

Processus RAD2

Ingénierie du logiciel



Adaptation du processus
Spécialisation du processus
Stratification du processus

Jean-Pierre Vickoff, RAD.fr

RAD2

Processus

Adaptation du processus

Spécialisation du processus

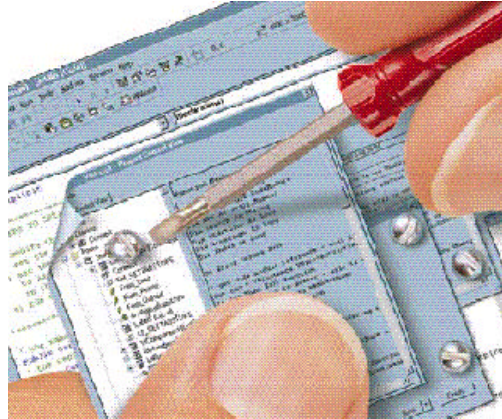
- Merise
- Flux
- UML

Stratification du processus

Jean-Pierre Vickoff
RAD.fr

SOMMAIRE DU DOCUMENT

PROCESSUS RAD2	1
1. ADAPTATION DU PROCESSUS D'INGÉNIERIE	5
1.1. Processus générique adaptable	5
1.2. Adaptation aux contraintes du projet	5
1.3. Adaptation aux exigences applicatives	5
1.4. Adaptation aux formes de modélisation	5
1.5. Prévention des défauts	6
1.6. Gestion des changements du processus	6
2. SPÉCIALISATION DU PROCESSUS RAD 2	7
2.1. Phase INITIALISATION	7
2.2. Phase CADRAGE	15
2.3. Phase DESIGN	20
2.4. Phase CONSTRUCTION	27
2.5. Phase FINALISATION	39
2.6. Processus optimisé « petit projet »	41
3. STRATIFICATION DU CHOIX DE SOLUTION	42
3.1. Couches externalisables	42
3.2. Exemples d'éléments externalisables	43
3.3. Processus général « externalisation »	44
3.4. Processus spécialisé « progiciel »	45
4. BIBLIOGRAPHIES ET RÉFÉRENCES	46
4.1. Bibliographie RAD, Conduite de projet	46
4.2. Bibliographie Objet	46
4.3. Bibliographie Management, Qualité	46
4.4. Divers documents, normes, standards	47
4.5. Principaux WEB et contributeurs	48
4.6. Index	49
4.7. Figures	51
4.8. Tableaux et listes	51



Processus RAD 2

**Organisation des développements,
conduite de projets,
ingénierie applicative,
RAD2.**

« Second précepte : ... diviser chacune des difficultés que j'examinerois, en autant de parcelles qu'il se pourroit, et qu'il seroit requis pour les mieux résoudre. »

René Descartes, *Discours de la méthode*

1. Adaptation du processus d'ingénierie

Un processus d'ingénierie « logiciel » est un ensemble de **tâches formelles** à accomplir pour **sécuriser** l'obtention d'une application en **conformité** avec les exigences des utilisateurs, dans le respect des **contraintes** imposées au projet.

Il est nécessaire d'**adapter** le processus d'ingénierie aux spécificités du projet et de l'application.

Le processus proposé dans le chapitre suivant est **RAD2**.

1.1. Processus générique adaptable

CMM : La *Gestion logiciel intégrée* vise à répertorier les activités d'ingénierie et de gestion logiciel sous forme d'un processus défini et cohérent adapté à partir du processus standard de l'organisation et des acquis processus associés (voir le secteur clé *Définition du processus de l'organisation*). Cette adaptation est fonction du contexte organisationnel et des exigences techniques du projet, tels qu'ils sont décrits dans le secteur clé *Ingénierie de produits logiciel*. La *Gestion logiciel intégrée* est le fruit des activités de *Planification de projet* et de *Suivi et supervision de projet* du niveau 2.

Le processus générique RAD 2 couvre certainement plus de 90% des activités de conduite d'un projet de développement d'applications de gestion. Néanmoins, aucun processus, aussi générique soit-il, ne saurait exhaustivement couvrir l'ensemble des cas de figure et des imprévus du domaine de la conduite de projet et se substituer ainsi à l'expérience et du professionnel. Pour adapter le processus d'ingénierie aux contraintes du projet et aux spécificités de l'application, il faut en premier lieu obtenir¹ un processus générique sous une forme utilisable par un outil dit « de gestion de projet » utilisable (Word, Excel, MS Projet, etc.).

1.2. Adaptation aux contraintes du projet

Le processus générique doit être modifié en fonction des contraintes s'appliquant au projet. Cette tâche s'instrumente à l'aide d'un outil de planification et nécessite :

- de choisir le niveau de décomposition approprié à la taille du projet ;
- de supprimer les paragraphes inutiles ou trop lourds compte tenu du cas particulier traité ;
- d'insérer les actions non prévues mais s'avérant nécessaires en fonction du contexte ou d'impératifs particuliers.

1.3. Adaptation aux exigences applicatives

Le processus générique doit aussi être modifié en fonction des exigences spécifiques de l'application. Cette tâche s'instrumente à l'aide d'un outil de planification et nécessite d'introduire les fonctionnalités spécifiques à l'application à chaque phase du projet.

1.4. Adaptation aux formes de modélisation

En fonction du projet, une forme de modélisation optimale sera choisie. Le processus inclut trois formes de modélisation : **MERISE** (version 1 ou 2, allégée), les **FLUX** (DFD) et **UML**. Ces formes de modélisation sont décrites dans le *chapitre 18*.

¹ Le processus RAD2 est téléchargeable à partir de www.RAD.fr

Les références aux représentations propres à chaque forme de modélisation sont intégrées au processus générique sous la forme de recommandations encadrées comme suit :

MERISE - Les modèles

FLUX - Les niveaux de DFD

UML - Les modèles ou diagrammes

1.5. Prévention des défauts

CMM : La Prévention des défauts vise à identifier et à éliminer les causes de défauts pouvant avoir altéré l'efficacité du processus (obsolescence, inadaptation).

A la fin des projets, le groupe de projet analyse les défauts, identifie leurs causes et modifie en conséquence le processus défini, tel que décrit dans le secteur clé *Gestion logiciel intégrée*.

Les changements de processus de portée générale sont également effectués dans les autres projets, selon la procédure décrite dans le secteur clé *Gestion des changements du processus*.

1.6. Gestion des changements du processus

La *Gestion des changements du processus* vise l'amélioration continue du processus de développement.

A la fin des projets, la *Gestion des changements du processus* permet d'améliorer la qualité logiciel, d'augmenter la productivité et de diminuer la durée du cycle de développement des produits.

La gestion des changements du processus s'applique à l'ensemble de l'organisation des développements informatiques.

La gestion des changements du processus englobe les améliorations progressives découlant de la *Prévention des défauts* et les améliorations novatrices produites par la *Gestion des changements technologiques*.

2. Spécialisation du Processus RAD 2

RAD : méthode de réalisation sous contraintes (temporelles, budgétaires, fonctionnelles et techniques) d'une application en parfaite adéquation avec les exigences de l'organisation et totalement approuvée par ses utilisateurs réels.

La méthode est un guide des pratiques fondamentales.

RAD 2 : processus d'ingénierie de produit logiciel, générique et adaptable à la typologie du projet comme aux formes de modélisation choisies.

Le processus est un répertoire des tâches indispensables.

Les fondements du **processus RAD2** sont les principes de la méthode RAD. Le processus supporte 3 formes de modélisation : **MERISE** (version 2, allégée), les **FLUX** (DFD) et la notation Objet **UML**.

CMM : L'Ingénierie de produits logiciel vise à exécuter systématiquement un processus d'ingénierie formellement défini. Il intègre l'ensemble des activités d'ingénierie permettant de produire efficacement des produits logiciel cohérents et corrects. Ce secteur clé décrit les activités techniques du projet, tels que l'analyse des exigences, la conception, la programmation et les tests.

La structure du processus RAD2 succinctement décrite est extraite d'un rapport du Gartner Group intitulé : **Réingénierie du Développement d'Applications [Vickoff 1999]**.

2.1. Phase INITIALISATION

2.1.1. Recrutement animateur et coordonnateurs

Un projet RAD commence lorsqu'une direction exige la réussite d'un projet informatique et engage un animateur spécialisé en communication RAD. Cet animateur doit être dépendant de la direction générale ou d'une instance neutre en regard des principaux acteurs engagés dans le projet.

Dans un projet important, il est possible d'utiliser les services d'un animateur interne non informaticien. Le profil de base peut être celui d'un ancien commercial rompu à la négociation et à l'organisation de réunions. Une formation complémentaire aux techniques de modélisation lui est dispensée. Le but n'est pas d'en faire un expert en modélisation, mais un vérificateur sachant lire un modèle. Lors d'un projet d'une dizaine d'années-homme, sa présence est suffisamment significative pour justifier un engagement à temps complet sur le projet. Il doit donc accepter de quitter momentanément sa fonction.

2.1.2. Présentation de la méthode RAD

La mission de l'animateur est en premier lieu d'informer les participants des contraintes inhérentes aux avantages offerts par la méthode RAD.

La direction nomme ensuite le coordonnateur des ressources utilisateur et le coordonnateur des ressources informatiques.

L'animateur détaille leurs engagements réciproques sous une forme contractuelle. Il organise avec eux le plan de communication du projet.

2.1.3. Immersion du groupe de coordination

Cette étape consiste en une immersion de l'animateur et des coordonnateurs dans le domaine fonctionnel du projet afin d'en délimiter le périmètre, puis ensuite d'en

évaluer globalement les coûts. Le produit de cette étape se matérialise sous la forme d'un diagramme de flux (DFD) où l'on peut observer les grands acteurs et les grands traitements ainsi que les principaux flux d'information. La production de cette phase se concrétise dans un document présentant le futur système sous la forme de 5 visions :

VISION - STRATEGIQUE

C'est un descriptif des ambitions de la future application. A ce niveau de granularité, la perception est qualifiée de « VISION ». Cette étude globale permet d'optimiser le périmètre fonctionnel des exigences couvertes.

VISION - FONCTIONNELLE

Au niveau de la conception, et particulièrement pour le décisionnel ou le stratégique, les modèles de traitement de Merise sont avantageusement remplacés par le principe de la hiérarchie des fonctions

VISION - TECHNOLOGIQUE

L'intégrateur de technologie prévoit les technologies innovantes et l'architecture technique. Il garantit la faisabilité de ces solutions.

VISION - ORGANISATIONNELLE

Les procédures organisationnelles doivent être clarifiées avant d'entrer dans les aspects fonctionnels de détail. Prévoir les impacts organisationnels et en informer la direction.

VISION - CONTRAINTES

Déterminer les ressources, les délais, le budget. Conclure par la réalisation de l'étude d'opportunité qui fixe le domaine, ses limites et les intervenants motivés, créer le groupe de projet RAD et faire valider par le comité de pilotage du projet.

2.1.4. Modélisation globale du périmètre

ACTEURS, CIRCUITS, FONCTIONS, ACTIONS

Cette étape est nécessaire à l'identification des utilisateurs participant aux sessions de travail et au déploiement des outils de communication.

MERISE - Modèle Conceptuel des Communications.

FLUX - DFD de niveau 0 (contexte).

UML - suivant le besoin, utiliser les diagrammes :

- ❖ **de cas d'utilisation relatif au périmètre (global),**
- ❖ **de collaboration associé au scénario global,**
- ❖ **de séquence du scénario,**
- ❖ **d'activité du scénario.**

La technique des « use case » ou cas d'utilisation popularisée par *Ivar Jacobson* est un moyen pratique et particulièrement accessible aux utilisateurs désireux de formaliser dans des scénarios leur activité.

Sans être totalement exhaustifs dans le détail, les scénarios doivent être représentatifs de toutes les situations à considérer.

Au-delà d'une aide intéressante à la validation des modèles, les « use case » définissent les cheminements fonctionnels et sont particulièrement utiles aux informaticiens. Ils facilitent les validations lors des sessions de travail, des Focus et lors de la recette générale avant déploiement. Cette dernière étape représente souvent une tâche ardue dans les grands projets. La technique des « use case » se révèle alors un investissement intéressant.

La technique des « use case » est un complément utile à la modélisation classique pour réduire le risque dans des situations fonctionnellement complexes. Son usage est conseillé.

2.1.5. Définition plan des communications « projet »

Commencer par un recensement des acteurs (entités organisationnelles et collaborateurs).

A. Structure de l'organisation

Nom de la structure, responsables, coordonnées, mission de la structure, implication dans le projet.

B. Acteur

Nom, prénom adresse postale, téléphonique, fax, mail, titre ou qualité, rattachement à la structure de l'organisation, rôle dans la structure.

En fonction des besoins du projet, proposer l'organisation des groupes de travail.

C. Identification des groupes de travail

- Complet (réunion de lancement, Focus).
- Dirigeant (lancement, partie organisationnelle, etc.).
- Verticaux (par phase) (Cadrage, Design, Construction).
- Horizontaux (par thèmes, parallélisation).

Dans chaque groupe réaliser une identification des rôles (relation acteur-groupe).

D. Relation acteur / groupe

- Rôle dans le groupe (animateur, rapporteur, membre, invité).
- Type intervention (pour action, pour information).
- Date de début, date de fin, actif / inactif, commentaire.

ACTEURS SIGNIFIANTS, DECIDEURS, OBSERVATEURS, EXPERTS

La nature même des nouvelles applications implique une participation accrue de la maîtrise d'ouvrage, donc sa formation préalable.

Il n'est pas possible d'imaginer aujourd'hui des utilisateurs significatifs ne portant aucun intérêt à une application qu'ils devront utiliser parfois durant plusieurs années. Il est fondamental, dans l'intérêt des utilisateurs, qu'ils définissent eux-mêmes le détail de leur outil de travail.

La composition des groupes respecte un équilibre des représentations. Les participants sont représentatifs de l'ensemble des clients de l'application et sont impliqués à des niveaux de hiérarchie dotés de pouvoirs suffisants à la prise de décisions.

◇ **Groupe de décision et de validation**

Cette structure n'est pas indispensable pour les petits projets où les intervenants disposent de pouvoirs suffisants. Dans les projets plus conséquents, une structure classique telle qu'un comité de pilotage se réunira à chaque phase pour décider de la poursuite du projet ou de ses grandes orientations.

◇ **Groupes de travail utilisateurs**

Les groupes de travail utilisateurs sont des structures de production qui ont pour rôle :

- d'intégrer à l'équipe de projet les connaissances pratiques nécessaires à la bonne conception du projet ;
- d'assurer le relais entre le SWAT et l'ensemble des utilisateurs ;
- de permettre aux utilisateurs de s'approprier le projet qui leur est destiné.

Les groupes de travail utilisateurs constituent la base privilégiée des entretiens RAD.

Ces groupes de travail ne doivent pas se limiter à une participation des niveaux supérieurs de la hiérarchie. Le risque est de passer à côté des vrais problèmes en obtenant une vision uniquement théorique de ce qui devrait être fait.

Lorsque des divergences marquées apparaissent entre les visions des divers groupes, une réingénierie des procédés est en général nécessaire.

Pendant la phase de Cadrage fonctionnel, on alterne des groupes réduits lors des réunions de spécification et des groupes plus larges lors des réunions de validation.

Durant la phase de réalisation on emploie le même principe.

Lors du prototypage, la séance implique un ou deux utilisateurs et on élargit au groupe lors des réunions de validation.

2.1.6. Entretien propriétaire (maîtrise d'ouvrage)

VALIDATION DES VISIONS GLOBALES

Suite à la définition d'une vision globale du projet et de son plan de communication, un entretien initial a lieu avec le propriétaire :

- la stratégie de la direction est précisée ;
- l'ENGAGEMENT pour l'emploi du RAD est obtenu ;
- le but du projet, ses limites, ses objectifs sont fixés.

