC sont une ressource indispensable au développement et à l'expansion des entreprises.

Mais pour obtenir cette valeur ajoutée, l'entreprise et son système d'information (SI) doivent être réactifs. Le développement du SI doit alors être mené dans un temps très court et être d'une grande qualité. Il n'est pas possible d'attendre plusieurs années pour voir aboutir un projet, le risque d'échec et donc de perte financière serait trop grand.

C'est pour atteindre ces objectifs que James Martin a élaboré dans les années 80 une nouvelle méthode de développement, la méthode RAD (Rapid Application Development). Cette méthode est basée sur une approche incrémentale du développement, l'utilisation de prototype évolutif et donne une place prépondérante aux personnes et à la communication.

Tout d'abord, je présenterai la méthode RAD puis sa place dans la gestion de projets.

## 2. La méthode RAD

La méthode RAD est apparue après une longue évolution des méthodes de développement.

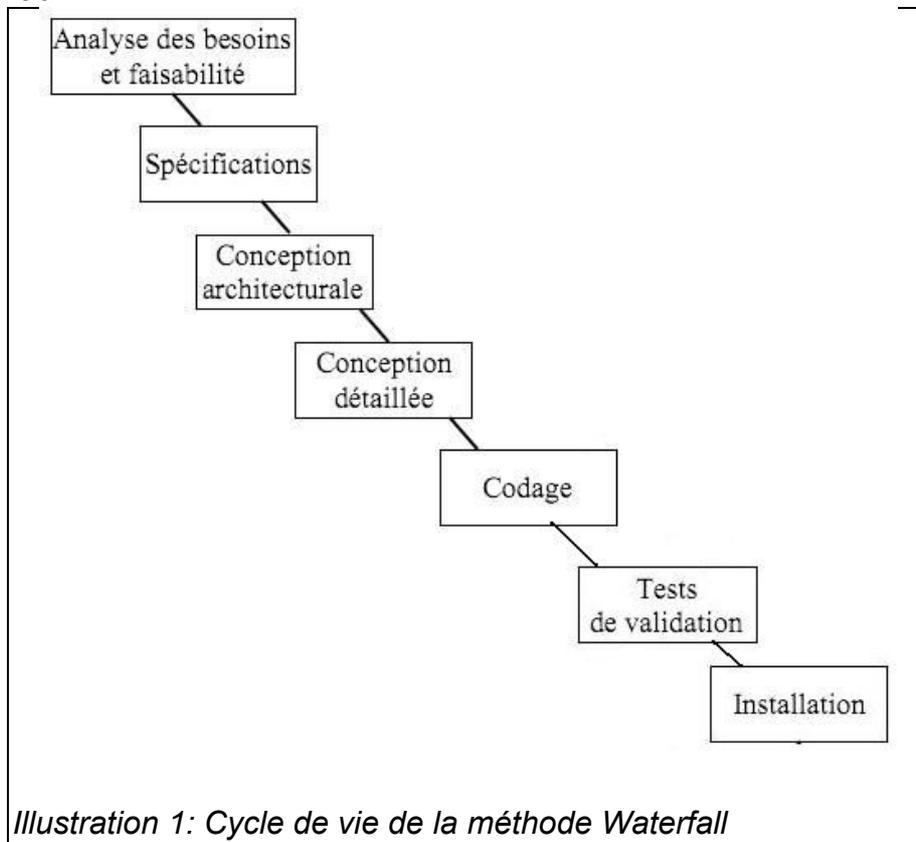
### 2.1. Histoire de la méthode RAD

#### a) Méthode Waterfall

Jusqu'à la fin des années 80, la méthode de développement employée par l'industrie du logiciel était la méthode en Cascade « Waterfall ». Elle suit un cycle de développement en séquences où les phases s'enchaînent l'une après l'autre. La mise en œuvre d'une nouvelle phase dépend de la complétude de la précédente.

C'est une adaptation de la méthode d'ingénierie classique qui consiste à définir dans les moindres détails, les spécifications attendues du projet à l'intérieur d'un cahier des charges. La réalisation du projet peut alors être réalisée par différents intervenants ou sous-traitants.

Cette méthode s'applique bien à l'ingénierie classique (génie civil) mais bute sur de nombreuses difficultés lorsqu'il s'agit de développements informatiques dictés par des utilisateurs finaux.



En informatique, il n'est pas toujours facile de recueillir de la part des utilisateurs l'ensemble des spécifications. Bien souvent ils ne savent pas décrire le système attendu mais ils sont capables de le reconnaître, si quelqu'un le leur montre.

En outre, le temps de réalisation du système est souvent long et difficilement estimable à l'avance en suivant cette méthode.

Enfin les besoins, le marché, la réglementation peuvent évoluer. L'application créée arrive alors trop tard et n'est pas en mesure de donner à l'entreprise le bénéfice recherché. Elle risque d'autre part de ne pas recevoir l'adhésion des utilisateurs car ils interviennent uniquement durant la phase d'analyse des besoins, au début du cycle et ne sont plus sollicités jusqu'à la fin. Pendant ce temps, bien souvent les besoins ont évolué et de nombreuses améliorations seront à apporter et rajouteront un coût important au projet en maintenance applicative.

## b) La méthode RAD : une synthèse

Une réponse apportée à ce problème a été proposée par Barry Bohem : un cycle de développement en spirale utilisant le concept de prototype, basé sur de nombreux échanges entre informaticien et utilisateurs et ainsi réduction de la part de risque du projet.

La méthode RAD est la fusion de ces deux modèles vers la fin des années 80, au moment où la micro informatique explose et où les entreprises peuvent obtenir un avantage concurrentiel important grâce à un système d'information innovant.

Cette méthode, définie par James Martin dans son livre « Rapid Application Development » est avant tout pragmatique. Elle regroupe les bonnes pratiques de l'époque, combine les cycles en cascade et en spirale, impose un cycle développement court (entre 90 jours, et 120 jours), tire avantage des outils les plus modernes de développement (atelier de génie logiciel, AGL) et met en œuvre des techniques éprouvées (Time-Box, JRP, JAD et prototypage).

