
FORUM LOGICIEL

Pour les professionnels des systèmes d'information

ISSN 1023-0114

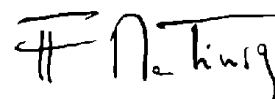
Août 1999 (9e année, numéro 4)

Vite, bien et moins cher...

C'est souvent le souhait formulé par les directions utilisatrices envers leurs services informatiques. L'évidence quotidienne, et pas seulement pour les projets informatiques, montre qu'il n'est pas évident d'associer ces trois qualités dans la même activité. La vitesse n'est ainsi pas l'alliée naturelle de la qualité. "Vite, bien et moins cher" est pourtant une vision simplifiée de l'offre groupée sous le terme RAD, même si le RAD 2 de Jean-Pierre Vickoff s'en distancie.

Si cette approche a le mérite de rappeler des principes majeurs, elle n'a pas inventé de nouveaux concepts de gestion de projet. L'importance de l'implication des utilisateurs, la fixation de priorités et le développement incrémental sont des notions qui existaient avant la création du RAD. Or, il se trouve que la plupart des projets ont toujours de la peine à respecter leur budget et leur délai, sans même parler de la satisfaction des utilisateurs. En dehors des concepts détaillés de "time boxing" ou d'itération, ces problèmes proviennent plus de l'environnement dans lequel se déroulent les projets. Prenons d'abord nos précautions en rappelant que les conditions idéales pour la réussite d'un projet ne sont pas faciles à réunir. Il n'est en effet pas évident de trouver sur tous les projets des utilisateurs et informaticiens engagés et compétents qui oeuvrent sur des objectifs réalistes et réalisables.

Au-delà de cette constatation, il reste souvent un grand effort à fournir dans les entreprises pour mettre en place une culture de réussite de projets. C'est plus particulièrement vrai dans le domaine de la collaboration entre utilisateurs et informaticiens. Le simple point de la gestion des changements, qui sont des phénomènes inévitables dans la vie d'un projet, est souvent traité de manière très inadéquate. Il est vrai que la réaction aux changements présuppose l'existence d'une structure organisationnelle et de moyens de pilotages qui font souvent défauts. Il est difficile de maîtriser ce que l'on ne peut contrôler, un état de fait qui est aussi bien valable pour les utilisateurs que pour les informaticiens. Inspirez-vous donc de l'article de Jean-Pierre Vickoff, mais vérifiez d'abord que les fondamentaux de la gestion de projet sont bien en place. Si le chef de projet est incapable de communiquer, de négocier ou d'animer une équipe, la meilleure méthode de gestion de projet ne servira qu'à gérer des problèmes.



Sommaire

RAD 2 : la conduite de projet "haute performance" page 2

Actualité page 15

RAD 2 : la conduite de projet "haute performance"

Jean-Pierre Vickoff
VICKOFF@compuserve.com
<http://www.rad.fr>

La recherche de qualité et de performance concerne désormais toutes les entreprises et les administrations. Vecteur privilégié de cette évolution, les systèmes d'information (SI) se doivent d'être sophistiqués pour répondre à des besoins de plus en plus complexes, et, dans le même temps, totalement évolutifs, afin de soutenir une adaptation permanente.

Parmi les réponses technologiques, l'architecture client-serveur dans son aboutissement « NET » (intranet, extranet, internet) s'avère le principal facteur structurant de la décennie. L'ouverture trans-domaines et même trans-organisations caractéristique de ce modèle, ainsi que les fortes contraintes de temps ou de coûts qui lui sont généralement associées, bouleversent la conduite de projets et consacrent la nécessité d'une totale synergie entre méthodes, techniques de communication, ressources humaines et outils.

Parmi les réponses méthodologiques, la méthode RAD (*Rapid Application Development*, [Martin 1991]) s'affirme comme une solution performante mais contraignante à mettre en œuvre. Malheureusement RAD est l'acronyme de *Développement Rapide d'Applications*. La notion de rapidité n'est pourtant qu'un aspect secondaire de la méthode. Il faut totalement dissocier le qualificatif de « RAD » appliqué aux outils « visuels » des principes et des objectifs de la méthode. La méthode RAD privilégie la recherche de la qualité applicative par la communication et la validation permanente de l'utilisateur. La qualité technique étant assurée par le respect de modes opératoires rigoureux. RAD est donc plutôt synonyme de *Réingénierie du Développement d'Applications* et représente à l'évidence le contraire de la facilité automatisée.