L'engagement RAD est un point fondamental. L'animateur s'assure ainsi de la reconnaissance des objectifs par les maîtrises d'œuvre et d'ouvrage. Il rédige et fait approuver le protocole du projet. Si tous les dirigeants signent, le principe du développement RAD est alors acquis.






Le plus important dans le fait de donner un nom à un projet est certainement le processus qui conduit à son choix en organisant une large concertation préalable à laquelle participera le plus grand nombre d'acteurs. Cette première action qui pourrait sembler insignifiante révèle en fait le style du management.

LETTRÉ D'ENGAGEMENT DE LA MAÎTRISE D'OUVRAGE

Compte tenu des exigences du plan de communication, un engagement nominatif des utilisateurs est planifié. Il est quantifié sur la base de l'évaluation globale du projet. Le coordonnateur de la maîtrise d'ouvrage le signe.

Tableau croisé des participants / groupes de travail

Légende :

	participe à plein temps au groupe de travail
	participe ponctuellement au groupe de travail
	et reçoit les comptes-rendus des réunions
	chef de projet utilisateur ou informatique
	présence requise à toutes les réunions

Groupes d'intervenants	Comité de projet	Thème 1	Thème 2	Thème 3
	Utilisateur A	○	✓	✓
Utilisateur B				○
Utilisateur C	✓		○	
Utilisateur D		✓		
Utilisateur E	✓			○
Utilisateur F		○		
Utilisateur G		✓	✓	
Informaticien 1	○	✓	✓	✓
Informaticien 2	✓	✓	✓	✓
Informaticien 3				✓
Informaticien 4		✓	✓	
Informaticien 5				
Total noyau dur	6	12	11	10
Groupe élargi	11	12	11	12

Figure 1. — Participation aux groupes de travail

ENVIRONNEMENT DE RÉUNION

L'animateur s'assure de la disponibilité de l'environnement matériel de réunion et de développement (salle équipée, vidéoprojecteur électronique, micros puissants, logiciels, réseau indépendant si nécessaire).

- Les intervenants du projet sont recensés et classifiés dans des groupes de diffusion (figure 1).
- Les moyens et les circuits de communication sont identifiés.
- Les informations sont formalisées dans des documents normalisés qui respectent les recommandations ISO.
- L'émission des convocations et des invitations est ordonnancée.

Cet ensemble constitue le plan de communication « projet ».

L'emploi d'un courrier électronique (MS Mail, E-Mail, Lotus Note, etc.) est un minimum pour l'efficacité requise du RAD.

La plupart des projets qui ont échoué ne disposaient pas d'une messagerie moderne. Cette même messagerie est un composant stratégique du *groupware* et du *workflow*. La

disponibilité, sur le réseau, d'un outil de présentation classique ou Intranet représente un autre pas significatif vers une communication intelligente et spécialisée.

Tous les entretiens de groupe et toutes les réunions officielles sont organisés suivant un mode opératoire précis. Ce mode opératoire est une des techniques incontournables du RAD, il consiste à structurer chaque réunion en trois étapes :

- ◇ 1. Pré-session (planification thèmes et minutage)
- ◇ 2. Session (clôture et ouverture d' ACTIONS)
- ◇ 3. Post-session (analyse des impacts)

Le mode opératoire des sessions est détaillé dans le chapitre 5.

2.1.7. La réunion « préparatoire »

Cette réunion (qui peut se confondre avec la réunion de lancement pour les petits projets) regroupe :

- l'animateur,
- le rapporteur,
- les responsables du projet,
- des utilisateurs de base (une dizaine au maximum),
- des experts des domaines concernés,
- éventuellement des observateurs (ou des futurs animateurs).

La réunion préparatoire (une demi-journée à une journée) a pour but :

- d'informer les participants sur la méthode ;
- de délimiter le projet ;
- d'identifier le travail préparatoire :
 - collecte d'informations sur l'existant (flux, volumes),
 - enquêtes internes de réflexion sur les exigences,
 - entretiens de direction sur les objectifs du projet,
 - prise en compte des éléments de l'étude préalable.

A l'issue de cette réunion, les participants disposent de quelques jours pour réaliser la préparation de la première réunion de CADRAGE.

2.1.8. Réunion de lancement

PARTIE PUBLIQUE (TOUTES LES PERSONNES CONCERNEES)

- ◇ **Présentation, conférence, principes, avantages et contraintes du RAD**

Un animateur RAD expérimenté dispose en général d'une présentation générale (15 à 60 minutes) et de présentations spécialisées propres à chaque maîtrise.

La durée de la formation dépend à la fois du projet et des connaissances des participants.

- ◇ **Limitation du périmètre initial du projet**

La limitation du périmètre est effectuée à partir d'un modèle global de flux, de cas d'utilisation ou simplement d'une liste des fonctionnalités globales attendues (30 à 60 minutes).

◇ Définition des groupes de travail

Dans le cadre du plan de communication, les utilisateurs sont répartis dans des groupes de travail organisés par thème. Cette étape doit s'effectuer en quelques dizaines de minutes.

Le travail de groupe implique la disponibilité totale et permanente de tous les participants, particulièrement des décisionnaires. Ce point doit être explicitement reconnu par les membres du groupe et par leur encadrement.

En Construction, il faut distinguer les groupes de travail des participants à la réunion du Focus. Un groupe de travail réunit généralement moins de dix personnes qui concourent à l'analyse d'un problème et à l'élaboration des solutions. Dans le cas du prototypage, la notion de validation permanente implique en général une équipe d'un ou deux utilisateurs par informaticien de développement. Le Focus peut rassembler un nombre élevé de participants pour une sensibilisation, une prise de connaissance ou un jugement général (validation) sur le produit en cours de réalisation.

Les utilisateurs associés en permanence à un développement RAD sont des « *end-users* ». Cette condition implique la participation des **exécutants effectifs** des tâches à automatiser.

Le choix des utilisateurs participant au projet est fonction de leur motivation. Ce point est essentiel au RAD, les futurs clients du système représentent une force de proposition fonctionnelle.

La réunion de lancement (d'une demi-journée à une journée) est souvent considérée comme le lancement officiel du projet RAD. La réunion de lancement regroupe :

- l'animateur,
- le rapporteur,
- les coordonnateurs du projet,
- des utilisateurs significatifs,
- des experts des domaines concernés,
- éventuellement des observateurs.

Cette réunion a pour objet en première partie :

- d'informer les participants sur la méthode ;
- de délimiter le projet ;
- d'identifier le travail préparatoire ;

et en seconde partie :

- de préconiser le lancement d'enquêtes et de réflexions sur les exigences ;
- de réaliser des entretiens de direction sur les objectifs du projet ;
- de prendre en compte les éléments de l'étude préalable.

◇ Proposition du planning de Cadrage initial

A l'issue de la réunion de lancement, les participants disposent de quelques jours pour réaliser la préparation de la première réunion de Cadrage. Ils en fixent la date définitive.

GROUPES DE TRAVAIL (PERSONNES IMPLIQUEES)

◇ Ouverture des ACTIONS (individualisation des travaux)

En seconde partie de la réunion, dans chaque groupe de travail, les points « d'action » ouverts sont attribués, en termes de responsabilité, à la ressource la mieux à même de rechercher une solution.

◇ **Validation du planning de CADRAGE**

Suite à l'analyse des divers points d'actions ouverts, la date de la première session de Cadrage est fixée définitivement.

◇ **Publication des principes du projet et de la composition des groupes et comités**

Le compte-rendu de la réunion de lancement est produit. Les principes du projet et la composition des groupes et comités sont publiés.

2.1.9. Trois formes de développement

Parfois, le développement spécifique d'une application n'est plus justifiable, particulièrement lorsque les délais représentent un critère primordial.

Un développement peut se réaliser économiquement à partir d'une combinaison de trois options :

- achat d'un progiciel,
- développement spécifique,
- achat et combinaison de « solutions ».

En fonction du problème, des solutions possibles et de l'environnement, le choix de la solution optimale peut pragmatiquement s'effectuer à trois moments décisifs de la vie du projet :

- à la fin de l'étape d'immersion de l'animateur dans le domaine fonctionnel si une solution « progiciel » couvrant la totalité des exigences d'un périmètre fonctionnel figé est évidente ;
- à la fin de la phase de Cadrage, lorsque les exigences du système cible sont définies et qu'une solution spécifique est écartée pour des raisons de coûts, de délais ou de ressources ;
- à la fin de la phase de Design, dans les cas complexes ou lorsque la modélisation des traitements, des données et des communications permet de comparer tout ou partie des solutions dans le détail de leurs avantages et inconvénients réciproques.

La mise en œuvre d'une solution « composants » nécessite lors de la phase de Cadrage de distinguer l'espace du besoin de celui de la solution (et de ses contraintes). Le RAD répond parfaitement à cette exigence en spécifiant :

- l'**expression des exigences** dans le document nommé suivant les habitudes Plan d'Assurance Qualité Fonctionnelle (PAQF) ou Plan de Développement Logiciel (PDL), ou dossier de Cadrage (exigences fonctionnelles) ;
- la **proposition de solution** dans le document nommé Plan d'Assurance Qualité Technique (PAQT).

2.2. Phase CADRAGE

2.2.1. Initialisation des Cadrages

FORMATION COMPLEMENTAIRE DES RESSOURCES PARTICIPANTES

Organisation de la formation des ressources participantes en tenant compte de l'environnement cible. Former la coordination en insistant sur l'aspect entretien de groupe. Si les utilisateurs ne sont pas familiarisés à Windows et au monde micro, il faut engager leur formation générale dès le début du projet.

COMPOSITION EQUIPE SWAT DE CADRAGE

Le prototypage déporte les spécifications détaillées dans la phase de réalisation. Les compétences de l'analyste et du programmeur fusionnent et imposent une mutation des métiers. En parallèle, la pluralité technologique requiert une spécialisation accrue de l'équipe de projet. C'est le principe du SWAT.

Un SWAT est composé de personnels de type spécialiste-généraliste, experts sur des techniques ou outils précis et nécessairement généralistes sur les autres. Une équipe de développement de type SWAT est avant tout autonome et responsabilisée : autonome dans les choix techniques des solutions et responsabilisée dans la performance de réalisation qui en découle. Utiliser des ressources d'un profil unique de type « concepteur-développeur » présente de multiples intérêts mais aussi des contraintes dans leur engagement : les ressources doivent intervenir simultanément dès le début réel du projet informatique.

DISPONIBILITE DE L'ENVIRONNEMENT DE CONCEPTION-DEVELOPPEMENT

Il est inutile d'engager des ressources humaines de qualité et coûteuses si les ressources matérielles (salle, vidéoprojecteurs, micros, logiciels) ne sont pas disponibles.

MISE EN CE UVRE DES OUTILS DE COMMUNICATION ET DE VALIDATION PERMANENTE

Vérifier la présence ou rendre opérationnels des outils de communication efficaces, tels que courrier électronique, agenda électronique, boîte vocale, etc.) . C'est une priorité dans les projets majeurs. Pour un petit projet, des moyens organisationnels de substitution pourraient être employés.

2.2.2. Cadrages globaux

CADRAGE - STRATEGIQUE

◇ Définir de manière exhaustive les ambitions du système « cible »

Estimer la priorité des exigences, évaluer, planifier. Le système « cible » est concrètement défini en termes de buts. Engager à cette étape une réflexion sur la gestion des risques.

CADRAGE - FONCTIONNEL (PDL OU PAQF)

◇ Réaliser les entretiens de groupe

Modéliser en direct avec un vidéoprojecteur électronique.

Cette étape nécessite des équipes réduites incluant des utilisateurs de tous niveaux. Le processus est le suivant :

- dans chaque domaine, les thèmes principaux sont déterminés ;
- dans le cas d'un domaine stable, il n'est pas nécessaire de réaliser une étude détaillée de l'existant en session plénière ;

- dans les domaines où les fonctionnalités sont instables, un effort de réflexion devra être engagé et le cercle d'interlocuteurs élargi.

La durée des sessions est d'une demi-journée ou d'une journée. Il est possible en cas de contraintes de délais de réaliser des sessions de 5 jours. Elles engagent les informaticiens de conception-développement (le SWAT) et les utilisateurs concernés (quatre à six au maximum). Seules les validations d'un ensemble conséquent modélisé nécessitent une session plénière.

En général, les groupes sont limités à quelques utilisateurs significatifs. Pour améliorer la couverture de cette activité et lever un maximum de risques, il est souhaitable d'élargir la base de participants dans le cadre d'une consultation préalable. Avant la réunion officielle, on engagera donc une réflexion sur l'ensemble des thèmes composant le domaine applicatif.

- ◇ Synthétiser en direct sous la forme dite de la « documentation utilisateur »

Réaliser une **hiérarchie de fonctions**. Ce formalisme peut être élaboré avec les utilisateurs, simple et pratique il est compréhensible par l'ensemble des intervenants.

Les « use case » doivent respecter une unicité de thème, de lieu ou de temps à l'instar de la structuration en actes d'une pièce de théâtre qui sert à marquer les ruptures de contextes ou les transitions temporelles.

Sans être totalement exhaustifs dans le détail, les scénarios doivent être représentatifs de toutes les situations à considérer. Les activités ne répondant pas à des critères d'unicité de thème, de lieu ou de temps font l'objet de traitements indépendants.

Au-delà d'une aide intéressante à la validation des modèles, les « use case » définissent les cheminements fonctionnels et sont particulièrement utiles aux informaticiens. Ils facilitent les validations lors des sessions de travail, des Focus et lors de la recette générale avant déploiement. Cette dernière étape représente souvent une tâche ardue dans les grands projets. La technique des « use case » se révèle alors un investissement intéressant.

MERISE - généralement :

- ◆ **Modèle Conceptuel des Traitements (détaillé).**
- ◆ **Modèle Conceptuel des Données (global).**

FLUX - généralement :

- **E-R modèle des données (global).**
- **les niveaux de DFD de 1 à 2 (ou 3).**

UML - suivant le besoin, pour chaque scénario, les diagrammes :

- ❖ **de cas d'utilisation,**
- ❖ **de collaboration « associé »,**
- ❖ **de séquence (du scénario),**
- ❖ **les diagrammes d'activité (du scénario).**

- ◇ Réaliser un PDL (Plan de Développement Logiciel)

Réaliser si possible le PDL dans la vision d'un manuel utilisateur en évitant le style cahier des charges « classique ».

La hiérarchie de fonctions offre l'avantage de préciser immédiatement et naturellement le menu de la future application.

◇ **Engager la phase de définition globale des fonctionnalités et des contraintes**

La technique de spécification dite de « documentation d'utilisation » est applicable si les exigences sont stables. Cela peut être le cas lorsque l'utilisateur s'est forgé, préalablement à la conception, une idée précise de ce que doit être son application. De par sa nature, cette forme d'expression n'inclut pas les aspects techniques. Elle est cependant particulièrement intéressante lors d'une refonte de système qui s'appuie sur une architecture technique connue ou standard. Ce qui est de plus en plus souvent le cas (Oracle ou SQL Server comme SGBD et Windows NT en serveurs et postes de travail).

Les éléments décrits sous cette forme sont la base du plan de Cadrage Fonctionnel. Si nécessaire, les informations additionnelles souhaitées sont intégrées dans le plan de Cadrage Technique :

- volumes des informations,
- nombre et type d'utilisateurs,
- temps de réponses attendus,
- profils des postes de travail,
- niveau de qualité ergonomique,
- niveau de validation,
- conditions de livraison,
- conditions de déploiement, etc.