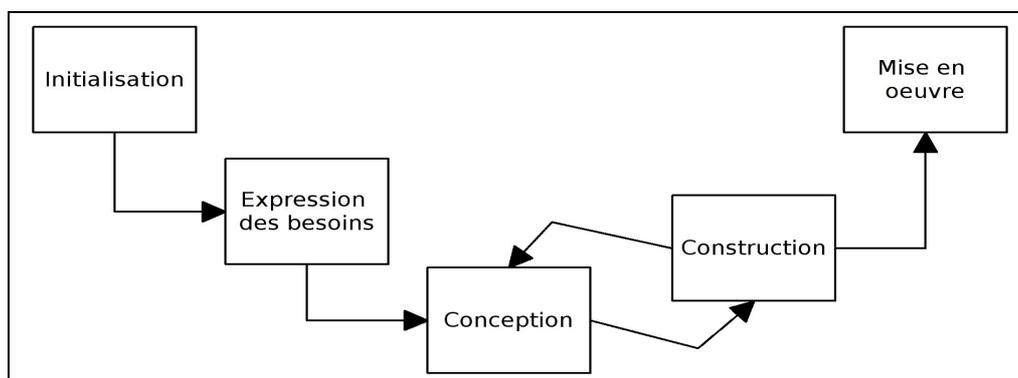


Illustration 2: Cycle de vie RAD

Le cycle RAD est constitué de cinq phases :

- Initialisation, détermine si le projet peut être suivi avec la méthode RAD,
- Expression des besoins, détermine les grandes lignes du projet : objectifs, participants, principales fonctions
- Conception, définit précisément les processus, les données
- Construction, est la phase de développement et de validation du système
- La Mise en œuvre constitue la livraison du système nouvellement créé

Les phases de Conception et de Construction sont reprises dans un cycle et chacune fait l'objet de plusieurs itérations afin de réduire les risques d'échec.

## **2.2.Objectifs**

La méthode RAD a pour objectif principal de produire un système dont la charge et le délai du projet soient réduits, sans en sacrifier la qualité et qui répondent aux besoins des utilisateurs.

Ainsi les objectifs de RAD sont de produire des logiciels :

- plus vite,
- mieux,
- moins cher.

En somme, l'objectif du RAD est de produire un logiciel utile au moment où l'entreprise en a besoin.

Pour cette raison, le cycle de vie RAD est fixé à maximum 120 jours. Le projet est contenu dans un temps court pour éviter la classique fuite en avant du projet:

- × *les délais de livraison et les attentes des utilisateurs qui augmentent retardant toujours plus la fin du projet.*

Pour atteindre ces objectifs, la méthode RAD repose sur quatre principes.

## **2.3.Principes**

RAD est basé sur un cycle de vie court et une approche incrémentale qui donne la première place aux personnes et à la communication.

La méthode RAD est une méthode pragmatique fondée sur quatre grands principes.

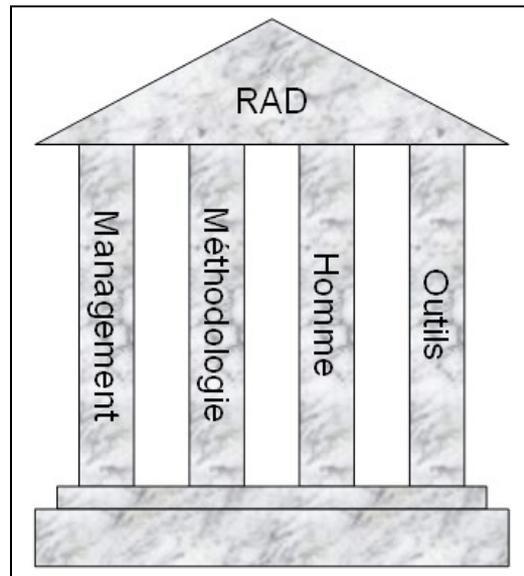


Illustration 3: Principes RAD

Si l'un des ces principes est manquant, le projet ne peut être qualifié de RAD et la construction ne sera pas rapide.

### a) Les hommes

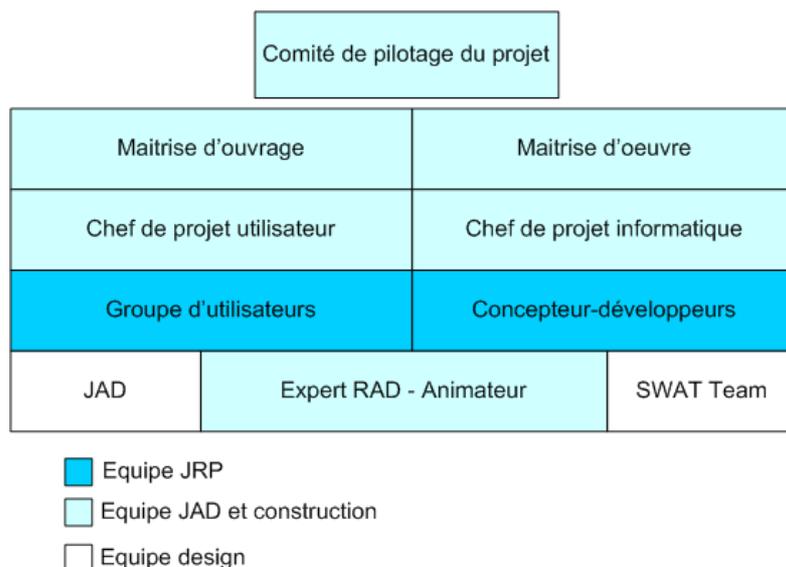
Le succès d'un projet RAD est obtenu grâce à la totale implication des personnes participantes. Elles doivent être compétentes et motivées. Les participants au projet doivent être choisis avec soin, ils doivent connaître et adhérer à la méthode, connaître les règles de gestion de l'entreprise et être formés aux outils.

Les acteurs d'un projet RAD doivent être responsables, polyvalents et décisionnaires par souci d'efficacité le nombre de participants est relativement réduit, environ une dizaine de personnes.

<b>Acteurs de la méthode RAD</b> Correspondance Anglais/Français	
Sponsor	Propriétaire / Commanditaire
User Coordinator	Chef de projet Utilisateur (CPU / MOA)
Project Manager	Chef de projet Informatique (CPI / MOE)
Workshop Leader	Expert RAD, animateur
Users	<ul style="list-style-type: none"> <li>● Utilisateur décisionnaire</li> <li>● Utilisateur final</li> </ul>
SWAT Team	Equipe informatique

- Le CPU et CPI forment le duo de chefs de projet (maître d'ouvrage et maître d'œuvre)
  - Le CPU représente les utilisateurs finaux et est garant des règles de gestion du