La méthode RAD est à la fois une évolution culturelle, un investissement dans la rigueur et une nécessité profonde d'industrialiser les pratiques de développement. Cet état de

choses et de contraintes explique l'ignorance ou le rejet de la méthode par ceux qui ne sont pas encore acculés à la performance et son acceptation inconditionnelle par ceux qui le sont.

1.1 Enjeux de la méthode

En synthèse, il est possible de considérer la réussite d'un projet informatique comme la somme des points suivants :

- une application totalement approuvée répondant aux besoins des utilisateurs;
- la livraison rapide des fonctionnalités stratégiques;
- des coûts de développement permettant un retour sur investissement concret;
- la sécurité d'une visibilité qualitative et quantitative de l'avancement des travaux.

Pour un professionnel, il n'y a pas de miracle ni de circonstance, seulement des techniques de réduction du risque : processus formel, phasage, dimension temporelle, validation permanente, jalons zéro-défaut, focus de visibilité. Ainsi se déclinent les attributs de la méthode RAD 2.

1.2 Réduction du risque

Il faut livrer rapidement un minimum de fonctions viables afin d'assurer un retour sur investissement accéléré¹ et d'éviter un « effet tunnel »².

¹ Stratégiquement, le RAD choisit d'optimiser 4 axes fondamentaux : coût (*target costing*), délais (*time boxing*), qualité technique (*code and project reviews*), qualité fonctionnelle (*prototyping and user's reviews*) ou la visibilité générale et le contrôle du projet (*Focus*).

² L'effet tunnel correspond à la période durant laquelle l'utilisateur ne voit pas l'avancement de l'application. Effet caractéristique des approches de développement classiques où le programmeur travaille à partir des dossiers de l'analyste.

L'enjeu premier du RAD est de supprimer l'effet tunnel, car plus la détection d'erreur est tardive, et plus elle est coûteuse. L'effet tunnel a des impacts simultanés sur la qualité de l'application, les délais, le budget et la visibilité. Son inverse, la visibilité, est aussi indispensable à l'utilisateur pour des raisons fonctionnelles qu'à la direction pour son management des risques. Les techniques RAD de validation permanente offrent à ce problème la meilleure des réponses.

Pour atteindre l'adéquation parfaite de l'application aux besoins de l'organisation, la participation active et soutenue de la Maîtrise d'Ouvrage est indispensable. La distinction entre les activités dépendantes de la Maîtrise d'Ouvrage et celles dépendantes de la Maîtrise d'Œuvre se détermine donc préalablement à toutes les étapes du projet.

1.3 Principes fondamentaux

Pour qu'un projet puisse se qualifier de « RAD », plusieurs conditions doivent être remplies et plusieurs techniques doivent être mises en œuvre. Parmi l'ensemble de ces techniques, certaines ont plus de poids que d'autres, et il faut impérativement les respecter :

- La présence d'un **animateur spécialisé** facilite les communications de groupe et doit obtenir un équilibre de participation entre les utilisateurs et les informaticiens. L'animateur doit rendre compte à la direction générale et être rémunéré par elle, car sa neutralité est primordiale.
- Un **plan de communication** et de formation des utilisateurs est planifié et approuvé sur la base d'un contrat fondamental au respect de la planification.
- Une organisation « plate » de **l'équipe projet (SWAT¹)**. Idéalement, les membres de cette équipe auront tous le même profil : concepteur-développeur, mais chacun disposera d'une spécialité complémentaire à celle des autres.

¹ SWAT : Skilled With Advanced Tools (équipe de projet RAD)

- Une **structuration du projet** en cinq phases distinctes produisant des livrables définis et le **respect d'une dimension temporelle** fixe « *The need for a 90 days life cycle* ».
- Un ensemble de **matériels et de logiciels performants**, isolés dans un environnement permanent, dédié en priorité au projet (salle RAD).
- Un **engagement des utilisateurs stable et permanent**. A la base du principe de validation permanente, il garantit visibilité et qualité.
- Des formes et les **techniques de modélisation et de documentation adaptées** tant à la méthode en elle-même qu'aux nouveaux besoins propres aux développements modernes et ce particulièrement en matière de communication interpersonnelle.

1.4 Techniques d'entretien

A toutes les phases du développement, le RAD introduit la spécialisation et la gestion des relations interpersonnelles. Le classique chef de projet fait place à un binôme composé d'un coordonnateur fonctionnel issu de la maîtrise d'ouvrage qui impose la dynamique applicative et d'un coordonnateur technique représentant la maîtrise d'œuvre qui intègre la dimension technologique.