◇ **Modéliser en direct avec un vidéoprojecteur**

Les données sont modélisées sous la forme entité-relation dans un MCD global (les attributs ne sont pas nécessairement détaillés). Les traitements sont modélisés sous la forme d'une hiérarchie de composants (les grandes fonctions de l'application).

Il faut conserver à l'esprit que vous serez jugé au final sur la qualité de l'application et non sur le papier produit. L'information formalisée nécessaire à la réalisation de cet enjeu doit être à la fois suffisante et minimale. Cet équilibre est un des défis auxquels le RAD s'attache à fournir une réponse optimale. Ne gaspillez pas un temps précieux à justifier les évidences.

◇ **Établir la priorité des exigences, évaluer, planifier**

Les divers éléments du plan de Cadrage sont à la fois nécessaires au bon traitement du problème et à l'évaluation du projet. Prenons pour exemple le niveau de qualité ergonomique. On pourrait imaginer en premier lieu que le niveau ergonomique est le même dans toutes les applications Windows. Depuis quelques années cette assertion est vérifiée par presque tous les progiciels Windows commercialisés par les grands éditeurs (*Microsoft, Lotus, Corel, etc.*).

En ce qui concerne les applications spécifiques développées en entreprise, le problème est tout autre. L'ergonomie sous Windows est devenue un art dont la maîtrise est indispensable. Selon sa spécificité, une application peut nécessiter un choix parmi plusieurs niveaux de qualité ergonomique.

Le choix d'appliquer un niveau plutôt qu'un autre a pour conséquence de modifier notablement la charge de travail nécessaire et par conséquent le coût de l'application (*tableau 1*).

Tableau 1. — Niveau de qualité de l'interface graphique

Niveau	Se caractérise (par) ou s'applique (à)
Rustique	Niveau d'une application « Kleenex » ou à budget réduit, utilisée par peu d'utilisateurs et limitée aux activités internes.
Standard	Niveau moyen d'une application classique, développée selon une charte graphique.
Elevé	Niveau d'une application commerciale, pouvant être justifié en entreprise par le nombre ou le type d'utilisateurs.
Extrême	Niveau d'application justifiant des moyens exceptionnels (graphisme de haut niveau, haute résolution) pour des raisons stratégiques (clientèle, concurrence, etc.).

CADRAGE - TECHNOLOGIQUE◇ **Rapprocher les exigences des technologies existantes (intégrateur de technologies)**

Le RAD fournit naturellement une assurance qualité détaillée au niveau le plus fin de granularité fonctionnelle. Néanmoins, les informations du PAQF et/ou du PAQT complètent le Cadrage en intégrant les diverses contraintes de qualité si la taille du projet ou sa typologie le nécessitent.

◇ **Prévoir l'architecture technique de développement et d'exploitation**

Un sujet aussi vaste et mouvant ne peut être traité sérieusement dans le cadre d'un ouvrage consacré aux méthodes.

◇ **Garantir la faisabilité technique et la montée en charge**

Seule la mise en évidence de particularités extrêmes nécessite une étude technique approfondie.

CADRAGE - ORGANISATIONNEL◇ **Prévoir les impacts organisationnels et en informer la direction.**

Bien souvent, cette étape est volontairement oubliée. C'est notamment le cas lorsque ces impacts existent et qu'ils risquent d'être ressentis négativement par les personnes concernées.

Le plus sage est d'obtenir, dans la transparence, un consensus fondé sur une revalorisation des postes ou un redéploiement.

Dans de nombreuses entreprises canadiennes, il m'a été donné d'assister à des actions de réingénierie sans drame.

◇ **Optimiser l'organisation préalablement à la modélisation organisationnelle**

Dans le cas où le projet induit des impacts sur l'organisation, les opérations de mise en place des changements organisationnels devront être planifiées immédiatement, car elles sont en général plus longues et difficiles à gérer qu'un projet RAD.

CADRAGE - CONTRAINTES◇ **Définir formellement les contraintes du projet**

Les contraintes de base sont le budget, les délais, la qualité fonctionnelle, la qualité technique, la visibilité, les ressources.

En fonction de ces contraintes et des priorités, la combinaison la plus appropriée est proposée, et les risques qui découlent de ce choix sont mis en évidence (figure 2).

Le lotissement qui permet des livraisons rapides et un retour sur investissement accéléré est un moyen de réduire le risque.

Le RAD, par sa démarche itérative, réduit naturellement les risques et particulièrement le risque de non-conformité aux exigences.

◇ **Sécurité du système**

Les risques concernant les données doivent être évalués dans l'optique de la charge qu'ils induiront.

Durant l'étude d'opportunité ou la phase de Cadrage, un projet informatique doit prouver sa rentabilité qui n'est pas uniquement financière. La qualité et la réactivité sont liées aux conditions de travail et doivent s'évaluer méthodiquement.

On distingue les fonctions d'usage des fonctions d'estime ainsi que les fonctions de services (attendues) des fonctions de techniques (contraintes). Lors de l'analyse fonctionnelle on spécifie aussi toutes les contraintes (environnement, technologie, performances, etc.).

La définition des fonctions, de leur importance, de leur valeur ajoutée et de leur coût est la base de calcul d'un poids qui justifie leur priorisation. L'analyse de valeur ajoutée, comparée aux coûts, met en évidence les anomalies, comme par exemple la présence ou la demande de fonctions secondaires à des coûts anormalement élevés.

◇ **Planification et ordonnancement**

Bien que les projets RAD induisent des cycles plus courts que les développements classiques, le suivi nécessite une attention particulière et l'usage d'un outil spécialisé permettant une gestion au niveau le plus fin). Cet aspect est traité en détail dans le chapitre 11.

Le pilote de projet RAD organise des réunions de contrôle. La fréquence hebdomadaire, en début de projet, peut passer à deux semaines en phase de réalisation. La durée ne doit pas excéder une heure par équipe RAD.

2.2.3. Validation des Cadrages

◇ **Faire valider les modèles globaux de communication, de traitement et de données par une équipe élargie d'utilisateurs**

Procéder à une validation fonctionnelle par l'ensemble des groupes concernés.

◇ **Faire valider les aspects stratégiques, organisationnels et financiers par une instance de direction**

Faire valider l'ensemble des travaux de la phase et les conditions de poursuite du projet par le comité de pilotage. L'ensemble des risques résiduels est mis en évidence et évalué.

2.2.4. Solution « progiciel »

La fin du Cadrage est souvent le moment idéal pour envisager le choix d'une solution de type progiciel ou composant. Parfois le choix judicieux d'un progiciel peut permettre une réalisation rapide et économique de l'application

2.3. Phase DESIGN

2.3.1. Architecture de conception

Actuellement, le vrai problème n'est pas AVEC QUOI modéliser, auquel cas la réponse serait Merise, DFD, UML ou autres formalismes et notations, mais COMMENT et POURQUOI modéliser une application décisionnelle, opérationnelle lourde, kleenex, stratégique, etc.

Pour le pragmatique, la vraie question se pose ainsi : quel but doit viser l'exercice de modélisation d'un SI moderne ? Malheureusement, pour de nombreux informaticiens, dans la réalité d'un projet, la réponse implique des notions de niveaux d'abstraction et d'axes de modélisation et la modélisation reste une chose abstraite. Il est vrai que les développements classiques concernaient rarement des applications transversales complexes et que le changement de typologie des projets est brutal et récent.

2.3.2. Modélisation des application transversales

Une conception efficace, quel que soit le formalisme (UML, OMT, Yourdon/DeMarco, Gane & Sarson, Structured System Analysis and Design Method, Merise), s'appuie sur la complémentarité de trois axes de modélisation distincts : l'axe statique fixe la structure des données, l'axe dynamique définit la vie des processus, l'axe fonctionnel détaille les traitements. La notion d'adaptation est la cible visée grâce à la maîtrise de ces axes. Une organisation est dans l'incapacité de muter rapidement si son système d'information impose de lourdes refontes à chaque étape de sa marche forcée vers la productivité et la réactivité concurrentielle [Yourdon 1989, Rumbauch 1996, Muller 1997].

Le rôle des axes de modélisation est de limiter la portée des changements en cours de développement et en phase de maintenance :

1. L'**axe statique** est le plus stable. Il faudrait un changement majeur du métier de l'entreprise pour que l'on soit contraint de le modifier. Sa structuration générique doit permettre toutes les formes d'évolution.
2. L'**axe dynamique** colle à l'aspect organisationnel. Seule une réorganisation des acteurs et des processus peut avoir un impact sur sa définition. Il doit servir de base à tous les redéploiements de l'organisation sans affecter l'axe statique.
3. L'**axe fonctionnel** doit s'adapter aux conditions immédiates de conduite de l'organisation. Face aux évolutions dans la manière de traiter les produits ou services offerts, il limite les changements à une redéfinition minimum des traitements.

2.3.3. Concepts « en vue de modifications »

Le concepteur utilise un niveau d'abstraction élevé. Il définit initialement un modèle de données, suffisamment généraliste pour couvrir en un seul axe de structuration l'ensemble du métier de l'organisation ou la partie concernée par le système. Le noyau stable acquis, il modélise en couches périphériques la variété de traitements permettant de mettre en œuvre des stratégies opérationnelles. En cas d'évolution, il suffit d'adapter la couche concernée.

Modéliser avec un niveau d'abstraction « métier » est le meilleur moyen de repousser les limites d'un existant et de garantir les capacités d'évolution ultérieure de l'applicatif.

Cette approche s'avère plus facile à développer malgré sa généralité :

- les parties les moins sujettes à évolution sont réalisées en premier ;
- les autres parties sont planifiées dans un ordre dicté par leur stabilité.

Le but est d'obtenir tous les modules du système simultanément disponibles au moment du déploiement. Structuration, isolation, cohésion, modularité, généralisation, encapsulation et surtout dissimulation sont les techniques de base de la modélisation qui permettent la mise en œuvre d'une architecture de conception évolutive.

Le RAD s'appuie sur les techniques dérivées de l'objet et préconise une architecture à « modèles variables » ainsi qu'une conception « en vue de modifications ».

2.3.4. Organisation et environnement du SWAT

Dans un projet complexe, l'ensemble du SWAT (les concepteurs-développeurs) participe à la phase de Cadrage. Dans un projet simple, l'ensemble du SWAT peut intervenir à partir du Design. Dans ce cas, le Cadrage doit être limité à ses grandes lignes. Cette logique d'engagement est également souhaitable dans le cas où une réorganisation doit être envisagée préalablement au développement. Le SWAT idéal regroupe la complémentarité de 4 à 6 concepteurs-développeurs.

MERISE - généralement :

- ◆ **Modèle Organisationnel des Traitements.**
- ◆ **Modèle Conceptuel des Données (détaillé).**

FLUX - généralement :

- **E-R modèle des données (détaillé).**
- **les niveaux de DFD de 2 à n (suivant le besoin).**

UML - généralement :

- ❖ **Modèle de classes incluant l'héritage, l'agrégation, les relations bidirectionnelles et les contraintes.**
- ❖ **Modèle des interactions entre objets.**
- ❖ **Modèle schématique de transitions entre états définissant le cycle de vie des objets au sein d'une même classe.**

2.3.5. Modélisation détaillée des communications

Une Modélisation Conceptuelle des Communications (MCC) ou des Flux de premiers niveaux (DFD) est la première étape d'une analyse. Elle est souvent réalisée lors du Cadrage. Elle permet de déterminer tous les intervenants, les informations et leurs supports. Les flux sont ensuite détaillés et font apparaître les points où l'information est visualisée, modifiée ou créée (générateurs de données primaires). Les concepts généraux de sécurité s'identifient à partir du modèle des communications.

2.3.6. Modélisation globale des traitements

La modélisation des traitements est en général la dernière des étapes de spécification d'un système opérationnel et souvent la première étape d'une application décisionnelle. A partir du modèle de communication et en utilisant le dictionnaire issu de la modélisation des données, on détermine les concepts globaux de traitement. Ils

sont ultérieurement affinés par une décomposition dont le niveau est fonction du type de projet.

2.3.7. Modélisation détaillée des données

La modélisation conceptuelle des données représente la phase suivante. Elle s'appuie sur l'analyse des points générateurs de données primaires, pour dégager la structure générale des données sous la forme entité-relation. Toutes les entités sont alors détaillées, au niveau attribut, dans un dictionnaire géré par l'AGL de conception. Il devient le référentiel de toutes les autres étapes du développement.

Les principes du client-serveur offrent une grande autonomie de manipulation aux utilisateurs. Il est nécessaire, en contrepartie, de renforcer la sécurité et l'intégrité référentielle directement au niveau du moteur relationnel.

2.3.8. Recherche de solutions

TECHNOLOGIES DE DEVELOPPEMENT

Le RAD doit s'appuyer sur la veille technologique pour offrir les solutions les plus adéquates dans les meilleures conditions : « *build the right system and build the system right* ».

DEFINIR DES INTERFACES FORMELLES

A partir de cette étape, la conception laisse place au développement. Lorsque les délais imposent la rapidité, il faut négocier la spécificité de certaines fonctionnalités.

APPROCHE « PROGICIEL »

Entre le tout-progiciel et le tout-spécifique, une nouvelle vision consiste à combiner la puissance du progiciel à la spécificité des développements sur mesure. On peut nommer cette technique "progiciel sur mesure" ou "progiciel partiel". Cette approche repose sur le mariage du spécifique et du générique. L'objectif est d'obtenir une réponse 100% satisfaisante pour le client. La raison d'être majeure de ce type de produit hybride est la recherche d'une triple optimisation en termes de prix, de rapidité et de fiabilité. Le critère principal est, surtout pour les applications stratégiques, la durée de mise en œuvre qui conditionne la réactivité des organisations soumises à une forte concurrence.

Le progiciel de base fournit un noyau de fonctions génériques, et une couche spécifique absorbe l'expertise métier propre à chaque entreprise. Le produit final est adapté à la mise en œuvre de solutions particulières, stratégiques et vitales. En principe, les performances et la fiabilité sont garanties, et des plates-formes permettant une certaine « visibilité » existent naturellement chez le producteur ou chez ses clients ayant déjà adopté un produit dérivé.

APPROCHE « COMPOSANTS »

En passe de devenir un standard du marché, la notion de composants réutilisables est à la base des solutions orientées métiers. Les grandes sociétés de services disposent déjà de bibliothèques de classes pour certains métiers.

Le *chapitre 20* détaille la mise en œuvre des consultations permettant de sélectionner le meilleur « produit » ou la meilleure « solution ».

APPLICATION « DECISIONNELLE »

Les méthodes de stockage (*DataWarehousing*) et d'accès (*Datamining, Drill*) aux grandes masses d'informations utilisées par les nouvelles applications de type « décisionnel - stratégique » ne relèvent pas simplement de techniques d'indexage. Un exemple pratique permettra d'éclaircir le sujet. Actuellement deux modèles basiques

d'organisation des données dans un SGBDR coexistent. Leur choix est fonction des types de traitements qui leur seront appliqués :

- **OLTP** (*on line transactional processing*) est le principe dont l'application régit le relationnel classique. Il s'applique en général à la création et à la mise à jour individuelle des informations par le biais de transactions légères.
- **OLAP** (*on line analytical processing*) est un principe plus récent dont l'application régit le relationnel d'analyse. Il fait généralement intervenir des requêtes complexes manipulant de grandes quantités de données.

En OLTP, un cube tridimensionnel avait classiquement des surfaces pour le temps, le produit, et le critère géographique (données dimensionnelles). Il était généré à partir de trois tables en relation. Techniquement, la table contenant la donnée à analyser (donnée basique) gérait des clés étrangères, chacune pointant sur une des trois tables représentant les dimensions de l'analyse. Ce modèle était généralement parfaitement normalisé (Codd, 3^{ème} forme normale).