- système.
- Le CPI représente l'équipe informatique en charge du développement.
  - L'expert RAD est le garant de la méthodologie et de la bonne marche à suivre. Il cadre le projet et anime les sessions de travail. Il est indépendant de la maîtrise d'œuvre et de la maîtrise d'ouvrage.
  - Les utilisateurs sont les clients (externe ou interne) et les futurs utilisateurs du SI. Leurs participations et leurs motivations sont les règles de base de la méthode RAD. Ils participent activement à la création du système et sont intégrés dans les équipes projets. Ils doivent connaître les règles de gestion du métier, être capables de prendre des décisions rapides et faire des retours d'information (feedback) sur le système.
    - Les utilisateurs décisionnaires valident les orientations du projet. Ils participent à la session JRP et JAD1. Pour valider les orientations, un à deux utilisateurs décisionnaires sont requis.
    - Les utilisateurs finaux travailleront avec le futur système. Ils valident son utilisation. Ils participent aux sessions JAD1 et JAD2.
    - Le propriétaire de l'application finance l'application et tranche en cas de blocage.
  - Les informaticiens disposent de moyens performants pour produire plus vite et efficacement. Ils forment l'équipe SWAT (Skilled With Advanced Tools). Il s'agit d'une petite équipe (2 à 6 personnes) pour une meilleure réactivité et pour éviter des problèmes de communication. Ils participent à toutes les phases du RAD. Ils peuvent être répartis en plusieurs équipes, chacune représentera une facette du projet (analyse, conception, prototypage, développement, tests, ...). Mais, il est préférable qu'ils soient polyvalents et puissent intervenir sur plusieurs domaines.



*Illustration 4: Equipe RAD*

## **b) Les outils**

Le succès du RAD dépend de la mise en œuvre d'outils performants et automatisés. Ces outils doivent répondre aux caractéristiques suivantes pour permettre un développement rapide :

- Utilisation d'un langage non-procédural
- Outils de prototypage permettant des développements itératifs pour tester, valider et modifier dynamiquement le système pour montrer aux utilisateurs les modifications et valider avec eux les orientations,
- Utilisation d'AGL (Atelier de génie logiciel – CASE tools) pour élaborer les modèles de conception graphiquement et produire la documentation.
- Générateur de code source directement à partir de modèles et de schémas.

Les outils RAD peuvent être regroupés en deux catégories :

### ***Les outils orientés développement***

Il s'agit principalement d'AGL (ateliers de génie logiciel), des logiciels graphiques d'aide à la conception et à création de système informatique. Ils sont utilisés pour générer le code, vérifier la syntaxe, réaliser automatiquement les prototypes et générer la documentation.

De nombreux diagrammes sont également utilisés pour représenter les données, les processus et ainsi permettre une meilleure compréhension entre utilisateurs et développeurs.

Un espace de travail commun est également nécessaire pour entreposer toutes les connaissances, les objets du projet et partager l'information entre tous les participants.

Les outils RAD mêlent la conception et la construction, c'est le propre du développement d'un prototype évolutif. Ils permettent de montrer directement aux utilisateurs l'évolution du développement.

### ***Les outils orientés communication***

La méthode RAD recommande également l'utilisation de moyens modernes de communication pour éviter de perdre du temps en tâches administratives et améliorer les échanges. Il est ainsi recommandé de saisir directement les comptes rendus de réunion sur un pc portable, d'avoir à disposition pour le projet une salle RAD qui offrira toutes les conditions pour réaliser les prototypes, ainsi que des lieux dédiés et coupés du quotidien pour les sessions de travail JRP et JAD.

Les outils RAD doivent être graphiques.
---

### **c) Un management**

Pour que le projet soit conclu rapidement, la direction de l'entreprise doit être très impliquée et très engagée afin d'éviter toute bureaucratie ou ingérence politique.

Le management RAD doit être actif et dynamique pour réduire les risques d'allongement des délais des cycles de développement, les incompréhensions et les échéances non respectées.

En outre, le management doit être rigoureux dans le choix des membres des équipes, attentif à leurs motivations, fournir des outils et bureaux adéquats.

Mais surtout le management doit être déterminé, décidé à suivre la méthodologie RAD et impliqué dans le projet pour prendre des décisions rapidement.

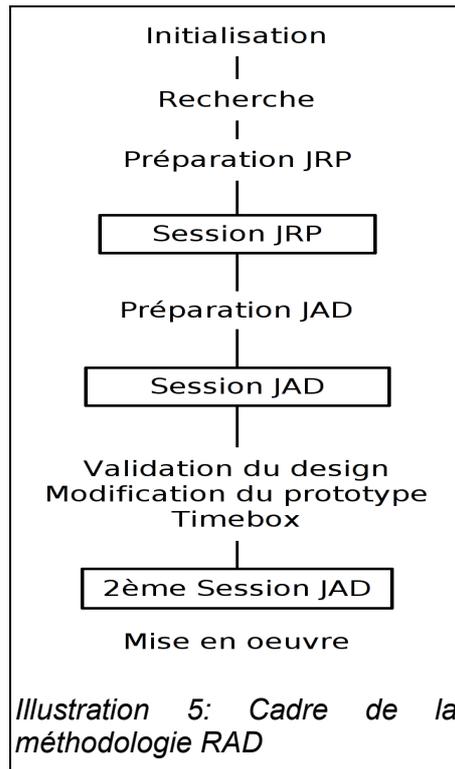
### **d) Une méthodologie**

Pour développer plus vite, mieux et moins cher le RAD s'appuie sur une méthodologie formelle et des techniques très opérationnelles :

- Des listes de tâches sont utilisées pour vérifier qu'aucune action ne soit oubliée,
- Les développements doivent être richement commentés,
- Utilisation d'un prototype évolutif qui deviendra le produit final,
- Utilisation d'AGL pour modéliser, prototyper et générer le code source
- Emploi de techniques d'aide à la définition des besoins comme JRP (Joint Requirements Planning) et JAD (Joint Application Design).
- Utilisation de la technique Time-Box pour livrer régulièrement une nouvelle version aux utilisateurs.

#### ***Cadre méthodologique du RAD***

Le RAD suit une méthodologie précise qui permet de développer un système par modifications successives et régulières du prototype de départ. Les utilisateurs sont pleinement impliqués et participent à de nombreux ateliers permettant d'orienter et de valider le développement.



### **JRP**

JRP (Joint Requirement Planning) est une méthode de travail participative qui aide à définir les besoins, lors de la phase d'expression des besoins. C'est une technique mise au point dans les années 80 par IBM.

L'objectif est de réunir au court d'une même session, courte et intense, les développeurs et les utilisateurs (décisionnaires et finaux) pour définir ensemble les besoins et les limites du système. L'avantage de la collaboration à l'inverse des interviews est de valider le but du projet. Une participation maximale et un minimum d'interruption réduisent le risque d'erreurs de définition des besoins.