Les rapports entre les partenaires du développement se complexifient et se contractualisent. Le RAD implique alors un troisième groupe d'acteurs neutres organisé autour d'un animateur dont la mission est d'obtenir un consensus et une expression formelle des besoins par la maîtrise des techniques d'entretien, de prévision et de gestion des conflits.

Des auxiliaires (rapporteurs) réalisent en direct la modélisation de l'application et sa validation lors de travaux de groupe organisés dans un espace de communication dédié (atelier de génie logiciel sur micros, moyens électroniques de rétroprojection, etc.). Un mode de travail collectif caractérise les entretiens RAD.

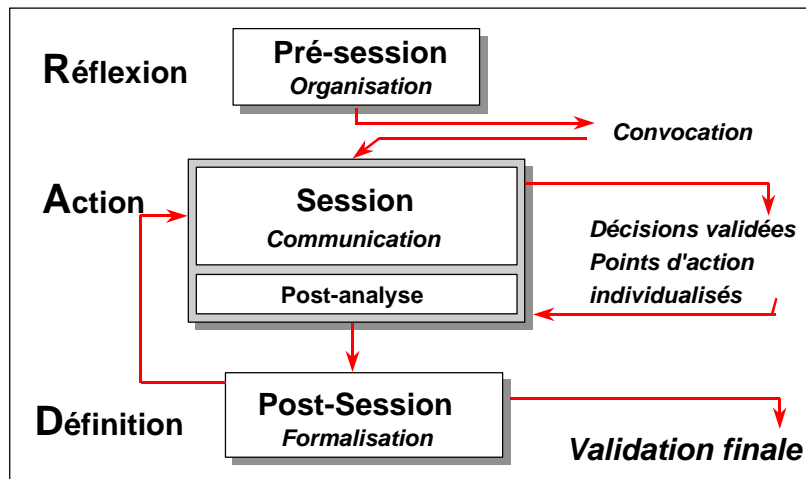


Figure 1. Communication structurée, mode opératoire

Il requiert une participation intensive des utilisateurs lors de l'expression des besoins. Le mode opératoire des communications est structuré en 3 étapes : pré-session, session, post-session (figure 1) :

La pré-session permet de délimiter les thèmes à traiter lors de la session et de lister les prérequis. Le minutage des thèmes est la première des techniques à imposer. Une session non minutée conduit au dépassement du temps prévu ou au report du traitement de certains thèmes. Au-delà du retard, cette situation déclenche une cascade d'inconvénients, particulièrement lorsqu'il existe des dépendances entre les thèmes traités et les thèmes non traités [Bartoli 1993]. Le produit de la pré-session est une convocation envoyée aux participants.

La session est une étape de résolution progressive des problèmes. Le principe est une recherche de consensus entre les participants suivie d'un verrouillage de la décision. L'animateur s'appuie sur le minutage prévu lors de la pré-session pour obtenir un avancement cohérent. Les thèmes programmés sont traités et clôturés successivement sans retour. Le traitement des thèmes s'effectue par *résolution directe*, par *consensus* ou par *arbitrage* de l'animateur.

Les décisions prises font l'objet d'une validation immédiate et d'un procès-verbal enregistré par le rapporteur. Le produit d'une session est un rapport récapitulatif des décisions validées et des points d'action

restant ouverts.

La post-session entérine la clôture de tous les thèmes et des points d'action s'y rapportant. Le produit de la post-session se concrétise par l'enrichissement du plan de développement de l'application.

1.5 Ressources humaines

L'ensemble des principes de cette **communication structurée** entre Maîtrise d'Ouvrage et Maîtrise d'Œuvre conduit au respect d'une frontière naturelle entre la prise de conscience des besoins et leur expression. La recherche de solutions et la validation fusionnent naturellement. Une profonde remise en question des habitudes de développement d'applications s'impose. Elle est fondée sur la dichotomie Maîtrise d'Ouvrage et Maîtrise d'Œuvre et entraîne l'intervention d'un troisième groupe spécialisé dans la communication. Il en découle une redistribution des rôles et un apprentissage :

- A. La **Maîtrise d'Ouvrage** représente l'utilisateur qui devient concepteur. Elle détermine les fonctions, leurs priorités et impose la « dynamique applicative »¹. Elle utilise des formes de modélisation simplifiées pour représenter la vision de son travail [Henry 1995] et ses scénarios opérationnels (*use case*) [Jacobson 1993].