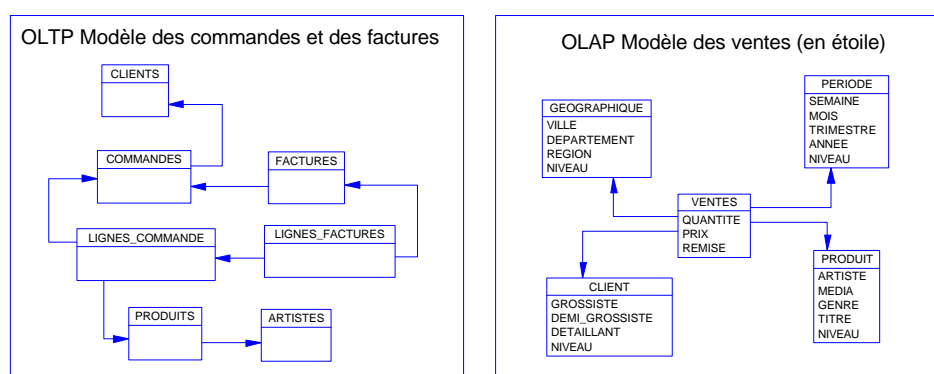


Figure 2. — Modèles comparés OLTP /OLAP

En OLAP, les tables de chaque dimension sont « dénormalisées » et organisées hiérarchiquement (figures 2 et 3). Pour exemple : la dimension temps, si elle décompose l'année, le trimestre, le mois et la semaine, nécessitera 5 tables relationnelles. Le schéma en étoile offre alors une simplification notable dans la définition de cette hiérarchie.

Les principes OLAP ont été conçus pour faciliter l'accès des utilisateurs aux données. Le concept de base d'OLAP est une organisation en étoile des informations. Pour des raisons pratiques, le principe d'OLAP s'est spécialisé en trois sous-groupes :

1) **MOLAP** (*Multidimensional OLAP*) stocke les données basiques et leurs agrégations sur un serveur spécialisé OLAP. Les langages de *query* MOLAP sont optimisés pour les applications d'analyse et la meilleure performance en termes de temps de réponse pour un type de *query* complexe. MOLAP est particulièrement adapté à la modélisation et à la simulation financières.

2) **ROLAP** (*Relational OLAP*) conserve la table de la donnée de base sur le serveur OLTP d'origine (ou dans un entrepôt de données) mais utilise un jeu (set) indépendant de tables relationnelles pour stocker les références dimensionnelles et les données agrégées. ROLAP est particulièrement adapté aux analyses de marché (pour, par exemple, déterminer l'intérêt d'une promotion des ventes).

3) **HOLAP** (*Hybrid OLAP*) (aussi appelé DOLAP, Database OLAP) utilise les données de base depuis sur le serveur OLTP d'origine mais stocke les agrégations dans un format multidimensionnel sur un serveur spécialisé. La solution HOLAP combine donc les avantages des solutions MOLAP et ROLAP, mais aucun standard sur ce principe n'a encore fait l'objet d'un consensus.

Dans la généralité, un moteur OLAP est un engin de recherche et d'accès à l'information. Il doit supporter ces trois modes d'organisation et d'accès aux données. Néanmoins, la plupart des utilisations se limitent à un choix entre ROLAP et HOLAP.

Ces deux approches sont capables d'opérer directement à partir des tables OLTP conventionnelles, mais cela génère un grand nombre d'accès et une dégradation des performances du serveur. Aussi, il est généralement admis qu'un serveur spécialisé duplique la base de données. Le nombre de niveaux hiérarchiques gérés en table est dépendant de la granularité des dimensions. Le volume d'informations ainsi stocké peut être conséquent, aussi, il est préférable de conserver uniquement les données de base et de les agréger « à la demande ».

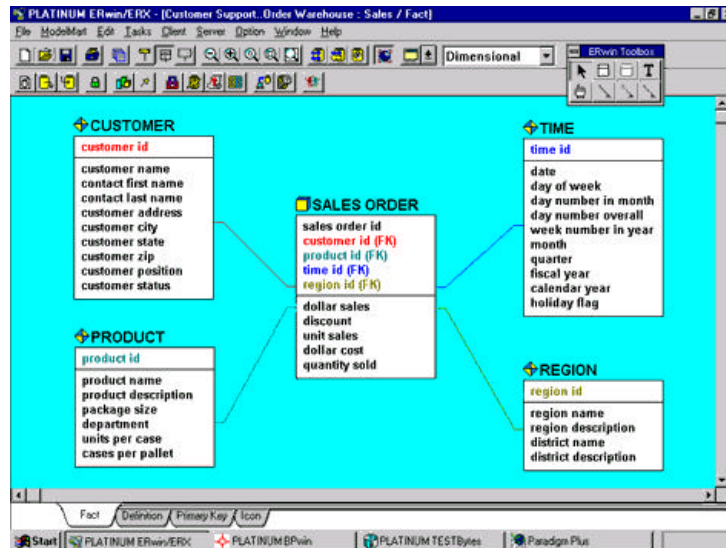


Figure 3. — Techniques dimensionnelles avec Erwin (Platinum)

Au-delà de la technique et afin d'en justifier l'évolution, le but ultime doit s'exprimer dans la simplicité pour l'utilisateur et la performance au service du facteur de différenciation concurrentielle. Microsoft, avec SQL Server 7, Plato (services OLAP) et Office 2000 (Excel 2000 pour la partie client), a bien compris le problème et offre un outil économique, simple et homogène qui va banaliser radicalement l'usage de l'analyse multidimensionnelle jusque dans les PME/PMI.

2.3.9. Planification détaillée de la Construction

La dimension temporelle du RAD, la capacité d'évolution et l'obligation de simplicité sont des aspects primordiaux de la méthode qui ont été très bien exprimés par P. Silvestre et D. Verlhac : « *La maîtrise du temps oblige à faire simple en un temps court. Ainsi un projet ne respectant pas l'une ou l'autre de ces conditions doit être revu.* »

SESSIONS PARALLELES

La phase de Cadrage représente un **passage commun** nécessaire à l'ensemble des concepteurs-développeurs.

La phase de Design permet de paralléliser la conception lorsque le projet le justifie.

Chaque concepteur-développeur peut alors prendre en charge une partie de l'application.

Il faut néanmoins maintenir un plan de travail commun pour tout ce qui concerne la modélisation des données.

LOTISSEMENT

En Construction, les techniques de parallélisation et de lotissement peuvent être appliquées conjointement.

Le lotissement s'appuie sur les délais et sur les priorités pour définir des lots. Si cela s'avère nécessaire et possible, leur réalisation est alors étalée dans le temps.

L'application est dans un premier temps livrée avec des fonctionnalités réduites.

Le respect de cette exigence est facilité par la technique dite d'état de livraison permanente que le RAD applique en tant que principe d'assurance qualité.

Pour garantir que l'application sera réellement opérationnelle, toutes les dépendances fonctionnelles devront avoir été considérées dans la planification du lotissement.

STRATEGIE DE LIVRAISON

Un planning de réalisation est proposé. Il prend en compte la priorité des exigences et leurs dépendances fonctionnelles. Une stratégie de livraison en découle. Elle englobe les problèmes d'acquisition, de logistique et, éventuellement, de sous-traitance.

ARCHITECTURE APPLICATIVE FINALE

MERISE - généralement :

- ◆ **Modèle Logique des Traitements.**
- ◆ **Modèle Physique des Données.**

FLUX - généralement :

- **E-R modèle physique des données.**
- **Décomposition des fonctions.**

UML - généralement :

- ❖ **Modèle de Construction décrivant les sous-systèmes et leurs interfaces.**
- ❖ **Diagrammes de Composants.**

2.3.10. Premier niveau de prototype

Un premier niveau de prototype peut être mis en œuvre soit à la fin de la phase de Design, soit au début de la phase de Construction.

Bien que théoriquement liée à la Construction, la présentation d'un prototype peut offrir une visibilité succincte mais appréciée lors du Focus de fin de Design.

Réaliser un premier niveau de prototype en débutant avec l'ergonomie générale et les menus. Poursuivre en incluant un jeu limité de fonctionnalités, autant que possible celles qui sont identifiées comme prioritaires dans la planification.

Le prototypage permet d'obtenir naturellement :

- un meilleur contrôle de la planification (granularité liée aux niveaux de prototypage) ;
- une application techniquement réduite (pas de dérive informatico - informaticienne) ;
- une application fonctionnellement moins complexe (les utilisateurs s'attachent en général à la simplicité) ;

- la réduction des risques par accroissement de la visibilité (feed-back immédiat) ;
- une gestion incrémentielle de la complexité des détails (spécification directe et « *just in time* ») ;
- une validation permanente et un verrouillage progressif.

2.3.11. Focus de conception

Un Focus représente un événement attendu qui offre une visibilité concrète à toutes les parties concernées par l'application. Le principe des Focus est décrit dans le chapitre consacré à la phase de Cadrage.

Dans le cas où l'importance du projet le justifie, il est possible de planifier des Focus de Design.

En général, un seul Focus de fin de phase de Design récapitule les exigences du projet et démontre que :

- la modélisation les a prises en compte ;
- la conception générale y apporte des réponses satisfaisantes.

2.3.12. Actualisation du Cadrage (budget, délais, ressources)

Réactualiser l'évaluation et la planification de la phase de Construction en fonction des connaissances acquises lors du Design.

2.3.13. Validation de fin de Design

L'ensemble des risques résiduels doit être mis en évidence et évalué. Le comité de pilotage valide l'ensemble des travaux de la phase terminée et les conditions de poursuite du projet. Le budget correspondant à la phase suivante est libéré.

2.4. Phase CONSTRUCTION

2.4.1. Les techniques de réalisation

Un développement RAD se distingue par la mise en œuvre de plusieurs techniques de réalisation :

- **SWAT** : Organisation d'une équipe de profil « concepteur-développeur » fondée sur la compétence, la complémentarité, l'autonomie et la démocratie.
- **Normes de codage** : Normes publiées et acceptées visant à uniformiser les techniques de codage.
- **Validation permanente** : Engagement continu des utilisateurs dans le prototypage. Les réunions sont informelles, elles engagent un ou deux utilisateurs par concepteur-développeur.
- **Revue de code** : Principes planifiés et acceptés de vérifications croisées de la conformité des pratiques de codage aux normes publiées.
- **Jalons Zéro Défaut (ZD)** : En planification, un jalon est une date particulière d'un projet, correspondant généralement à une fin d'étape importante. Les Jalons Zéro Défaut correspondent à une revue du code suivie d'une intégration et d'une validation complète de cheminement par l'utilisateur de base. Le positionnement et le suivi de ces jalons permettent de contrôler l'avancement du projet en matière de visibilité et de qualité. Des jalons ZD peuvent être, si nécessaire, planifiés entre les Focus.
- **Focus** : Réunion plénière au niveau du projet qui correspond obligatoirement au positionnement d'un des jalons ZD.
- **Etat de livraison permanente** : Suite à un Focus ayant confirmé le jalon ZD, une version de l'application est maintenue dans un état livrable. Elle peut être présentée à tout moment ou utilisée si nécessaire en fonctionnalités réduites.

2.4.2. Organisation des SWAT de réalisation

En principe, le SWAT complet est engagé dès le Cadrage. Dans certains cas, en fin de Design.

Face à une application stratégique où les délais sont incompressibles et les fonctionnalités indispensables, il faut parfois se résoudre à augmenter la taille du SWAT. Cette opération est souvent dangereuse et toujours coûteuse. Il est de loin plus économique et plus sûr de prévoir une marge de sécurité dès le début du projet.

La réduction de la taille d'un SWAT ou la livraison anticipée de l'application sont des solutions plus faciles et agréables à gérer qu'un retard ou que la nécessité d'ajouter des ressources.

Un projet RAD nécessite :

- **des utilisateurs secondés par des ressources additionnelles, disponibles à l'optimum du besoin ;**
- **des concepteurs-développeurs totalement engagés dès la phase de Cadrage.**

Dans un tel contexte, les coûts et les temps de développement sont en général réduits de 50 % par rapport à un projet identique où toutes les conditions de la performance RAD n'ont pas été réunies.

Engager des utilisateurs implique clairement l'allocation de moyens additionnels suffisants à la réalisation des tâches quotidiennes. Les comptes réalisés à la fin du projet sont à cet égard éloquentes.

Le prix payé par ceux qui ne rencontrent pas ces exigences d'engagement est sans commune mesure avec l'investissement initial requis pour travailler efficacement.

2.4.3. Mise en œuvre de l'environnement de Construction

Sur le plan des ressources matérielles, l'environnement de Construction est semblable à celui utilisé en Design. La différence se situe au niveau logiciel. A l'AGL de conception succède l'AGL de réalisation.

La recherche de solutions techniques a mis en évidence les solutions les plus appropriées. Dans la panoplie, en plus de l'outil principal, on trouve les composants métier, les ActiveX, l'utilisation de langages (VBA, Delphi, etc.) et des logiciels de bureautique communicante.

2.4.4. Architecture de réalisation

L'architecture de réalisation sécurisée offre une « visibilité optimale ».

La clé de la productivité est de s'appuyer sur des développeurs de haut niveau dotés de compétences couvrant à la fois la conception détaillée et la réalisation.

L'engagement de ce type de ressource présente de multiples intérêts dans la pratique du RAD, mais aussi des contraintes : l'ensemble de l'équipe doit intervenir simultanément dès le début du projet. La compréhension est alors globale et unique (conception-réalisation).

Ce principe évite les déperditions d'énergie car les utilisateurs ne sont pas sollicités à de multiples reprises. La formalisation et le transfert d'informations sont minimisés ainsi que les risques d'erreurs (ce qui n'est pas le cas lorsque les dossiers sont rédigés par des analystes et développés ensuite par des programmeurs).

Le développement d'une IHM graphique, pour des raisons techniques et économiques, fusionne la conception détaillée, la réalisation et les tests en une seule étape : le prototypage. C'est en fait le seul moyen raisonnable de réaliser actuellement une application sous Windows.

Assurance de qualité technique (fiabilité)

Dès le début du projet, il faut exposer les principes qui régissent la qualité et son contrôle.

Le coordonnateur technique publie une normalisation formelle. Les normes de programmation nécessitent un livre à elles seules (SDK). Les principaux points à respecter sont les suivants :

- norme de déclaration du nom des objets,
- sécurité généralisée (table utilisateur/module),
- procédure généralisée de gestion des erreurs,
- règles d'utilisation des « customs-controls »,
- norme de documentation intégrée des procédures,
- création d'un module standardisé de transaction.

Exemples : Si les développements utilisent Visual Basic, on impose l'option EXPLICITE qui oblige à déclarer les variables utilisées (en tête de procédure), la mise au point est ainsi facilitée. Les noms de variables, objets et procédures respectent le vocabulaire des mots clés.

Les variables globales sont, elles, préfixées du nom du module où elles sont définies. Les objets Visual Basic sont nommés selon les conventions utilisées par Microsoft. Il faut aussi en normaliser pour tous les objets additionnels qui seront utilisés (add-on, Activex).

Le coordonnateur technique organise ensuite une conférence technique sur les principes fondamentaux structurels. Dès le développement du premier module, il démarre la communication et la collaboration entre les développeurs. Il sera ensuite plus aisé d'organiser les revues en préalable aux Focus (figure 4).

Les revues sont des examens critiques de *code* et de *projet*. Afin de produire un interface et un code homogène, les développeurs soumettent leur production à un examen réalisé par leurs collègues et sollicitent des commentaires critiques.

Obtenir cette confiance réciproque est la première des étapes à franchir par un coordonnateur technique RAD.

La réalisation d'une application de qualité nécessite qu'un audit soit effectué par des spécialistes extérieurs à l'équipe de développement.