La session JRP a lieu généralement en une à deux journées et ne détaille pas trop les aspects techniques. Il s'agit avant tout de définir le cadre du projet, d'impliquer tous les acteurs et de les faire se rencontrer.

### **JAD**

JAD (Joint Application Design), est une méthode de conception d'application participative utilisée à plusieurs reprises lors de la phase Conception. L'objectif est d'augmenter la qualité de la conception. Et de définir le contenu fonctionnel de l'application.

JAD permet de concilier innovation, rigueur et de lutter contre les incompréhensions entre les utilisateurs et les développeurs.

De plus, JAD favorise l'expression créative des utilisateurs qui seront amené à travailler avec le futur système. *Ils façonnent leur futur système.*

Cette expression est retranscrite dans un AGL afin de produire une description rigoureuse et les maquettes du futur système.

Enfin au cours des ateliers JAD sont réalisées les différentes versions du prototype. Chaque session se conclue par la prise en compte des modifications sur l'AGL du

prototype.

Le JAD est rendu possible par la participation d'utilisateurs finaux et par la totale implication de l'entreprise. Il permet lors des sessions « Focus » de réduire les divergences entre les acteurs et ainsi la part de risque du projet.

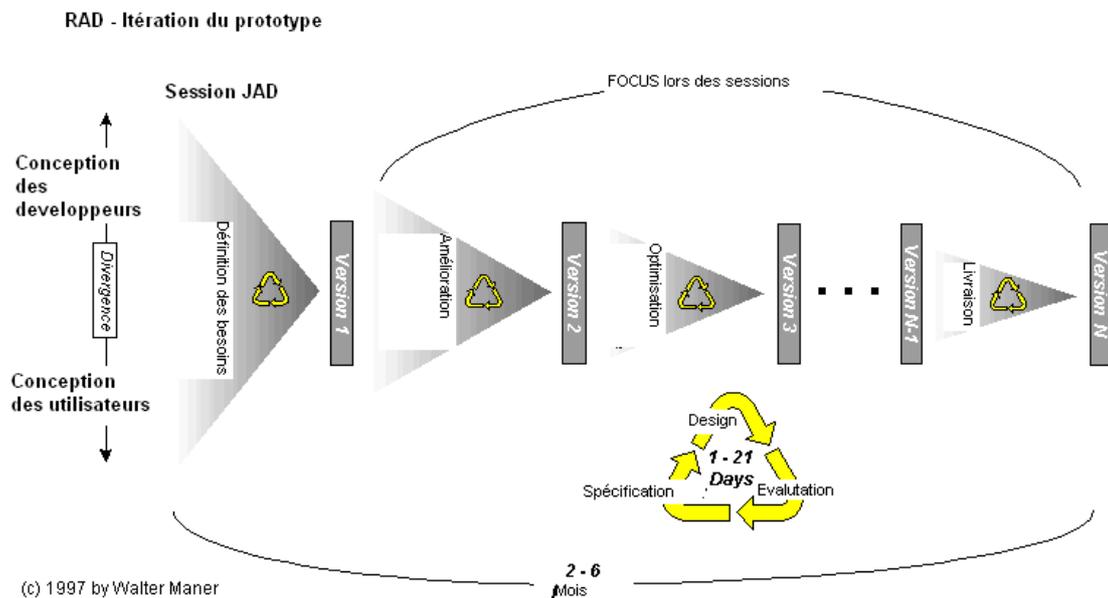


Illustration 6: JAD

Les sessions JRP et JAD ont en commun d'éviter de produire une documentation papier lourde et figée : les résultats sont présentés directement permettant une utilisation et une évolution immédiate. Ces sessions ont les caractéristiques communes :

- ✓ La participation des utilisateurs est au cœur du projet, cet engagement pousse vers une conception efficace,
- ✓ Les utilisateurs participent aux deux ateliers JRP et JAD,
- ✓ Le travail s'effectue en session à temps complet sur une ou plusieurs journées,
- ✓ Les sessions ont lieu dans des locaux en dehors du cadre de travail pour immerger dans l'équipe dans le projet,
- ✓ La session est planifiée et préparée de façon rigoureuse,
- ✓ La session est animée par l'expert RAD, une personne neutre aux enjeux de la conception,
- ✓ Le langage utilisé est celui du domaine et non celui des technologies,
- ✓ Le travail s'appuie sur des outils automatisés,
- ✓ Il ne s'agit pas de techniques statiques mais bien souple en nombre de session et en durée,
- ✓ La participation des utilisateurs favorise l'appropriation immédiate du système par les utilisateurs, réduisant l'importance d'une formation ultérieure

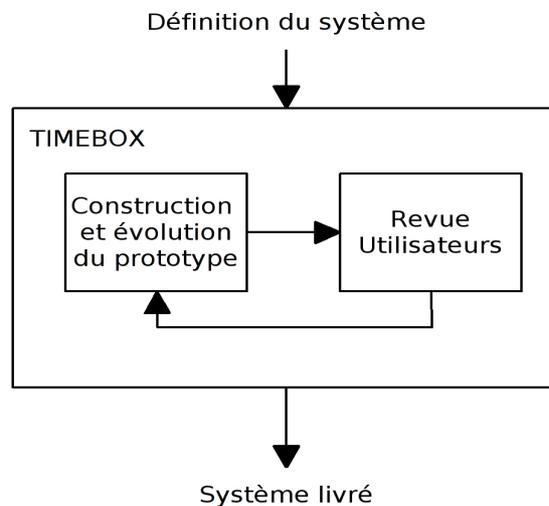
## ***Time-box***

Time-Box est une technique qui sous la forme d'une enveloppe de temps, d'une durée fixe correspond à la livraison d'une nouvelle version du prototype agrémenté de nouvelles fonctionnalités. Le time-box se comporte un peu comme une *deadline*. La date limite ne doit être dépassée sous aucun prétexte,

En effet, il arrive qu'au cours d'un projet les applications soient dans un état « presque terminé » pendant plusieurs mois. Le but de la time-box est d'éviter ce glissement. En effet, on admet souvent que 80% d'efforts doit être fait pour achever les 20% restant.

Ainsi, la time-box est considérée comme un garde-fou, d'un commun accord, les acteurs du projet RAD choisissent les fonctionnalités attendues et les répartissent dans des time-box. De cette façon, les utilisateurs finaux sont responsabilisés et choisissent les fonctions dont ils ont besoin, ceux qui vers qui seront portés tous les efforts.

Les développeurs doivent ensuite dans le temps maximum alloué à la time-box développer les fonctions attendues.



*Illustration 7: Mécanisme d'une Timebox*

De cette façon, on est sûr que le système produit contienne les fonctions désirées.