¹ La **dynamique applicative** focalise une synergie d'évolutions (organisation, communication, technologie).

B. Le **groupe d'animation RAD** organise la communication du projet. Il facilite l'expression des besoins et réalise en « temps réel » leur formalisation. Il se compose d'intervenants spécialisés en communication (animateur ou facilitateur) et en entretiens de groupe [Sary 1990]. Il dispose de matériels et de logiciels adéquats dans une salle dédiée, isolée et communiquant uniquement vers l'extérieur. Il réalise « en direct » la synthèse (*rapporteur-secrétaire*) et la modélisation (*rapporteur-modélisateur*) à partir du discours utilisateur [Vickoff 1996].

C. La **Maîtrise d'Œuvre** représente une force de solution et de proposition technique. Sous la double pression des nouveaux types d'applications et des contraintes économiques, elle fusionne en un seul profil de concepteur-développeur les rôles de l'analyste et du programmeur [Bouchy 1994].

« La connaissance a deux composantes, la contemplation et l'action : très souvent, pour comprendre, il faut être capable de faire. La méthode, si on la conçoit autrement que comme une codification des moyens de preuve, est du côté de l'action ».

En ce qui concerne La Maîtrise d'Œuvre, 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** (*Skilled With Advanced Tools*). Un SWAT est composé de personnels de type spécialiste-généraliste, experts sur des techniques ou outils précis et nécessairement généralistes sur les autres. La coordination harmonieuse de ces profils à travers un contrat de projet initie une dynamique de groupe basée sur la gestion de la complémentarité. Cette synergie favorise le sens de l'identité de l'équipe, le partage de l'information et la notion d'entraide.

1.6 Méthode et cycle de vie

« Le mot méthode, d'origine grecque, signifie chemin : celui, tracé à l'avance, qui conduit à un résultat. La méthode se rapporte à la meilleure façon de conduire un raisonnement ».

Jusqu'à maintenant, les méthodes se distinguaient par leur approche monolithique des problèmes. On décomposait la structure (*top-down*) ou l'on élaborait à partir des besoins (*bottom-up*). Le RAD réconcilie les deux tendances.

La méthode RAD structure le cycle de vie du projet en 5 phases :

1. La **Préparation** (*initialisation*) définit l'organisation, le périmètre et le plan de communication.
2. Le **Cadrage** (*requirement*) définit un espace d'objectifs, de solutions et de moyens.
3. Le **Design** (*user design*) modélise la solution et valide sa cohérence systémique.
4. La **Construction** (*construction*) réalise l'application en prototypage actif (validation permanente).
5. La **Finalisation** (*cutover*) est un contrôle final de qualité en site pilote.

Le RAD s'appuie d'abord sur une approche systémique (*top-down*) dans la première partie de sa mise en œuvre : le Cadrage. Cette phase correspond à une analyse, étendue aux aspects organisationnels, technologiques et financiers.

Lors de la seconde partie d'un projet RAD, le Design, l'approche objet est mise à contribution avec pragmatisme afin de structurer et de pouvoir paralléliser la conception. L'apport de l'objet réside alors dans la parcellisation naturelle qu'il implique et dans l'isolation et l'homogénéité qu'il impose en termes de fonctionnalités. En ce sens, il ouvre la porte à la réutilisation et à l'industrialisation des composants « métier ».

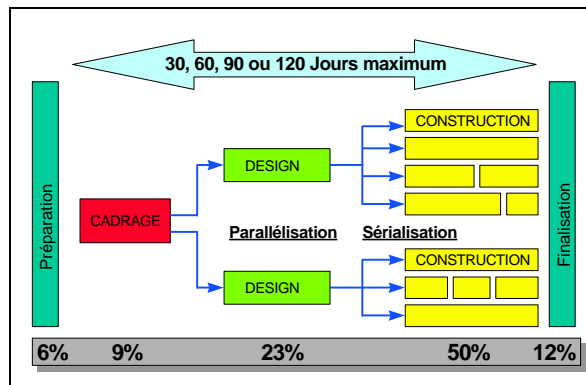


Figure 2. Parallélisation et sérialisation

La troisième partie, la Construction, s'appuie sur le prototypage (*bottom-up*). Le RAD plonge alors au cœur des besoins pour construire une application répondant aux critères de qualité et de conformité fonctionnelle. Et ce, jusque dans le moindre détail des attentes d'un utilisateur, participant actif depuis le début du projet. Dans cette phase, l'objet technique trouve de nouveau sa place en fournissant les briques qui accéléreront la Construction.