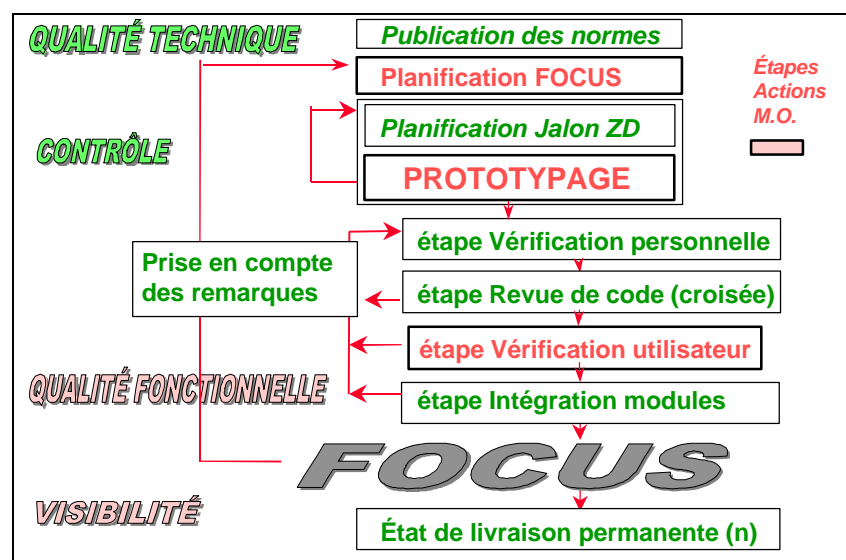


Figure 4. — Architecture de réalisation, FOCUS de Construction

2.4.5. Assurance qualité fonctionnelle (conformité)

La validation permanente, comme son nom l'indique, est réellement permanente. Elle s'effectue à chaque séance de travail avec l'utilisateur. Elle garantit, à tout ajout de fonctionnalité, la conformité au besoin. Le groupe de travail idéal comprend un membre du SWAT et un ou deux utilisateurs.

En général, l'animation RAD ne participe pas à ce type d'entretien. La présence des utilisateurs sur le site ou la nécessité de les faire se déplacer sont les paramètres à prendre en compte pour planifier ces séances.

Il est possible d'organiser deux validations de 15 ou de 30 minutes par jour comme de se contenter d'une validation d'une demi-journée par semaine.

Le plan classique d'un entretien de prototypage est le suivant : l'utilisateur manipule l'application, l'informaticien prend note de toutes les remarques. Lorsque l'utilisateur est disponible, l'informaticien peut modifier immédiatement les détails. Les deux parties fixent les objectifs et les conditions du rendez-vous suivant. Immédiatement après la séance, l'informaticien enrichit la fiche d'évolution prototype.

Le Focus représente une forme de réunion RAD uniquement dédiée à la validation et à la visibilité de l'avancement des travaux.

Lors d'un Focus, l'application est incomplète, mais stabilisée dans la limite de ses fonctionnalités réduites. Chacun peut alors, de l'utilisateur de base au dirigeant, avoir une vision globale du produit en cours de Construction, le critiquer et contribuer à son perfectionnement.

Le Focus est à la fois une opération de relation publique et un travail général de validation.

L'ensemble des intervenants participe à tous les Focus.

Le Focus garantit contrôle, visibilité, qualité fonctionnelle et technique. Lorsque, dans le respect d'une planification préalable, l'avancement des travaux est considéré comme suffisant, l'application est présentée à l'ensemble des intervenants.

Il faut une progression significative par rapport au Focus précédent pour justifier un nouveau Focus.

L'interface graphique ne rend pas systématiquement aisée la planification régulière de Focus.

Aucune technique du RAD n'est superficielle. La visibilité est un élément indispensable aux dirigeants et aux pilotes du projet.

Lors d'un Focus, chaque utilisateur manipule la partie de l'application qu'il a contribué à développer.

Les spectateurs observent et critiquent. Le groupe d'animation note les remarques. Dans ce type d'action, la discrétion de l'informaticien est proportionnelle à son efficacité technique et à la puissance de sa méthode.

Sur le plan des communications, l'organisation d'un Focus est proche de celle d'une session de travail en 3 étapes. Le nombre de Focus dépend de la durée du projet et de sa complexité². Il est en général planifié 3 ou 4 Focus pour un « petit » projet et de 4 à 8 Focus pour un projet intermédiaire (tableau 2).

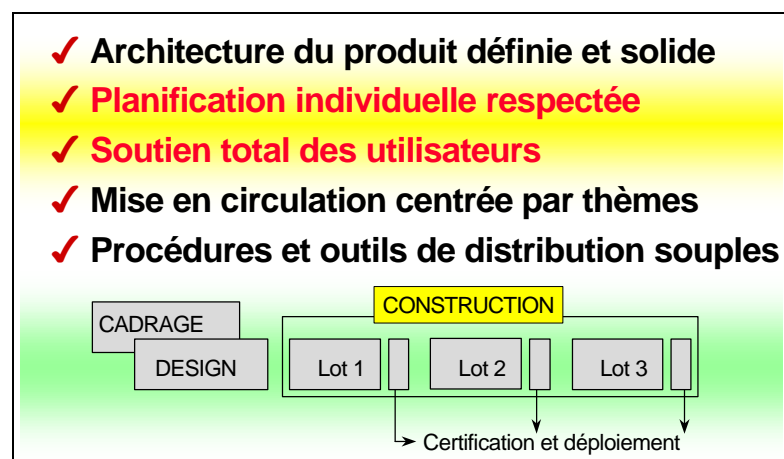


Figure 5. — Prérequis à la livraison par étapes

² Un projet considéré comme « petit » selon les critères du RAD engage une ou deux ressources expérimentées pour une durée de 30 à 90 jours. Un projet « intermédiaire » engage une équipe (SWAT, Skill With Advanced Tools) de 4 à 6 concepteurs-développeurs pour une durée de 60 à 120 jours. Les grands projets utilisent des techniques de parallélisation durant la phase de DESIGN et de sérialisation durant la phase de CONSTRUCTION. La planification des Focus dépend alors du nombre d'équipes et du style de projet.

Le RAD n'est pas dédié aux petits projets. Au contraire, il faut une organisation raisonnablement conséquente pour bénéficier pleinement des avantages offerts par cette méthode rigoureuse et formelle. En revanche, le RAD réduit l'ampleur des projets, particulièrement en termes de délais.

Il ne faut pas confondre la validation permanente qui découle naturellement de la présence de l'utilisateur lors du prototypage et l'état de livraison permanente³ qui est obtenu uniquement lors de la préparation d'un Focus de Construction après la validation d'un jalon zéro-défaut⁴.

Exemple : Considérons un projet type où la maîtrise d'œuvre est engagée pour 360 jours (SWAT de 4 personnes dans une dimension temporelle de 90 jours), la maîtrise d'ouvrage doit prévoir une charge minimale de 54 jours, le groupe d'animation et de rapport de 36 jours. La prévision du nombre de Focus fait état d'une possibilité de 6 à 8 Focus mais comporte seulement 3 états de livraison permanente et la livraison finale.

Tableau 2. — Nombre de Focus (NF) pour un projet moyen

Phase	Nbre	Type Focus	Objectifs du Focus
Initialisation	1	Lancement	Informar et individualiser les tâches préparatoires
Cadrage	1	Fin de Cadrage	Valider objectifs, fonctionnalités, budget et planning
Design	1-2	Fin de Design	Valider modèles et prototype de premier niveau
Construction	2-3	Visibilité	Valider l'application en fonctionnalités réduites
Finalisation	1	Livraison	Validation du dernier état de livraison permanente

RECHERCHE DE SOLUTIONS TECHNIQUES

CMM : La *Gestion des changements technologiques* vise à identifier les nouvelles technologies performantes (outils, méthodes et processus) et à en effectuer l'introduction ordonnée au sein de l'organisation. Cette pratique est mise en œuvre conformément au processus décrit dans le secteur clé *Gestion des changements du processus*. La priorité est donnée à la mise en œuvre pratique des innovations.

Définir les architectures et les techniques à employer immédiatement pour satisfaire les utilisateurs et rencontrer les objectifs d'un projet sous contraintes. Il ne faut pas négliger d'effectuer une étape de recherche de solutions techniques avant la phase de Construction. Les technologies et les outils de développement récents offrent souvent des possibilités de réduire les délais et les charges dans des proportions parfois considérables.

Prenons pour exemple une application actuelle mettant en œuvre une partie spécifique sous *Windows*, quelques écrans « NET » et un minimum de bureautique communicante. Pour la développer, une option consiste à éviter les produits *Microsoft*. La solution peut dans le meilleur des cas impliquer *Delphi* pour le spécifique, *Java* pour la partie NET et une suite bureautique disposant d'un langage « macro ». L'autre

³ L'état de livraison permanente correspond à une application fiable mais dont les fonctionnalités sont réduites.

⁴ Jalon zéro-défaut : état stabilisé de l'application à un niveau validé de qualité fonctionnelle et technique.

option est 100 % *Microsoft*. La puissance de son homogénéité est sans appel. Le langage *Visual Basic* se charge du spécifique, *VB Script* et les *Activex* de la partie NET, *VB Application* automatise la partie bureautique communicante. Dans les faits, outre vitesse et simplicité, cela se traduit par un seul langage et une seule compétence à maintenir. Ce point est fondamental.

Faire simple, vite et bien, voilà le principal secret d'une réussite qui semble actuellement échapper aux adeptes de la complexité hétérogène.

Il est indispensable de comprendre que *Visual Basic* est à la base d'un nouveau paradigme du développement graphique d'applications *Windows*. Il remplace progressivement les langages de *macro-commandes* et se retrouve associé à la plupart des produits de bureautique. Il est de plus en plus souvent directement mis en œuvre par des utilisateurs avertis. Son usage conduit alors à une continuité directe entre l'exploitation d'un contexte bureautique personnel et l'application stratégique desservant plusieurs milliers d'utilisateurs. Son champ d'utilisation couvre 99% des applications de gestion (*Visual Studio* se charge du reste). Aux USA, où des magazines entiers lui sont consacrés, 67 % des professionnels qui utilisent *VB* dans des applications stratégiques, maîtrisent par ailleurs *C++*. Son usage est un choix d'efficacité.

ANALYSER LES POSSIBILITES DE LA BUREAUTIQUE COMMUNICANTE (VBA, OLE)

La suite de ce chapitre s'applique uniquement au graphique sous Windows. D'ailleurs en existe-t-il un autre qui lui soit économiquement comparable ? Sans les atouts natifs de ce système désormais universel, le mode graphique, en ce qui concerne les applications de gestion, aurait plus d'inconvénients que d'avantages. De simples facilités comme les « copier-coller » et les « glisser-déposer » ont révolutionné la manipulation de l'information.

L'essentiel se situe dans la parfaite intégration des solutions complémentaires offertes par plusieurs centaines d'éditeurs indépendants. Pour comprendre l'incroyable puissance globale de cette solution, il faut considérer entre autres :

- la facilité des échanges de données entre applications (un cauchemar avant *Windows*) ;
- l'intégration des multiples documents par l'intermédiaire des liens OLE ;
- l'automatisation des processus de manipulation inter-applications et de personnalisation par *VBA (Visual Basic pour Applications)* ;
- l'incidence de la standardisation sur les temps de formation et sur la courbe d'utilisation réelle des fonctionnalités des produits.

Un nombre croissant d'utilisateurs intermédiaires maîtrise la bureautique classique et les subtilités des macros. Dans les projets stratégiques, ces connaissances permettent une transition vers la bureautique communicante.

Les outils comme *Word*, *Excel*, *Access* combinent un langage fédérateur *VBA (Visual Basic for Application)* et des capacités d'accéder, sans expertise exceptionnelle, aux données corporatives (ODBC, OLE). Ces nouvelles possibilités rendent les utilisateurs plus autonomes et exigeants. Une telle osmose entre l'organisation et la technique n'est pas neutre et elle a un impact évident sur la planification et l'évaluation.

Une part non négligeable de l'effort de spécification et de recette se reporte sur la maîtrise d'ouvrage.

RESERVER LES SERVICES DE SPECIALISTES POUR ASSURER L'ASSISTANCE PERMANENTE

Le recours à l'expertise est toujours favorable sur le plan économique, les meilleurs développeurs ne l'ignorent pas. L'investissement que représente la présence d'un expert praticien est en général immédiatement rentabilisé par l'accroissement de productivité obtenu grâce à ses conseils. Il faut aussi comptabiliser le coût des erreurs qu'il permet d'éviter.

PUBLIER ET COMMENTER LES NORMES DE DEVELOPPEMENT

Nul n'est censé ignorer la loi !

2.4.6. Mise en œuvre d'une politique de mesures et d'actions**MISE EN ŒUVRE D'UN PLAN D'ASSURANCE QUALITE AU NIVEAU REALISATION**

- Expliquer les principes dès le début du projet.
- Publier une normalisation formelle.
- Organiser une conférence technique sur les principes fondamentaux.
- Inciter la communication entre les développeurs dès le départ.
- Organiser les revues en préalable aux Focus (examen critique de la conception et du code).

SEANCE DE TRAVAIL, VALIDATION PERMANENTE PAR L'UTILISATEUR

La validation permanente, comme son nom l'indique, est réellement continue. Le prototypage actif implique physiquement l'utilisateur qui manipule lui-même l'application en cours de construction.

2.4.7. Techniques orientées planning**ORGANISER UN ETAT DE LIVRAISON PERMANENTE**

Réaliser dès le début du prototypage une application techniquement fiable dont on élargit les fonctionnalités tout en préservant cette fiabilité (*figure 4*).

Ce principe est très important, car il rend possibles les concepts de « qualité permanente » et de « livraison permanente ». Une telle maîtrise est loin d'être courante dans les équipes conventionnelles de développement.

La clé de la réussite⁵ consiste donc à maintenir constamment l'application dans un état connu, livrable. *Microsoft* a nommé cette technique : les jalons ZD (zéro-défaut). Dans ce contexte de qualité, la livraison réelle est simplement le dernier jalon du projet.

Dans une réunion de type Focus on présente concrètement une application incomplète, mais stabilisée et en évolution. Chacun peut alors, de l'utilisateur de base au dirigeant, avoir une vision complète du produit en cours de Construction, le critiquer et contribuer à son perfectionnement. C'est à la fois une opération de relation publique et un travail général de validation.

Dans l'approche RAD, aucune technique n'est superficielle. La sécurité offerte par une vraie visibilité du pilotage d'un projet est déjà pour un dirigeant un élément concret et indispensable à sa tranquillité d'esprit.

Pour les utilisateurs qui effectuent les démonstrations, le Focus est un des instants privilégiés où ils se sentent vraiment propriétaires de l'application.

Au cours du Focus, plus l'informaticien est en retrait et plus son efficacité et la puissance de sa technique deviennent évidentes.

Une application de qualité, permettant une livraison « par étapes » ou « en fonctionnalités réduites » (figure 5) nécessite une évaluation régulière.

Le contrôle d'assurance qualité devrait être le fait d'un groupe d'assistance technique utilisateur.

La plupart du temps, cette fonction, ainsi que la documentation utilisateur, fait partie intégrante des tâches planifiées du projet. Pourtant, ces tâches devraient dépendre globalement d'une assistance à la maîtrise d'ouvrage.

Dans les projets importants, il est généralement admis que, pour un SWAT, un documentaliste et une ressource d'assurance qualité sont nécessaires. A chacun son métier, et les projets seront bien terminés.

GERER LES INDICATEURS QUANTITATIFS ET QUALITATIFS D'AVANCEMENT

Le suivi consiste, à partir de l'évaluation, à analyser la réalité observée pour tirer les conséquences des écarts éventuels et apporter les correctifs appropriés. Les outils modernes et conviviaux de gestion de projet comme MS Project automatisent ces aspects.