C'est également une technique pour valider et tester les développements réalisés et s'assurer que les prochains à faire ont bien un sens, afin d'éviter de se lancer dans tous les développements possibles et inimaginables.

En somme, le domaine d'application du système est défini comme une variable, fractionné en time-box.

Le technique Time-box est employé lors de la phase de Construction. Cette technique est mise en œuvre en continue lors des différentes itérations du développement.

## ***Prototypage***

Le prototypage est une technique qui permet de construire rapidement une version rudimentaire du futur système. Le prototype représente le système aux yeux des utilisateurs et des concepteurs et leur permet de s'entendre et de visualiser ses interactions et son utilisation.

Dans la méthode RAD, le prototype est évolutif et se transforme après à chaque itération pour devenir le produit final.

### ***Règles d'or de la méthode RAD***

Voici les 10 règles d'or de la méthode RAD :

1. Confier l'expression des besoins aux utilisateurs
2. Organiser l'expression de besoins
3. Introduire une dimension temporelle dans l'expression des besoins
4. Ajuster les besoins
5. Raccourcir les circuits de décision
6. Structurer les problèmes selon la structure de décision
7. Utiliser les méthodes existantes
8. Travailler en session participative
9. Anticiper
10. Utiliser des outils performants

### ***2.4.Cycle de vie RAD***

Le cycle de vie de RAD assure que le système créé est bien celui que les utilisateurs attendent.

Au cours de chaque phase les utilisateurs sont consultés et font un retour d'information aux concepteurs.

#### **a) Initialisation**

La phase d'initialisation a pour but de faire entrer ou non le projet dans un processus RAD. Elle rassemble les ressources informationnelles et humaines nécessaires tout au long du projet RAD.

Cette étape est pilotée par le chef de projet informatique (CPI) et le chef de projet utilisateur (CPU) qui vont travailler ensemble tout au long du projet.

Les chefs de projet déterminent les caractéristiques principales du projet et son contexte. Afin de dresser le profil du projet et faire une première évaluation des risques.

Le CPU décrit le projet du point de vue du gestionnaire et en situe les enjeux pour l'entreprise. Il explicite les objectifs de gestion et les objectifs d'organisation, identifie les services concernés et repère les utilisateurs qui feront partie de l'équipe projet.

Le CPI situe le projet du point de vue informatique, il positionne le futur système dans l'ensemble du SI, détaille les applications qui seront amener à communiquer et prévoit les interactions possibles avec d'autres projets en cours.

L'expert RAD analyse le projet et cherche à établir un contexte favorisant le succès du projet en RAD.

- Le domaine du projet est-il stable ?
- Les chefs de projet vont-ils pouvoir travailler ensemble ? Sont-ils suffisamment disponibles et impliqués ?
- Les utilisateurs choisis ont-ils le pouvoir de décision nécessaire ? Peuvent-ils modifier les règles de gestion ou s'engager sur une nouvelle organisation ?
- Les utilisateurs identifiés sont-ils assez nombreux ?
- Les utilisateurs ont-ils une bonne connaissance des règles de gestion ?

Cette étape doit être brève, elle s'achève sur la constitution de l'équipe projet et la préparation de la session JRP.

## **b) Expression des besoins**

L'étape d'expression des besoins est relativement courte et repose principalement sur les utilisateurs et la session JRP où sera déterminé le domaine d'application du projet, sa portée, ses limites, ses fonctions essentielles.

L'étape s'achève sur la production d'un dossier d'expression des besoins présenté par thèmes. Ce document contient la description du système existant, les critiques et les demandes d'évolution ou de fonctions nouvelles. Mais il ne s'agit pas d'un cahier des charges exhaustif puisque l'expression des besoins se poursuit et s'affine dans les étapes Conception et Construction.

Chaque participant du JRP expose le fonctionnement du système existant et ses attentes du futur système. De cette façon, est construite une première image des fonctions attendues qui comprend :

- Le dictionnaire de données
- Le dictionnaire des règles de gestion
- La modélisation du domaine sous la forme de diagrammes (modèle de flux, de processus).

La durée de cette étape ne doit pas dépasser 30 jours (environ 1 à 4 semaines).

## **c) Conception**

L'étape de Conception s'appuie sur la représentation du système existant et la formalisation des besoins exprimés. La conception s'effectue de façon descendante: elle porte d'abord sur l'ensemble du domaine, puis chaque fonction est précisée. Elle est organisée autour de l'utilisation de la technique du JAD.

Cette étape conduit à une description du futur système, telle que l'on puisse planifier les cycles de construction des prototypes, chacun couvrant une liste de fonctions identifiées.

L'étape de conception comprend deux phases :

- La phase de JAD1 animé par le CPU donne une vue modélisée du futur système d'information organisationnel.
- La phase de JAD2 recense et décrit les fonctions du futur système et est animé par le CPI et prépare directement la construction des prototypes.

### ***La phase JAD1***

Cette phase a pour but d'établir, en concertation avec les utilisateurs, le fonctionnement conceptuel et organisationnel du futur système et d'effectuer les choix d'architecture technique.

L'atelier conduit à une conception générale de l'application grâce à la réalisation du modèle de données, des flux, de l'organisation et de l'architecture technique.

Cette session permet à la fois d'affiner la conception et de créer un consensus dans l'équipe des utilisateurs.

### ***La phase JAD2***

La phase JAD2 a pour but d'établir, en concertation avec les utilisateurs, la liste des fonctions logiques, de construire les maquettes et de déterminer la répartition des fonctions dans chaque time-box.

Elle se déroule en deux parties :

- La première partie est consacrée à consolider, modifier, affiner la solution proposée. dans la mesure du possible, les modifications sont effectuées en séance de façon à clore la session avec des maquettes acceptées par tous.
- La deuxième partie planifie le développement de l'application, c'est à dire l'étape Construction qui est soumise à deux règles d'organisation. Elle doit s'effectuer dans un délai volontairement réduit et la taille de l'équipe de prototypeurs ne doit pas dépasser trois ou quatre personnes, afin de minimiser la charge de coordination.

La phase de Conception s'achève par la constitution d'un dossier de conception présenté par processus. Il contient une description à deux niveaux du futur système. Celui-ci est d'abord modélisé à travers ses flux et ses données et ensuite par une vision externe de l'application avec des écrans et des états. Le dossier comprend également une planification de l'étape suivante en énumérant les time-box. Mais il ne s'agit pas non plus d'un cahier des charges exhaustif puisque la phase suivante, Construction, sera entrecoupée de sessions d'affinements et de conception par les utilisateurs.