1.7 Organisation des grands projets

Les projets majeurs impliquent une métrique précise et le respect d'une dimension temporelle des cycles. Il en découle un découpage en sessions parallèles (*figure 2*). On optimise en planifiant des lots de 60, 90, 120 jours. Au-delà, apparaissent la baisse de productivité, la démotivation et le turnover : un projet long est un mauvais projet. A titre indicatif, il est utile de savoir, qu'en général, le poids respectif des phases est le suivant :

- Préparation (6%);
- Cadrage (9%);
- Design (23%);
- Construction (50%);
- Recette RAD (12%).

L'engagement utilisateur minimum est de l'ordre de 12% à 15% de la charge incombant à l'informatique pour les applications de gestion classique.

1.8 Facilités de développement

Parfois, le développement spécifique d'une

application n'est plus justifiable, particulièrement lorsque les délais représentent un critère primordial. La recherche d'une solution « progiciel » est alors envisagée voire préconisée sans autre forme de raisonnement. Le tout progiciel ne représente pourtant pas la seule alternative au tout spécifique.

Une solution alternative permet d'éviter le piège qui consiste à sacrifier la couverture fonctionnelle à la raison économique. Cette solution se construit par l'achat de composants (modèles, écrans, données, documentation, objets, etc.). Cette ingénierie représente souvent l'option optimale en termes de réduction des risques (fonctionnel, technique et économique). Actuellement un développement peut se réaliser économiquement à partir d'une combinaison de trois options : achat d'un progiciel, développement spécifique, achat de « solutions ».

La notion de "solutions" recouvre l'obtention de :

- divers modèles (données, traitements);
- de scripts de création de bases de données;
- de jeux d'essais et de tests de cheminements;
- de dessins d'écrans;
- de spécifications et d'algorithmes;
- de modules complets, de parties de code déjà programmé;
- de composants ou d'objets "métier".

Réaliser un développement dans ces conditions présente de nombreux avantages en termes de :

- Budget : réduction jusqu'à 90 % des coûts de conception et jusqu'à 70 % des coûts de réalisation.
- Délais : réduction drastique des délais de mise en exploitation.
- Qualité : suppression de la plupart des conditions d'erreurs et aménagement de la solution au plus près de nouveaux besoins.
- Visibilité : levée de la majorité des risques fonctionnels, techniques et de ceux liés aux contraintes de délais.

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 :

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

1.9 Evaluation et planification

La planification optimale d'un projet de développement implique un découpage par lots privilégiant les composants à forte valeur ajoutée. La précision de l'estimation et la possibilité de comparer instantanément des scénarios complexes acquièrent alors une importance vitale alors que le nombre de paramètres représentatifs de l'environnement à la fois technologiques, organisationnels, méthodologiques et

« motivationnels ¹ » s'accroît.

Il faut donc avoir recours à des outils spécialisés dans l'évaluation de la charge et de la performance de la ressource. Un de ces outils « Évaluateur » est téléchargeable gratuitement à partir du site WEB de l'auteur (<http://www.RAD.fr>).

Les principes de l'Évaluateur sont identiques à ceux des Points de Fonctions. Mais, si les P.F. évaluent grossièrement un projet, ils ne prennent pas en compte les nouveaux et nombreux paramètres affectant les conditions de sa réalisation et n'autorisent pas une finesse suffisante pour faciliter des négociations de délais, de budget et de réduction de fonctionnalités.

1.10 Sécurité des transitions de phases

La politique de sécurité lors des transitions de phases applique le principe du livrable minimum. Il ne faut pas sous-estimer la quantité de livrables devant être produits par le travail conjoint de la Maîtrise d'Ouvrage, de la Maîtrise d'Œuvre et du groupe d'animation.

Dans cet esprit « nécessaire et suffisant » du livrable intermédiaire, le *tableau 1* offre une synthèse des documents facilitant la clôture sécurisée d'une étape ou d'une phase.

Pour plus de détails, se référer au processus formel RAD2 publié dans le cadre du *rapport Ré-ingénierie du Développement d'Applications* par la division Produits et Services du Gartner Group (01 41 35 15 15). Ce processus sécurise et fiabilise la conduite de projet RAD en mettant en œuvre :

- A. Une **structure de développement** sécurisant un cycle court basé sur un phasage simple : Cadrage, Design, Construction et l'absolu respect d'une dimension temporelle (90 jours optimum, 120 jours maximum) [Martin 1991].