PROCEDER A DES REVUES DE PROJET (AVANT FOCUS)

La planification et l'ordonnancement de l'engagement des ressources sont la base d'une gestion de projet efficace. Le suivi nécessite une attention particulière et l'usage d'un outil spécialisé. Il doit permettre une gestion aisée jusqu'au niveau le plus fin de détail (tâche, heure ou demi-journée). Il fournit automatiquement des rapports d'avancement, suffisamment complets pour satisfaire une direction qui doit apprendre à se contenter de l'essentiel. Les activités de contrôle sont liées à des livraisons de versions de développement du produit aux responsables de l'Assurance Qualité.

Laisser toujours l'utilisateur manipuler



- 1 Présenter les améliorations apportées
- 2 Laisser l'utilisateur manipuler
- 3 Prendre note de toutes les remarques
- 4 Modifier immédiatement les détails
- 5 Négocier les demandes lourdes
- 6 Remplir la fiche d'évolution prototype
- 7 Fixer les objectifs du RDV suivant

" Commencer par fixer les fonctionnalités primordiales "

Figure 6. — Protocole d'une séance de prototypage

ORDONNANCER LES ACTIONS CORRECTIVES

Un contrôle de l'avancement des travaux est effectué à la semaine. La ventilation des heures travaillées sur chaque tâche élémentaire est réelle. Chaque membre de l'équipe indique le pourcentage de réalisation et estime le reste à produire. L'analyse de ces deux nombres peut être édifiante.

⁵ La répétition de cette phrase est justifiée par l'importance du concept.

Le pilote de projet organise des réunions de contrôle :

- la fréquence est hebdomadaire en début de projet ;
- en phase de réalisation un rythme d'une réunion toutes les deux semaines peut s'avérer suffisant ;
- la durée n'excède pas une heure par équipe.

Un plan classique pour ce type de réunion consiste à :

- faire le point sur l'avancement des travaux ;
- rappeler les points d'action levés lors de la dernière réunion ;
- citer brièvement les solutions apportées ;
- étudier les nouveaux problèmes, exigences et/ou risques ;
- formaliser des solutions immédiates ;
- ouvrir et individualiser des points d'action.

Le secrétaire de projet produit directement le compte-rendu et le diffuse immédiatement ou, au plus tard, dans le cours de la journée.

◇ **Retard et dérive**

Il est de la plus grande importance de comprendre la différence entre retard et dérive.

◇ **Retard ou dérive en phase de Design**

Dans les grands projets, ce type de problème peut survenir dès la phase de Design. Il suffit pour arriver à ce point de réunir simultanément une absence de direction dans le projet et des interlocuteurs ne disposant pas de l'information minimale ou du pouvoir de décision pour stabiliser l'expression des exigences.

Il peut se passer plusieurs mois avant que les indicateurs d'avancement du projet ne mettent clairement en évidence la situation.

◇ **Retard ou dérive en phase de Construction**

C'est en phase de Construction que les dérives majeures se produisent le plus généralement. Elles sont toujours la conséquence des points suivants :

- négligence des techniques d'assurance qualité,
- non-respect des jalons zéro-défaut,
- omission de la validation permanente,
- refus ou oubli des revues de code,
- absence ou report de Focus.

Les principales sources de dérives proviennent d'une carence de validation en fin de phase de Design.

Si le Design n'a pas produit un modèle de données stable, des modifications à répétition se propagent à travers les traitements et fragilisent les modules. Dans le pire des cas, c'est la structure même des blocs de traitements qui est affectée. Des groupes d'écrans entiers sont déstructurés. Ensuite, chaque modification, même mineure, entraîne inexorablement de multiples répercussions majeures dans la plupart des procédures. A ce point, la dérive ne peut plus être contenue sans une reconstruction totale du code.

◇ **Réduction du risque de dérapage**

Les dérives ne surgissent pas par un hasard malheureux à la veille d'un Focus. Les dérapages surviennent souvent lorsque les membres de l'équipe commencent à se préoccuper de fonctionnalités tangentielles ou marginales n'appartenant pas au corps central du produit.

Augmenter les ressources est une solution parfois efficace, mais qui s'avère le plus souvent être un piège. L'ajout inopportun de nouvelles ressources à un projet peut produire un ralentissement du travail en cours.

2.4.8. Focus

PLANIFICATION DES FOCUS

Les techniques de conception en vue de modifications, mises en œuvre lors de la phase de Design, avaient pour but de faciliter l'évolution ultérieure de l'application. En phase de Construction il existe des pratiques dont le but est similaire. L'ensemble de ces techniques a pour finalité de permettre une évolution continue depuis la conception, la réalisation et jusqu'à la maintenance, prolongement naturel de la vie d'une application.

Dans un développement RAD, la Construction est intégralement structurée selon les principes de l'itération incrémentale.

Dès le début de cette phase, un plan d'évolution du prototype est précisé. Il définit et segmente l'ensemble des fonctionnalités à produire, lors de chaque borne de validation, appelée Focus. La segmentation peut être verticale ou horizontale :

- une segmentation verticale repose sur un simple découpage en modules ou en écrans ;
- une segmentation horizontale se fonde sur la notion de priorité des fonctions et impose souvent que l'ensemble des modules soit mis en chantier simultanément, mais à un niveau de fonctionnalité limité.

Si l'un des membre du SWAT fait défaut ou accuse un retard, l'ensemble de l'équipe doit compenser pour assurer la tenue du Focus. Dans le pire des cas, trois fonctionnalités au lieu des quatre prévues sont présentées dans certains écrans, mais les dix écrans sont présents.

PREPARATION DU FOCUS

Le Focus est une « grand-messe » où l'application prouve qu'elle a atteint un niveau précis d'avancement et de qualité de développement qui mérite d'être présenté. Un Focus représente un événement attendu qui offre une visibilité concrète à toutes les parties concernées par l'application.

Avant d'aboutir à l'apparente sensation de réussite et de facilité qui doit dominer un Focus, un travail considérable de validation a été préalablement effectué :

- L'assurance qualité fonctionnelle aura été validée par les équipes opérationnelles d'utilisateurs.
- L'assurance qualité technique aura été vérifiée lors de revues de code et d'essais d'intégration.

Un Focus est essentiellement une affaire de validation. Pour présenter un minimum d'intérêt, le nombre de fonctionnalités additionnelles incluses dans l'application à chaque Focus doit être raisonnablement élevé. Dans un projet où la Construction s'étend sur deux mois, il est possible d'envisager un Focus toutes les deux ou trois semaines (tableau 3). La planification de Focus trop rapprochés est sans intérêt et pénalise lourdement le développement.

Il faut compter un minimum de deux jours de charge par personne et par Focus à réaliser.

La notion de Focus couvre en pratique l'enchaînement de plusieurs étapes extrêmement contraignantes pour le SWAT.

Les Focus ne se planifient pas à dates fixes ou à intervalles réguliers. Le facteur décisif pour fixer la cadence des Focus doit être le nombre de fonctionnalités additionnelles attendues.

Concrètement, un Focus débute par la planification d'une charge de travail à réaliser dans un délais précis (de 10 à 20 jours).

Participent au Focus les groupes d'utilisateurs impliqués dans la spécification et la validation, les membres du SWAT, les dirigeants intéressés à divers titres. Outre le travail de développement, la planification comprend la préparation du Focus, sa réalisation et le laps de temps nécessaire à l'exploitation des informations que cette manifestation permet de dégager.

L'organisation de la communication préalable au Focus est du ressort de l'animateur.

Tableau 3. — Nombre de Focus idéal

Durée du projet en jours ouvrés	30	60	90	120
Nombre moyen de Focus	2	3	4	5

VALIDATIONS AVANT FOCUS

Quelques jours avant la date fixée pour le Focus, en fonction :

- du niveau de maturité du SWAT,
- de l'avancement validé,
- de la qualité du code,
- le coordonnateur technique stoppe les développements.

Les utilisateurs ayant participé aux séances de spécification et de validation permanente sont alors impliqués dans une étape de vérification de la qualité fonctionnelle. Le maximum de corrections de divergences visuelles ou mineures doit être effectué. En général, la validation permanente ramène cette tâche à une simple formalité. Cette étape cesse au plus tard deux jours avant le Focus.

Chaque membre du SWAT focalise alors sur l'amélioration technique qualitative de sa propre production. Si les normes de codage publiées et acceptées ont été respectées, une journée suffit pour vérifier une production de deux semaines. Le niveau de qualité requis peut être très variable d'une application à une autre selon sa typologie.

CMM : Les Revues par les pairs visent à éliminer tôt et efficacement les défauts des produits. La revue par les pairs est une méthode indispensable et efficace utilisée en ingénierie de produits logiciel. Elle consiste en une série de vérifications croisées dont le but est de valider la qualité et la conformité technique du produit.

L'équipe procède alors à des revues croisées nommées selon la terminologie du CMM « Revues par les pairs ». Cette étape prend fin lorsque la qualité de l'application atteint un niveau suffisant en :

- robustesse,
- normalisation,
- documentation,
- conformité aux exigences fonctionnelles.

Le membre du SWAT chargé de l'intégration des modules réalise alors cette opération. Il s'attache tout particulièrement à tester les interfaces applicatifs mettant en œuvre des données communes (externes, globales).

Les participants des séances de spécification et de validation permanente sont alors rappelés pour la dernière étape de vérification de la qualité fonctionnelle et des cheminements.

L'intervention des utilisateurs est indispensable, car ils devront manipuler l'application durant le Focus.

A ce point, seules les divergences bloquantes et majeures sont corrigées. Cette opération impliquant un minimum de tests de non-régression et une étape complète d'intégration. Une demi-journée avant le Focus, l'heure n'est plus aux améliorations cosmétiques.

Durant le Focus, les rapporteurs devront consigner les éventuels incidents et observations qui ne manqueront pas d'être émises par une nombreuse assistance.

Les personnes ne participant pas aux spécifications ont souvent un avis a posteriori et parfois une vision plus profonde ou plus précise des évolutions du métier. Ces ressources étant le plus souvent des décisionnaires, leurs recommandations doivent être étudiées sous un angle de vision prospective. C'est d'ailleurs souvent cette absence de vision qui confine une application dans un cadre strictement opérationnel alors qu'elle aurait pu revêtir une dimension stratégique.

L'intervalle entre deux Focus ne doit pas être assimilé à un effet tunnel. Le Focus est une affaire de visibilité externe ponctuelle et non un moyen de validation interne. Entre deux Focus, chaque concepteur-développeur est régulièrement en contact avec un ou plusieurs utilisateurs significatifs qui participent en direct à la spécification du détail des fonctionnalités. Cette technique de validation permanente est la vraie réponse du RAD au fameux manque de communication baptisé « effet tunnel ».

REPERTORIER LES RETARDS, ANALYSER LES DERIVES

Les retards s'accumulent journalièrement, d'où la nécessité, pour les maîtriser, de contrôler l'avancement du projet à un niveau de granularité très fin (par module ou même par fonction).

Pour que ce contrôle représente concrètement la situation de l'application, il comprend un volet qualité. L'avancement est représenté par un quantitatif constitué d'une liste des fonctionnalités et d'une densité de divergences relevée lors des séances de validation ou lors des Focus. Les activités de contrôle doivent être liées, si possible, à des livraisons de versions du produit aux responsables de l'Assurance Qualité.

2.4.9. Prototypage « RAD »

Le prototypage ACTIF (méthode RAD) combine :

- la spécification détaillée, son codage et sa documentation,
- la validation permanente,
- les tests unitaires,
- les tests d'enchaînement.

Ce principe fondé sur la collaboration des utilisateurs significatifs permet de créer naturellement et progressivement un alpha site.

2.4.10. Documentation utilisateur contextuelle

Une documentation moderne doit être en ligne et contextuelle. En ligne, elle concerne les principes de fond. L'aide contextuelle est immédiatement visible et associée à l'action ou au champ sur lequel l'utilisateur est positionné.

2.4.11. Mise en place du site pilote

Ce qui suit est fondamentalement dépendant de l'application ainsi que du type et du nombre d'utilisateurs. Le site pilote doit concerner un nombre limité mais significatif d'utilisateurs. Les données sont réelles, par contre, leur exploitation n'est pas réellement opérationnelle. Pour cette raison, l'ancien système est maintenu.

2.4.12. Validations par l'ensemble des utilisateurs

Une procédure d'arbitrage entre Maître d'Œuvre et Maître d'Ouvrage est prévue pour déterminer les modalités de prise en compte des modifications ou des améliorations souhaitées par rapport aux exigences fixés lors des phases précédentes et du prototypage. Cette procédure est particulièrement nécessaire dans des environnements en constante évolution (commercial, aide à la décision, réorganisation).

REALISER LA PREMIERE PHASE DE FORMATION

Avec la méthode RAD, un groupe déterminé d'utilisateurs participe en permanence au développement. Pour partie d'entre eux (2/3), leur choix, outre la motivation qu'ils éprouvaient initialement, a été fonction de leur représentativité vis-à-vis des autres utilisateurs. Le tiers restant peut être composé de sceptiques. Dans de petits projets, ces utilisateurs sont formés en vue de transmettre la connaissance à leurs collègues. Dans les projets importants, la fonction de formation est encadrée par des professionnels. Les utilisateurs participant au RAD sont considérés comme des ressources de support. Ils fournissent les explications et les raisons ayant conduit aux fonctionnalités retenues.

RENDRE LE SITE PILOTE OPERATIONNEL

Le site pilote doit concerner un nombre limité mais significatif d'utilisateurs. Les conditions d'exploitation sont réelles.

2.5. Phase FINALISATION

2.5.1. Stratégie de recette homologation

Les phases de Cadrage, puis de Design donnent lieu à des recettes de fonctionnalités générales. Le RAD est en lui-même porteur des validations au niveau unitaire. La vérification d'aptitude du produit est donc acquise dès lors que toutes les fonctionnalités sont présentes et opérationnelles. Il suffit de formaliser cet état par un document constatant le fonctionnement du système point par point.

En RAD, les tests unitaires et d'enchaînement sont permanents durant la phase de développement.

Des tests complets de recette seront néanmoins nécessaires dans le cadre de systèmes complexes, importants ou impliquant un grand nombre d'utilisateurs. C'est le principe des chemins fonctionnels classiques⁶. L'ensemble des parcours envisagés constitue une « couverture ».

⁶ Ensemble des cheminements utilisés par les utilisateurs pour réaliser complètement la totalité de leurs tâches.

2.5.2. Stratégie de distribution

LIVRAISONS EN FONCTIONNALITES REDUITES

La livraison en fonctionnalités réduites offre l'usage immédiat de fonctionnalités indispensables et permet d'accélérer le retour sur investissement dans les conditions suivantes :

- une collaboration sans faille des utilisateurs ;
- un environnement technique et organisationnel de livraisons répétitives efficace et peu onéreux.

2.5.3. Stratégie de déploiement

Les diagrammes de Déploiement facilitent la livraison des composants.

IMPLANTATION DES BETA SITES

Cette technique permet de faire utiliser (ou tester) une application « relativement testée » (ou relativement non testée) par ses utilisateurs. Lesquels rapportent aux informaticiens les problèmes leur apparaissant. Elle implique les prérequis de la « livraison en fonctionnalités réduites ». La technique des bêta sites est un principe efficace, mais dangereux, qui doit être étudié au cas par cas.

INDUSTRIALISATION DE L'APPLICATION

Dépendant généralement de l'organisation plus que de la typologie du projet, l'industrialisation peut être un processus long, émaillé de nombreuses phases de tests et d'homologation effectuées par des entités externes au développement. Les applications finalisées sont ensuite livrées à un service chargé du soutien des utilisateurs, de l'assistance technique, de la maintenance corrective et évolutive de premier niveau.