## **d) Construction**

L'objectif de l'étape est de développer l'application dans un délai limité. Sa durée est celle de la somme des time-box.

Par conséquent cette étape comprend autant de phases qu'il y a de cycle de prototypage. Elle fusionne les étapes classiques de spécification détaillée, de réalisation (codage), de tests unitaires et de tests d'intégration, ainsi que la plus grande partie des tests de cheminement fonctionnel qui constituent la recette initiale de l'application

Le premier prototype joue un rôle important. Il montre aux utilisateurs que le projet avance. Il les rassure sur le contenu de l'application.

Le respect des time-box doit être la préoccupation de tous.

Si des développements sont faits en parallèle, les différents prototypes sont assemblés à cette étape.

La conclusion de cette étape est la fourniture de l'application sous la forme d'un prototype complet et validé.

### e) Mise en œuvre

L'étape Mise en œuvre correspond à la livraison globale du nouveau système et au bilan du projet. L'équipe doit particulièrement être attentive à la mémorisation du savoir faire, des objets utilisés lors du projet afin de capitaliser le savoir-faire RAD.

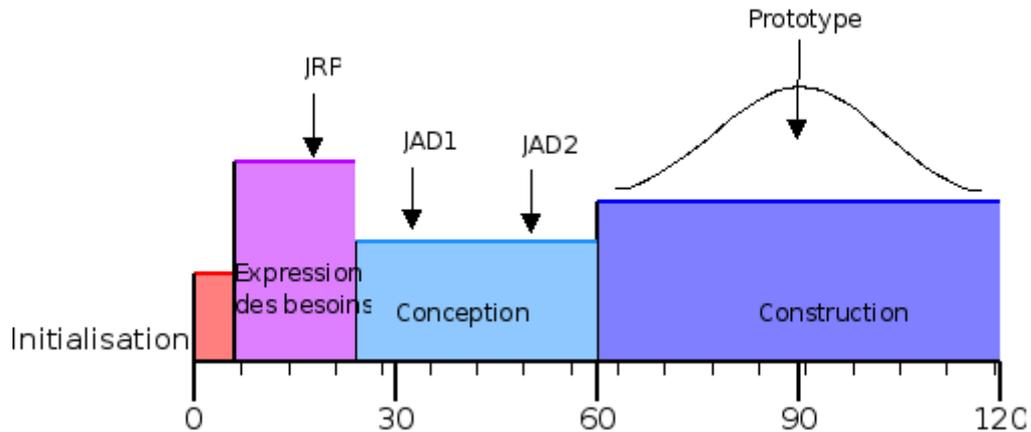


Illustration 8: Déroulement d'un projet RD

Cette première partie, a permis de présenter la méthode RAD telle qu'elle a été mise au point à la fin des années 80. Cette méthode tire profit des techniques et outils qui la précèdent, il est intéressant de savoir si plus de vingt ans après, si cette méthode est toujours appliquée et d'en faire le bilan.

### 3. La place de la méthode RAD dans la gestion de projet

La méthode RAD est un cadre de travail qui permet, s'il est suivi, de réaliser un système rapidement qui satisfasse les utilisateurs, diminue la maintenance applicative et réduit les coûts de formation des utilisateurs.

Cette méthode peut se combiner avec d'autres méthodes ou techniques, tel Merise, UML, la programmation objet et entretient un lien fort avec les méthodes agiles mais souffre de quelques limites et critiques.

#### 3.1. RAD, une méthode AGILE ?

Comme le RAD, les méthodes agiles sont une réponse au fait que le client est souvent dans l'incapacité de définir ses besoins de manière exhaustive dès le début du projet.

La notion de méthode agile est née à travers un manifeste signé par 17 personnalités (parmi lesquelles Ward Cunningham, l'inventeur du Wiki), créateurs de méthodes ou dirigeants de sociétés qui prônent quatre valeurs fondamentales :

- individus et interactions plutôt que processus et outils
- développement logiciel plutôt que documentation exhaustive
- collaboration avec le client plutôt que négociation contractuelle
- ouverture au changement plutôt que suivi d'un plan rigide

Plusieurs méthodes se sont regroupées sous l'intitulé « Méthodes Agiles » et partagent ses quatre valeurs.

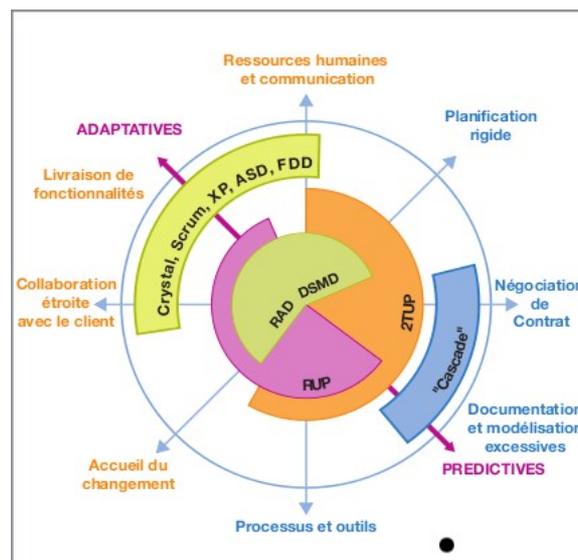


Illustration 9: Position de RAD face aux méthodes Agiles

RAD partage de nombreuses valeurs avec les méthodes agiles :

- Il met en avant les ressources humaines et la communication par l'existence d'un animateur,
- Il privilégie la production d'applications fonctionnelles par rapport à l'écriture massive de documentation,
- Il requiert une collaboration étroite avec le client (utilisateur).

Mais, RAD n'est pas une méthode flexible par rapport au changement. Des modifications peuvent intervenir à l'intérieur du projet (lors des sessions JAD) mais la planification du RAD doit être rigide et rigoureuse pour se dérouler en 120 jours maximum.

Le RAD présente donc trois des quatre caractéristiques des méthodes agiles. Apparu chronologiquement avant les méthodes Agiles, il en préfigure cependant certains principes mais ne peut pas être considéré comme une méthode Agile.

### **3.2.RAD, une méthode souple**

RAD est une méthode de projet pragmatique et souple qui peut si on le souhaite être combiner avec d'autres techniques telles que :

- un langage formel de modélisation comme UML ou Merise,
- la mise en œuvre de Extrem Programming lors la phase de Construction,
- la programmation orientée objets.

#### **a) RAD – UML – MERISE**

RAD est une méthode de gestion de projet souple qui n'impose pas de langage de modélisation précis.

De cette façon, l'équipe RAD utilise la technique de modélisation qui lui est la plus familière. RAD peut s'adapter aussi bien à Merise pour son formalisme qu'à UML.

Ces techniques s'appliqueront aux parties hautes du cycle RAD, les phases d'Analyse des besoins et de Conception.

RAD	Merise	UML
Analyse des besoins	modèle de Contexte (MC), modèle Conceptuel de Communication (MCC)	Diagramme des cas d'utilisation Diagramme d'état transition Diagramme de séquence Diagramme d'activité Diagramme de classe Diagramme de déploiement Diagramme de paquetages
Conception	Modèle Conceptuel de	Diagramme de composant

	Traitement (MCT) Modèle Conceptuel de Données (MCD), modèle Organisationnel de Traitements (MOT)	mise à jour des précédents diagrammes
--	---	---------------------------------------

## b) RAD – Extrem Programming (XP)

XP, la méthode agile la plus en vogue, peut être adaptée à la phase « Construction » et devenir un sous-ensemble du projet RAD.

Une de ses originalités réside dans l'approche de la planification qui se matérialise sous la forme d'un jeu intitulé « planning game » qui implique simultanément les utilisateurs et les développeurs. Le jeu consiste à laisser les utilisateurs classer les demandes et fonctions dont ils ont besoin un peu comme avec les time-box.

XP se déroule par un cycle itératif d'au maximum une semaine, comme l'enveloppe de temps d'un time-box.

XP préconise aussi des techniques particulières liées à la production du code comme la programmation en binôme, l'appropriation collective, le refactoring et l'intégration continue. La méthode RAD utilise tout ou partie de ces techniques mais limite généralement la programmation en binôme aux parties les plus stratégiques de l'application.

Fonder sur un cycle court, un feedback régulier et une évolution continue de la solution pour ces raisons, XP peut avantageusement remplacer la phase classique de Construction de la méthode RAD.

## c) RAD – la programmation Objet

Un facteur essentiel permettant la rapidité d'un projet RAD est l'emploi de technique de « réutilisabilité ». Cette technique peut être mise en œuvre grâce à l'emploi d'objet déjà créé ainsi que par les techniques de généralisation et de spécialisation qui sont issues de la technologie objet.

Par exemple, un projet RAD fonctionne souvent avec une base de données. Par conséquent, il est intéressant de manipuler des objets déjà créés pour se connecter à la base ou pour exécuter des requêtes. Dès lors, l'emploi de Framework peut être un plus pour le RAD.

La généralisation et la spécialisation consiste à s'appuyer sur des entités de référence pour hériter de leur description et à y rajouter des éléments spécifiques.

D'autres part, l'utilisation de la programmation objet se combine bien avec la méthode RAD lorsqu'UML est utilisé lors de la phase de conception.

### **3.3.Etat des lieux de RAD**

La méthode RAD est une réponse aux critiques formulées à l'encontre de la méthode traditionnelle mais elle n'est pas exempte elle-même de critiques et de limites.

#### **a) Avantages et inconvénients de la méthode**

##### ***Pour l'utilisateur :***

###### Avantages

- L'utilisateur est placé au premier rang du projet informatique. Il est parti intégrante de l'équipe de projet et participe à tous les stades de l'élaboration du système. Il définit les besoins, les complète et les affine au fur et à mesure de l'avancement du projet. Il décide si des modifications sont à apporter au système.
- L'utilisateur est acteur du résultat final.
- L'utilisateur connaît déjà le système développé, la formation se fait en parallèle au processus de développement.
- L'utilisateur peut travailler directement avec une partie du système. Ce système n'est pas encore terminé, mais est déjà doté des fonctions majeures.
- L'utilisateur reçoit dans un intervalle court une application opérationnelle.

###### Inconvénients

- Le classement des besoins par priorité et le choix entre délais et fonctionnalités n'est pas toujours facile à faire. Cela implique de l'utilisateur un pouvoir de décision et l'acceptation de compromis.
- L'implication dans un projet RAD nécessite de l'utilisateur une motivation forte et un investissement personnel supplémentaire.
- La livraison d'une version allégée du système peut entraîner frustration et insatisfaction chez l'utilisateur.
- Une application inachevée risque de le rester faute de nouvelles ressources (hommes, argent, temps).

##### ***Pour l'informaticien***

###### Avantages

- L'informaticien-développeur est plus sûr du résultat final, car il l'aura réalisé avec son client.
- L'implémentation du système est facilitée, car les prototypes sont testés tout au long du cycle de développement.
- L'informaticien voit plus vite le fruit de son travail.
- L'informaticien s'enrichit par le contact permanent qu'il a avec son client et apprend à mieux connaître le métier de ce dernier.

## Inconvénients

- L'informaticien doit faire preuve de discipline lors de la réalisation du prototype. Les besoins prioritaires doivent être traités en premier, autrement les objectifs et les délais ne pourront être atteints.
- L'informaticien doit passer d'un rôle de décideur à celui de conseiller.

## ***Pour l'entreprise***

### Avantages

- L'entreprise dispose d'un système de bonne qualité qui respecte les coûts et les délais.
- L'entreprise dispose d'un système qui répond à ses objectifs au moment nécessaire.
- Le projet RAD permet à l'entreprise d'accroître les compétences et la formation de son personnel.
- Le projet RAD rapproche les personnes de métiers différents et favorise une culture d'entreprise.
- Les risques de projets sans fin et de gouffres financiers sont minimisés.

### Inconvénients

- La documentation du système peut être insuffisante pour la maintenance et l'évolution future du système.
- La mise en œuvre de RAD nécessite un investissement important de départ: le personnel doit être formé et disposer de matériels et d'outils performants.
- L'entreprise doit recourir un consultant indépendant comme animateur/expert RAD.

## **b) Quels projets pour le RAD ?**

Tous les projets ne peuvent pas être adaptés au RAD. La méthode RAD s'adresse principalement à des systèmes de gestion, où l'utilisateur va créer des données, les stocker et les restituer sous une autre forme.

La méthode RAD n'est pas adaptée, par exemple à la création :

- d'un système d'exploitation,
- de jeux vidéo,
- d'outils automatisme industriel
- pilotes matériel

La méthode RAD se prête plus particulièrement à la création d'application type site web ou à des applications nécessaires mais avec des espérances de vie courtes. Le système produit n'évoluera pas après sa création. Il s'agit plutôt d'applications jetables.

### **c) Souci de performance des systèmes RAD**

L'une des critiques les plus importantes faite au RAD concerne la performance du système développé.