2.5.4. Stratégie de capitalisation de l'expérience

Les difficultés mis en évidence par le projet complèteront, si nécessaire, le répertoire des risques. Plus largement, l'essentiel est d'offrir un moyen structuré de se questionner, d'influer sur la performance et surtout de mesurer objectivement la qualité du changement. Cette qualité dépasse alors le concept subjectif, pour devenir une valeur d'investissement objectivement mesurée. « La mesure dans tous les sens du terme est au cœur des processus d'amélioration » (Ishikawa). Les enquêtes de satisfaction auprès des utilisateurs et les ratios d'erreurs résiduelles sont des grandeurs mesurables. Le CMM, à chacun de ses niveaux, pose des questions concrètes et offre des réponses qui le sont tout autant.

Il ne suffit pas de s'imaginer être bon, malgré les retards et les imperfections. Il faut mesurer objectivement l'efficacité. C'est d'ailleurs le seul moyen d'atteindre l'efficacité et de la conserver.

2.5.5. Maintenance et évolution

La maintenance et l'évolution sont des parties intégrantes du cycle de vie du système. Prolongement naturel de la réalisation RAD, la maintenance est cependant différente de l'activité de développement dont elle hérite les lacunes. Le fait de concevoir « en vue de modifications » a un impact évident sur la maintenance. De plus, le cycle évolutif du RAD ainsi que ses techniques de prototypage, de lotissement et d'état de livraison permanente accentuent à l'extrême cette possibilité intéressante de considérer la maintenance applicative comme prolongement naturel et intégré du cycle de développement.

2.6. Processus optimisé « petit projet »

100	%	Phases, étapes et tâches du processus RAD2 simplifié
6		1 Phase INITIALISATION
	4	1.1 Immersion coordonnateur technique
	0,5	1.2 Définition du plan de communication
	0,5	1.3 Entretien propriétaire (maîtrise d'ouvrage)
		1.3.1 Validation des Visions globales
		1.3.2 Environnement de réunion
		1.3.3 Ordonnancement des invitations et des convocations
	0,5	1.4 Réunion de lancement
		1.4.1 Partie publique (toutes les personnes concernées)
		1.4.1.1 Limitation périmètre initial du projet
		1.4.1.2 Définition des groupes de travail
		1.4.1.3 Proposition planning de Cadrage
		1.4.2 Groupes de travail (personnes impliquées)
		1.4.2.1 Ouverture des Actions (individualisation des travaux)
	0,5	1.5 Validation planning de Cadrage
9		2 Phase CADRAGE
	0,5	2.1 Cadrage Stratégique
	5,5	2.2 Cadrage - Fonctionnel
		2.2.1 Réaliser entretiens de groupe
		2.2.2 Engager la phase de définition globale des Exigences
		2.2.3 Modéliser en direct avec un vidéoprojecteur
		2.2.3.1 Modélisation des données (MCD global)
		2.2.3.2 Modélisation des traitements (hiérarchie)
		2.2.4 Etablir la priorité des Exigences, évaluer, planifier.
	0,5	2.3 Cadrage - Technologique
		2.3.1 Rapprocher les Exigences des technologies existantes
		2.3.2 Modéliser l'architecture technique
		2.3.3 Garantir la faisabilité technique et la montée en charge
	1	2.4 Cadrage - Organisationnel
		2.4.1 Prévoir les impacts organisationnels
	0,5	2.5 Cadrage - Contraintes
	1	2.6 Validation des Cadrages (modèles globaux)
23		3 Phase DESIGN
	3	3.1 Modélisation détaillée des communications
	5	3.2 Modélisation globale des traitements
	10	3.3 Modélisation détaillée des données
	2	3.4 Recherche de solutions
	1	3.5 Planification détaillée de la Construction
	2	3.6 Validation de fin de Design
50		4 Phase CONSTRUCTION
	1,5	4.1 Environnement de Construction + normes + facilitation
	1,5	4.2 Politique de mesures et d'action (mise en œuvre + suivi)
	8	4.3 Prévision + Organisation des Focus
		4.3.1 Organiser un état de livraison permanente
		4.3.2 Fixer les dates des Focus et des Jalons Zéro Défaut
		4.3.3 Répertoire des retards, analyser les dérives
	1	4.4 Mettre en place le site pilote en configuration de test
	36	4.4 Prototypage RAD (itération)
		4.4.1 Spécification détaillée, Réalisation, tests, alpha site.
		4.4.2 Documentation utilisateur contextuelle.
	2	4.5 Assistance initiale aux formateurs (MOA)
12		5 Phase FINALISATION
	1	5.1 Organiser les stratégies de recette, livraison, déploiement
	8	5.2 Recette finale
	1	5.3 Focus final
	2	5.4 Transmission de l'application (cut-over) ou déploiement

Cette représentation simplifiée du processus RAD2 peut être utilisée :

- en théorie pour appréhender de façon plus aisée la structure du processus ;
- en pratique, pour organiser la conduite d'un projet de taille intermédiaire.

Les pourcentages des deux premières colonnes sont représentatifs d'une planification de projet à horizon temporel de 90 ou 120 jours.

Ces chiffres ont pour origines des statistiques de projets, l'expérience de l'auteur et ont été validés en commission AFITEP (IM-CP-Processus).

Je remercie pour leur participation à la formalisation de ces points : Christian Alglave et Alain Cathiard.

3. Stratification du choix de solution

Dans un projet classique, **7 couches**, éventuellement **externalisables**, composent l'ensemble de la solution.

Le responsable du projet, doit s'ouvrir à toutes les **possibilités concurrentes** et les envisager comme des **moyens** de remplir sa mission

L'objectif de cette mission est d'obtenir la « **qualité raisonnable** » : la meilleure **qualité** au meilleur **coût**, dans le meilleur **délai** en fonctions des **contraintes** imposées à l'application à produire.

3.1. Couches externalisables

Les 7 couches d'une possible externalisation sont détaillées dans la *figure 7* représentant la structure d'un processus « ouvert » (progiciel, composants, infogérance, etc.) de définition des exigences et d'évaluation des possibilités d'obtention de la solution optimale.

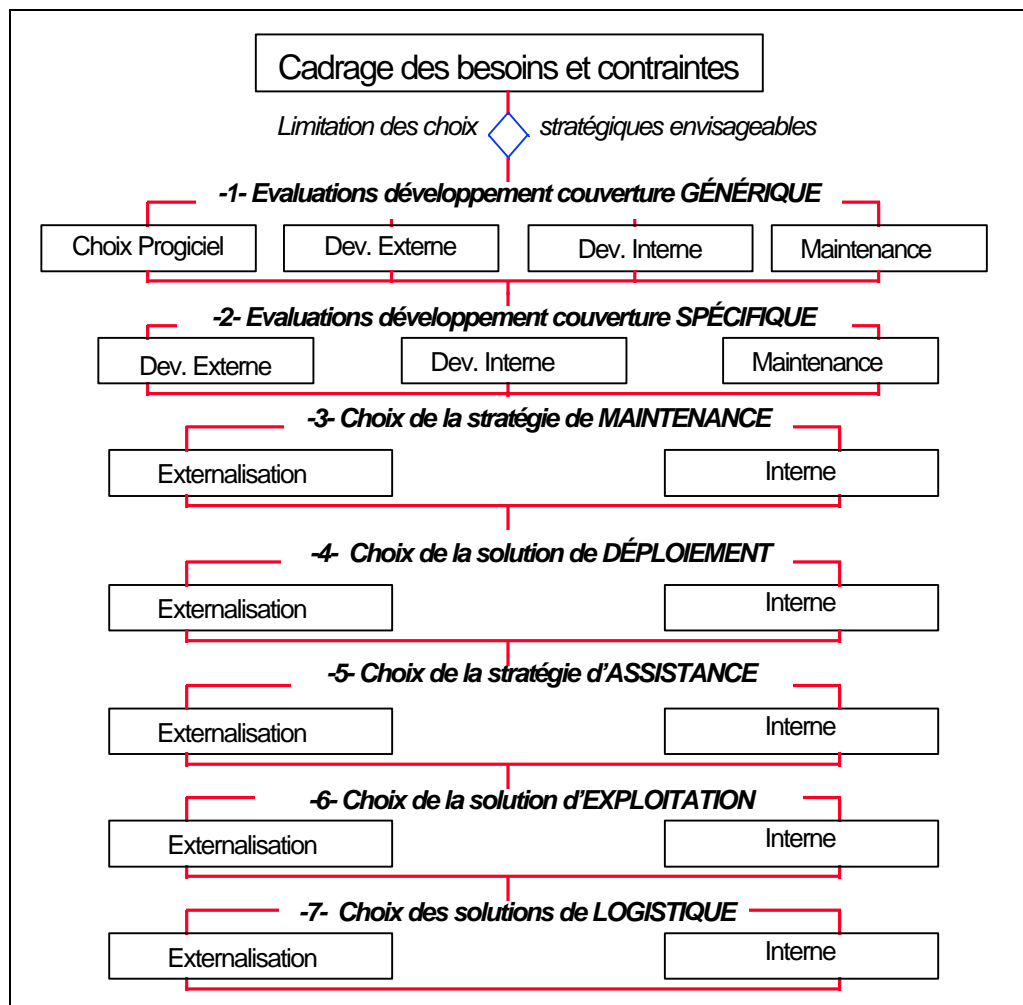


Figure 7. — Structure d'un processus ouvert de choix de solution

3.2. Exemples d'éléments externalisables

Les 7 couches structurelles de la solution « ouverte » se détaillent en de nombreux composants qui peuvent se combiner en termes de solutions externalisées ou réalisées en interne :

1. Développement du générique « progicialisable ».
2. Développement du spécifique non « progicialisable ».
3. Maintenance corrective.
4. Maintenance adaptative (obligatoire ou réglementaire).
5. Maintenance évolutive.
6. Installation matériel serveur.
7. Installation matériel client.
8. Déploiement applicatif sur serveur.
9. Déploiement applicatif sur client.
10. Gestion des télécommunications.
11. Assistance poste client aspects matériels.
12. Assistance poste client aspects fonctionnels.
13. Exploitation transactionnelle.
14. Exploitation batch classique.
15. Sécurité et sauvegarde.
16. Evolution matériel.
17. Logistique (fabrication, expédition, etc.).

Dans le cas où une externalisation de tout ou partie de la solution est envisagée, il est utile de disposer d'un processus complémentaire à celui qui guide à la phase classique de Cadrage.

3.3. Processus général « externalisation »

- 1 Détermination du niveau d'externalisation**
- 2 Définition du rôle et des responsabilités des intervenants dans la solution**
 - 2.1 La maîtrise d'ouvrage
 - 2.2 La maîtrise d'œuvre
 - 2.3 Les acteurs externes (prestataires, sous-traitants)
 - 2.3.1 Acteurs liés aux produits
 - 2.3.2 Acteurs liés à la maintenance
 - 2.3.3 Acteurs liés à l'exploitation
- 3 Etude des prestataires externes**
 - 3.1 Recensement des prestataires significatifs
 - 3.2 Détermination des critères de sélection
 - 3.2.1 Adéquation technique
 - 3.2.2 Assurance économique
- 4 Rédaction du document des principes généraux**
 - 4.1 Architecture applicative
 - 4.2 Architecture technique et communication
- 5 Consultation générale de faisabilité**
- 6 Analyse des réponses et contacts directs**
- 7 Limitation de la liste pour consultation détaillée**
- 8 Consultation détaillée avec soumission**
 - 8.1 Rédaction cahier des charges « externalisation »
 - 8.2 Rédaction d'un PAQ « externalisation »
 - 8.2.1 Obligation de moyens
 - 8.2.2 Obligation de résultats (indicateurs)
- 9 Rédaction questionnaire d'externalisation**
 - 9.1 Partie technique
 - 9.2 Partie financière
- 10 Analyse des réponses et rédaction de la synthèse**
- 11 Choix interne de la solution (comité de pilotage)**
- 12 Rédaction du contrat (service juridique)**
- 13 Négociation externe (fournisseur / solution retenue)**
- 14 Signature du contrat ou retour à une autre solution**

3.4. Processus spécialisé « progiciel »

2. Processus de cadrage « progiciel »
2.1 CAHIER des CHARGES (expression des Exigences)
2.1.1 Communication, planification, organisation du projet
2.1.2 Spécification des Exigences (actuelles + envisageables)
2.1.2.1 Etude des Exigences fonctionnelles
2.1.2.1.5 [Modèles selon typologie de l'application]
2.1.2.2 [Etude des autres Exigences à considérer]
2.1.2.2.1 Impacts organisationnels
2.1.2.2.2 Risques et sécurité
2.1.2.2.3 Contraintes et orientations imposées
2.1.2.2.4 Obligations externes
2.1.2.2.5 Critères technologiques
2.1.3 Rédaction du Cahier des Charges « progiciel »
2.1.4 * FOCUS (présentation et validation des Exigences)
2.2 CONSULTATION (choix des Solutions)
2.2.1 * Analyse du marché (progiciels et solutions)
2.2.1.1 Formalisation des critères de sélection
2.2.1.2 Elimination basée sur la couverture fonctionnelle
2.2.1.3 Elimination basée sur les aspects organisationnels
2.2.1.4 Elimination basée sur les aspects techniques
2.2.1.5 Visites de sites de référence (clients), benchmarking
2.2.1.6 Evaluation initiale et obtention d'une liste réduite
2.2.1.7 Obtention des éléments économiques (licences, coûts)
2.2.1.8 Solutions retenues et choix de poursuite de l'opération
2.2.2 [Validation détaillée par prototypage (si nécessaire)]
2.2.2.1 Préparation des scénarios significatifs et cas critiques
2.2.2.2 Réalisation / obtention de prototypes ou de maquettes
2.2.2.3 Déroulement des scénarios et validation de conformité
2.2.3 Rédaction Cahier de Consultation, CONSULTATION
2.2.3.1 Comparaison des couvertures fonctionnelles
2.2.3.2 [Comparaison des solutions techniques]
2.2.3.3 [Comparaison des impacts organisationnels]
2.2.3.4 Identification des limites et des contraintes
2.2.3.5 Production du comparatif et des préconisations
2.2.3.6 FOCUS (Présentation du choix de solution), CHOIX
2.3 ASSISTANCE MISE EN ŒUVRE (déploiement)
2.2.4.1 Estimation et Planification
2.2.4.2 Aspect architecture technique
2.2.4.3 Aspect mise à niveau des données
2.2.4.4 Modalités d'accompagnement du changement

Cette démarche peut aussi s'appliquer à une recherche de composants, d'objets ou de solutions d'aide au développement.

4. Bibliographies et références

4.1. Bibliographie RAD, Conduite de projet

Martin James, *Rapid Application Development*, Macmillan, 1991.

Vickoff Jean-Pierre, *RAD - Développement Rapide d'Applications, Méthodes et outils, les règles à respecter dans le développement d'application client-serveur*, MGI, 1995, puis Macmillan, 1996.

Vickoff Jean-Pierre, *REENGINEERING, Vite fait bien fait, Le Paradigme du futur immédiat, QI*, 1998.

Vickoff Jean-Pierre, *Réingénierie du développement d'application, Méthodes, processus, techniques et pratiques de conduite de projet sous contraintes*, Gartner Group, 1999.

Boehm B. & Bose P. *A Collaborative Spiral Software Process Model*, USC, 1994

Bouchy S. *L'Ingénierie des systèmes d'information évolutifs*, Eyrolles, 1994

Clark B. *The Effects of software process maturity on software development effort*, USC, 1997

Englewood C. *JAD the Group Session, Approach to system design*, Prentice Hall, 1991.

Egyed A. & Boehm B. *Telecooperation Experience with the WinWin System*, USC, 1998

Henry A. & Monkam-Daverat L. *Rédiger les procédures de l'entreprise*, Les Éditions d'organisation, 1995.

Mc Carty J. *54 Règles d'or pour un grand logiciel*, Microsoft Press, 1997.

McConnell S. *Stratégie de développement rapide*, Microsoft Press, 1996.

Panet G. & Letouche R. *Merise 2, modèles et techniques Merise avancés*, Les Éditions d'organisation, 1994.

Yourdon E. *Modern Structured Analysis*, Englewood Cliffs, 1989.

4.2. Bibliographie Objet

Budd T. *La Programmation par objets*, Edition Addison-Wesley, 1991.

Jacobson I. *Le Génie logiciel orienté objet*, Addison-Wesley, 1993.

Lai M. *UML - La Notation unifiée de modélisation objet*, InterEdition, 1997.

Lemay L. & Perkins C. L. *Le Programmeur JAVA*, Macmillan, 1996

Muller P. A. *Modélisation Objet avec UML*, Eyrolles, 1997.

Rumbauch J. *OMT Modélisation et conception orientées objet*, Prentice Hall Masson, 1996.

Vauquier D. *Développement orienté objet, principes, processus, procédés*, Eyrolles, 1993.

4.3. Bibliographie Management, Qualité

Ballay J-F. *Capitaliser et transmettre les savoir-faire de l'entreprise*, Eyrolles, 1997.

Bartoli A. *Communication et organisation, pour une politique générale cohérente*, Les Éditions d'organisation, 1994.

Beaudoin P. *La Gestion du changement, une approche stratégique pour l'entreprise en mutation*, Stratégies d'entreprise, 1990.

Champy J. *Reengineering Management*, HarperCollins Publishers, 1995.

Chapman C. & Ward S. **Project Risk Management : Processes, Techniques and Insights**, Edition Wiley, 1997.

- Courtot E. **La gestion des risques dans les projets**, Edition Economica, 1998
- Delafollie G. *Analyse de la valeur*, Hachette, 1991.
- Hammer M. & Champy S. *Reengineering the Corporation, A manifesto for Business Revolution*, Harper Business, 1993.
- Henry A. & Monkam L., *Rédiger les procédures de l'entreprise*, Éditions d'organisation, 1995.
- Laudoyer G. *La Certification, un moteur pour la qualité*, Les Éditions d'organisation, 1993.
- Lamprecht J-L. *ISO 9000 Se préparer à la certification*, AFNOR, 1994.
- Lefebvre C. *Concevoir et conduire un projet de changement*, Les Presses du management, 1997.
- Maders H-P. & Gauthier E. & Le Gallais C. *Conduire un Projet d'Organisation*, Editions d'Organisation, 1998
- Nizet J. & Huybrechts C. *Interventions systémiques dans les organisations, Intégration des apports de Mintzberg et de Palo Alto*, Editions DeBoeck Université, 1998
- Mucchielli R. *L'Interview de groupe*, ESF, 1987.
- Renaud-Coulon A. *La Désorganisation compétitive*, Maxima, 1996
- Richard J. & Paula S. *Delivering Results: Evolving BPR from Art to Engineering*, Texas A&M University, 1998
- Sary P. *La Stratégie de la programmation neurolinguistique dans l'entreprise*, Editions Retz, 1990.
- Wenger E. 1998, *Communities of Practice - Learning, Meaning and Identity*, Cambridge University Press.,

4.4. Divers documents, normes, standards

En ce qui concerne les travaux du SEI sur CMM :

- SEI/CRIM, 1993, Paulk *Modèle d'évolution des capacités logiciel*.
- IEEE Software, 1993, Paulk & M.C. Curtis & B. & Chrissis, M.B. & Weber, C.V. *Capability Maturity Model, Version 1.1*, Vol. 10, No. 4, July 1993, pp. 18-27.
- SEI, 1995, Paulk *The Capability Model : Guidelines for Improving the Software Process*.

En ce qui concerne les normes ISO (Evaluation, Qualité, SPICE) :

- ISO 9001-1994, *Model for quality assurance in design, development, production, installation and servicing*.
- ISO 9000-3-1991, *Quality management and quality assurance standards (Part 3: Guidelines for the application of ISO 9001 to the development, supply and maintenance of software)*.
- ISO 9004-4-1993, *Quality management and quality system elements (Part 4: Guidelines for quality improvement)*.
- ISO 8402 et X 50-125, *Management de la qualité et assurance de la qualité – Vocabulaire*
- ISO/IEC12207-1-1994, *Software life cycle processes (ingénierie du logiciel)*
- ISO/IEC12119-1995, *Software products - Evaluation and test*.
- ISO/IEC 9126-1991, *Software quality characteristics*.
- ISO 9294, *Ligne directrice pour la gestion de la documentation technique du logiciel*

Ainsi que diverses références aux documents suivants :

- Craigmyle, M., and I. Fletcher, *Improving IT effectiveness through software process assessment*, Software Quality Journal, Vol. 2, pp 257-264 (1993).

- Humphrey, W.S., *Managing the Software Process*, Addison Wesley, 1989.
- Kuvaja, P., Simila, J., Krzanik, L., Bicego, A., Koch, G. and Saukkonen, S., *Software Process Assessment and Improvement: The BOOTSTRAP Approach*. Blackwell, 1994.
- Bell Communications Research, Inc., *Quality Program Specification for Surveillance Management Process(SMP) Software (General)*, QPS 88.001, Issue 1.
- AFCIQ-PDL, *Recommandation de plan de développement logiciel*.
- AFCIQ-PAQL, *Recommandation de plan d'assurance qualité logiciel*.
- SCE - Software Capability Evaluation Training Guide / SPA - Software Process Assessment Training Guide, Software Engineering Institute, Pittsburgh Pennsylvania.
- F. Coallier, N. Gammage, A.W. Graydon, Trillium - *Software Process Self-assessment Capability Assessment* , Bell Canada, PI Q008, Issue 4.0, March 1993.

4.5. Principaux WEB et contributeurs

AFAV, Association française pour l'analyse de la valeur, <http://www.afav.asso.fr>

AFITEP, Association francophone de management de projet, <http://www.afitep.fr>

American Society for Quality, <http://www.asqc.org>

Business Processes Ressources Centre, <http://bprc.warwick.ac.uk/bp-gold.html>

Business Process Reengineering Advisory Group,
<http://www.eil.utoronto.ca/tool/list.html>

COCOMO, Constructiv Cost Model,
<http://sunset.usc.edu/CORADMO/coradmo.html>

CORADMO, Constructiv Cost Model,
<http://sunset.usc.edu/CORADMO/coradmo.html>

CRIM, Centre de génie logiciel appliqué de Montréal, <http://www.CRIM.CA/cgla>

DACS, Data Analysis Center for Software (Cost estimation),
<http://mach10.rome.kaman.com/cgi-bin/keylist>

Esprit Project 27599 - RAMSES RAD for MSS,
<http://www.esi.es/ESSI/Reports/All/27599/>

IDEF method, <http://www.idef.com>

IFPUG, International Function Point Users Group, <http://www.ifpug.org>

ISO, Software Process Assessment, <http://www.sqi.gu.edu.au/sc7/wg10/>

SEI, The Software Engineering Institute, <http://www.sei.cmu.edu>

Software Engineering & its Applications (TFGL du CNAM),
<http://web.cnam.fr/TFGL>

Staffordshire University
<http://www.soc.staffs.ac.uk/~cmtrnk/rad/new/course/rad1.htm>

UML, Informations sur UML, <http://www.essaim.univ-mulhouse.fr/uml>

University of California, Davis,
http://sysdev.ucdavis.edu/webadm/document/rad_toc.htm

Pour terminer, je remercie particulièrement les participants assidus des réunions de la commission AFITEP *IM-CP-Processus*. Leur contribution à la formalisation des divers aspects de la conduite de projet, a facilité la rédaction de plusieurs parties de cet ouvrage.

4.6. Index

A

abstraction · 21
abstraction
 niveaux · 20
acquisition · 25
Activex · 28
AFNOR · 47
aide à la décision · 39
animateur
 objectif · 10
applications
 stratégiques · 22
 transversales · 20
arbitrage · 39
architecture
 technique · 8

C

certification · 47
collaboration
 livraison · 40
 prototypage · 38
comité de pilotage · 8, 9, 19, 26, 44
communication
 modèle · 21
 plan · 7, 10, 12
 qualité · 7, 29, 33, 38
contraintes
 de qualité · 18
cycle de vie · 40

D

décomposition · 5
décomposition
 communication · 22
dérive
 causes · 25, 35
 principe · 35
dictionnaire · 21, 22
dictionnaire des données · 22
Disponibilité · 15
dynamique
 axe · 20

E

effet tunnel · 38
engagement RAD · 10
entité-relation · 17, 22
entretien
 de direction · 13
 de groupe · 15

 initial · 10
 RAD · 10
ergonomie · 17, 25
état de livraison permanente · 25
étude d'opportunité · 8

F

feed-back · 26
Focus · 9, 12, 26, 27, 29, 30, 31, 33, 34, 35, 36, 37, 38
Focus
 communication · 29, 33, 36
 pratiques · 30, 36, 37
 principe · 12
 validation · 6, 16, 25, 27
fonctionnalités
 générales · 39
fonctionnalités réduites · 25
fonctionnel
 axe · 20
 cadrage · 10
 cheminement · 6, 16, 39
 conformité · 19
 conformité · 37
 couverture · 8
 dépendances · 25
 domaine · 7
formation
 programme · 39
 simplifiée · 7
 swat · 15
 utilisateur · 15

G

Gane & Sarson · 20
Gartner Group · 7, 46
générateurs de données primaires · 21
groupe d'animation et de rapport · 31
groupe de travail · 12

H

hiérarchie de fonctions · 16
hiérarchie de fonctions
 modélisation · 16
 pratiques · 8

I

immersion · 7
indicateurs · 35
intégrité référentielle · 22
ISO · 10, 47

J

jalons ZD · 27, 33, 35
James Martin · 46

L

lancement officiel · 13
livrable · 27, 33
livraison · 17, 25, 27, 30, 31, 33, 34, 40

M

maintenance
 applicative · 36, 40
 corrective · 40
motivation · 13, 39
mutation
 métiers · 15

N

non-régression · 38
normalisation · 28, 33, 37
normes de programmation · 28

O

outil
 bureautique · 32
 communication · 15

P

PAQF · 18
PAQT · 18
participation
 équilibre · 10
 M.O. · 9
pilote
 site · 39
planification · 5, 11, 25, 26, 27, 30, 32, 37
planification
 de Focus · 36
 lots · 25
 pratiques · 25, 37
Planification
 de projet · 34
points d'actions · 13
présentation
 conférence · 12
 outil · 11
 visibilité · 25
processus
 métier · 20, 40
progiciel · 19, 22
prototypage

RAD · 40
séance · 10
spécifications · 39
validation · 27, 33

prototype
 évolution · 36
 niveau · 25

Q

qualité
 assurance · 18
qualité fonctionnelle · 30, 31, 36, 38
qualité technique · 36

R

rapporteur · 13
rapporteurs · 38
réactivité
 optimisation · 19
 organisation · 22
recette
 finale · 6, 16
 M.O. · 32
 partielle · 39
 test · 39
réingénierie · 10, 18
René Descartes · 4
ressource
 additionnelle · 27, 34
 coordination · 7
 formation · 15
 profil · 15
retard
 principe · 35, 38
 solution · 36
revues de code · 35

S

secrétaire · 35
session plénière · 15, 16
spécialiste-généraliste · 15
spécification
 réunion · 10
SPICE · 47
statique
 axe · 20
structuration
 données · 20
 technique · 21
structure
 de production · 10
SWAT · 10, 15, 21, 27, 30, 31, 34, 37
SWAT
 composition · 15, 16
 culture · 36
 organisation · 21, 27, 29

T**tests**

- de recette · 39
- unitaires · 39

thèmes

- découpage · 15
- minutage · 11
- use-case · 16

travail de groupe · 12**typologie**

- adaptation · 18
- risques · 37

U**use case · 6, 16****utilisateur**

- bureautique · 32
- coordination · 7

engagement · 37

enquête · 40

exigences · 25

formation · 15

groupe · 10

réel · 12

use case · 6

validation · 27, 33

utilisateurs significatifs · 9, 13, 38

V**validation permanente · 26, 29, 33, 38****verrouillage · 26****vidéoprojecteur · 10, 15****visibilité · 18, 22, 25, 26, 27, 28, 29, 33, 36, 38****vision prospective · 38****Visual Basic · 32**

4.7. Figures

Figure 1. — Participation aux groupes de travail	11
Figure 2. — Modèles comparés OLTP /OLAP	23
Figure 3. — Techniques dimensionnelles avec Erwin (Platinum)	24
Figure 4. — Architecture de réalisation, FOCUS de Construction	29
Figure 5. — Prérequis à la livraison par étapes	30
Figure 6. — Protocole d'une séance de prototypage	34
Figure 7. — Structure d'un processus ouvert de choix de solution	42

4.8. Tableaux et listes

Tableau 1. — Niveau de qualité de l'interface graphique	18
Tableau 2. — Nombre de Focus (NF) pour un projet moyen	31
Tableau 3. — Nombre de Focus idéal	37



Jean-Pierre Vickoff est avant tout pilote de projet et concepteur-développeur. Canadien et Français, il s'est spécialisé en Amérique du Nord dans les méthodes de développements sous contraintes.

Consultant en qualité et en performance, il produit des communications pour la presse professionnelle et des rapports pour le Gartner Group. Il est l'auteur de plusieurs ouvrages, dont *RAD* et *Réingénierie du développement d'applications*. Il préside la commission conduite de projet IM-CP de l'AFITEP et intervient auprès des établissements membres de la Conférence des Grandes Ecoles.

Pilote 2010 n'est pas un livre dans le sens littéraire du terme mais un support d'étude, de réflexion et d'action. *Pilote 2010* instrumente l'ensemble des méthodes, pratiques et outils indispensables à la productivité du développement ainsi qu'à la qualité des applications. *Pilote 2010* constitue une source unique de références opérationnelles en matière de :

- Conduite de projet (classique, décisionnel, NET)
- Plan d'Assurance-Qualité du développement
- Pilotage des risques et métrique quantitative des charges
- Méthodes, pratiques, outils de la performance
- Processus d'ingénierie du développement (RAD2-UML)
- Processus spécialisé e-commerce
- Gestion des *Exigences*, des *Validations* et des *Divergences*
- Gestion de la communication et des rapports entre acteurs
- Notations et techniques de modélisation (Merise, Flux, Objet)
- Ingénierie « métier » et accompagnement organisationnel, BPR, MTQS
- Industrialisation, choix de solutions, progiciel, externalisation
- Planification, documentation, plan de tests, gestion de configuration
- Standards d'évaluation et d'amélioration (SEI-CMM/ISO-SPICE)

Ces thèmes sont traités sous une forme directement utilisable.

Pilote 2010 propose aussi dans un format MS Office :

- Un ensemble normalisé de documents de projet
- Un plan d'Assurance-Qualité générique
- Divers processus (RAD2, progiciel, externalisation)
- Un rapport standard de pilotage de projet
- Un logiciel d'évaluation de charge, d'optimisation et de négociation
- Une présentation et une formation à la méthode RAD

Dans un univers de performances, où l'obligation de résultats s'impose à tous progressivement, la rigueur devient l'alliée du changement. Les pilotes de SI se doivent alors d'envisager immédiatement l'acquisition d'une nouvelle culture « projet » et de l'appliquer à l'évolution de leur profession. Dans cette optique, *Pilote 2010* leur propose une couverture complète et en l'état de l'art, des évolutions du métier. Cette ambition positionne l'ouvrage comme la Bible du développement d'applications pour la décennie en cours.