En effet, la méthode RAD permet de créer un système rapidement mais ce système risque de souffrir d'un problème de performance, de scalabilité. L'application développée ne sera pas faite de temps optimisée. Elle risque d'offrir des performances médiocres en cas de montée en charge importante (augmentation des connexions ou du volume de données).

Pour cette raison, la méthode RAD est adaptée à de petits systèmes qui ne sont pas critiques.

### **d) La méthode RAD, incompréhensions**

L'un des facteurs le plus contraignant est son manque de compréhension.

Tout d'abord la méthode RAD repose pour une part importante sur des outils de conception et de développement graphiques qui permettent de développer dynamiquement et d'automatiser de nombreuses tâches. De nombreux éditeurs d'AGL ont réduit l'idée de rapidité à la simple utilisation de leurs outils. Le marketing s'est approprié l'idée et vend des outils « rapides ». Ainsi le concept « rapide » et le label « RAD » ont été récupéré par les éditeurs de solutions qui proposent du « rapide » mais sans la méthode adaptée.

Une autre source d'incompréhension vient du fait que la méthode n'a jamais été introduite en France. Le livre de James Martin est épuisé et non traduit en français. En outre, elle n'est pas enseignée.

Cette méthode est mal comprise et dans l'esprit de beaucoup est réduite à l'utilisation d'outils graphiques et au développement d'un prototype.

### **e) Des fonctionnalités réduites**

Intrinsèquement, le système créé à partir de la méthode RAD est limité fonctionnellement. En effet pour répondre aux délais fixés et aux exigences des time-box, les fonctionnalités sont volontairement réduites. Seules les fonctionnalités nécessaires sont réalisées, les autres seront ajoutées dans une nouvelle version.

### **f) Le RAD nécessite un changement de culture**

Autre facteur limitant la compréhension et l'adoption de RAD vient du fait que la méthode

implique un changement d'organisation et de mentalité.

Pour être efficace, les acteurs de la méthode doivent être entrepreneurs, compétents et responsables. Ils doivent être en capacité de prendre des décisions rapidement qui vont impacter sur l'organisation et la gestion de l'entreprise. Par exemple, ils doivent pouvoir choisir les fonctionnalités importantes.

Or traditionnellement dans les entreprises françaises les décisions sont concentrées au sommet de la hiérarchie. Ce type de contrôle de la décision convient plus à une méthode de projet en cascade (GO, NO-GO) qu'à la méthode RAD.

Pour tirer réellement profit de la méthode RAD, un changement de culture d'entreprise doit être opéré, les utilisateurs décisionnaires doivent être impliqués dans le projet et être acteurs de son déroulement.

## 4. Conclusion

Si l'entreprise souhaite être compétitive, l'acquisition d'une méthode de gestion de projet est indispensable à la construction et à l'évolution d'un système d'information performant. Aujourd'hui de nombreuses méthodes existent, allant de la plus rigide à la plus agile, d'un déroulement en séquence à un déroulement en spirale.

La méthode RAD est une méthode pragmatique et non intuitive. Elle n'est pas adaptée à tous types de projet. Cependant c'est une méthode intéressante et souple qui permet si elle est bien mise en œuvre de produire un système rapidement de qualité.

L'utilisation de l'adjectif rapide a eu comme effet de dénaturer ses principes et sa rigueur. L'industrie du Logiciel et son marketing se sont emparés de l'idée de « développement rapide à moindre coût » pour proposer une pléthore de solutions loin de l'esprit d'origine de la méthode. La méthode RAD est devenue alors pour certains Rough and Dirty (brouillon et sale).

Néanmoins cette méthode RAD ne peut pas être réduite simplement à l'utilisation d'AGL ou au développement d'un prototype. Elle repose sur une vraie méthodologie rigoureuse qui se fonde sur une communication, une implication et un changement de culture qui sont parfois difficiles à mettre en place dans une entreprise mais évitent le bricolage.

Elle connaît même un certain succès puisque elle a donné naissance à la méthode DSDM (Dynamic Systems Development Method) ou a permis par exemple de réaliser des projets importants et critiques comme la gestion des trains de la plus importante compagnie de chemin de fer anglaise, EWS (English, Welsh & Scottish Railway).

## 5. Annexe 1 – Outils RAD

- PC Soft Windev
- Borland C++ Builder
- Borland Delphi
- Microsoft Visual Basic
- Microsoft Visual Foxpro
- Sybase PowerBuilder
- Microsoft Visual Studio.NET 2003
- Sun Java Studio Creator
- BEA Web Logic Workshop 8.1
- Borland C# Builder
- IBM WebSphere Studio Application Developer 5.1.2

## 6. Bibliographie

- AGARWAL Ritu, PRASAD Jayesh, TANNIRU Mohan, LYNCH John, Risks of Rapid Applications Development, ACM 2000, 12p,
- BOEHM Barry, Making RAD work for your project, IEEE Software Mars 1999, 3p,
- CARD David N., IEEE Software Septembre 1995, 2p,
- CARMEL Erran, IEEE Software Septembre 1995, 2p,
- CHOUKROUN Marc, Techniques de l'ingénieur, H3240 Février 200, 14p,
- CROS Franck, Dr LAWRENCE Elaine, Synergies In Applying RAD to Support e-Business Project Management, IEEE Software Avril 2002, 8p,
- GORDON Scott, BIEMAN James, Rapid Prototyping: Leasson Learned, Janvier 1995, 11p,
- HOWARD Alan, Rapid Application Development: Rough and Dirty or Value-for-Money Engineering, IEEE Software Septembre 2002, 3p,
- HUGUES Jean, LEBLANC Bernard, MORLEY Chantal, *RAD une méthode pour développer plus vite*, Paris, InterEditions, 1996, 217 p. ISBN 2-7296-0636-X,
- MILLINGTON Don et STAPLETON Jennifer, IEEE Software Septembre 1995, 2p,
- REILLY John P., IEEE Software Septembre 1995, 2p,
- VICKOFF Jean-Pierre, RAD contexte d'émergence, [www.rad.fr](http://www.rad.fr) 2000, 23p,
- VICKOFF Jean-Pierre, RAD méthode de développement rapide d'applications, [www.rad.fr](http://www.rad.fr) 2000, 32p,

## 7. Glossaire

CPI : Chef de projet informatique

CPU : Chef de projet utilisateur

JAD : Joint Application Design

JPR : Joint Planning Requirement

MOA : Maître d'ouvrage

MOE : Maître d'oeuvre

NTIC : Nouvelle technologie de l'information et de la communication

RAD : Rapid Application Development

SI : Système d'Information

SWAT : Skilled with advanced tools

UML : United Modeling Language