¹ Ensemble des éléments qui concourent à la satisfaction profonde des intervenants

Management

Phases	Etapes ou travaux	Documents produits avant clôture
Préparation	Entretien initial	Engagement réciproque des Maîtrises
	Immersion animateur	Périmètre applicatif. Plan de communication
	Réunion de lancement	Travaux individualisés. Planning accepté
Cadrage	Sessions définition problème Focus de fin de Cadrage	Modèle global des flux (DFD) Modèle hiérarchique des traitements
Design	Sessions conception solution Focus de fin de Design	Modèle détaillé des données et (si utile) Modèle détaillé des flux et traitements Prototype initial
Construction	Revue de code et de projet Revue fonctionnelles État de livraison permanente Focus de présentation	Application opérationnelle validée : - fonctionnellement par les utilisateurs - techniquement par Jalons Zéro-Défaut
Finalisation	Cheminelements fonctionnels	Homologation et recette

Tableau 1. Documents conseillés pour assurer la clôture d'une phase

- B. Des **méthodes, techniques et outils** permettant de définir et d'appliquer des choix portant sur quatre stratégies conflictuelles : budget, délais, fiabilité (qualité technique), visibilité (qualité fonctionnelle) [Vickoff 1998].
- C. Une **architecture de communication** respectant un mode opératoire précis structuré en trois étapes : pré-session, session, post-session [Mucchielli 1987, Vickoff 1996].
- D. Une **architecture de conception** s'appuyant sur les techniques de l'objet et particulièrement sur celles permettant une conception « en vue de modifications » [McCarty 1997].
- E. Une **architecture de réalisation** imposant pour la qualité technique des normes minimales, des revues de projet, des jalons zéro-défaut et recommandant pour la qualité fonctionnelle le prototypage actif et les Focus de visibilité (*figure 3*) [McConnell 1996, Vickoff 1998].

Ces principes se déclinent dans le cadre des 5 phases représentatives de la méthode RAD :

1. **PREPARATION** (de l'organisation) et communication RAD. Cette phase permet de définir le périmètre général du projet, de structurer le travail par thèmes, de sélectionner les acteurs pertinents et

d'amorcer une dynamique de projet. Cette phase représente environ 6% du projet.

2. **CADRAGE** (techniques RAD d'analyse et d'expression des besoins). L'expression des besoins est effectuée par les utilisateurs qui présentent leur travail lors d'entretiens de groupe (session). Il est généralement prévu entre 2 et 5 jours de sessions par commission (thème). Cette phase représente environ 9% du projet.

3. **DESIGN**¹ (techniques RAD de conception et de modélisation). Les utilisateurs sont également impliqués dans cette étape. Ils participent à l'affinage et à la validation des modèles organisationnels : flux, traitements, données. Ils valident également le premier niveau de prototype présentant l'ergonomie et la cinématique générale de l'application. Il est prévu entre 4 et 8 jours de sessions par commission. Cette phase représente environ 23% du projet.

4. **CONSTRUCTION** (techniques RAD de réalisation, prototypage). Durant cette phase, l'équipe RAD (SWAT), doit construire l'application module par module. L'utilisateur participe toujours activement aux spécifications détaillées et à la validation des prototypes. Plusieurs sessions itératives sont nécessaires. Cette phase représente environ 50% du projet.

¹ **DESIGN** : deuxième phase fondamentale du RAD. Elle recouvre des parties de conception et de modélisation classiques ainsi que le premier niveau de prototypage « actif ».

5. FINALISATION. Des recettes partielles ayant été faites à l'étape précédente, il s'agit dans cette phase d'officialiser une livraison globale et de transférer le système en exploitation et maintenance. Cette phase représente environ 12% du projet.

1.11 RAD en pratique

Voici pratiquement et succinctement comment se déroule un projet RAD. La première des conditions réside dans la présence d'un animateur ou d'un coordonnateur maîtrisant parfaitement tous les aspects du RAD. Dans les projets où des dissensions pourraient apparaître entre les différents interlocuteurs ce spécialiste de l'entretien de groupe doit nécessairement être neutre. Fonctionnellement, deux catégories d'intervenants participent à un projet RAD : ceux pour action qui sont convoqués et ceux pour information qui sont invités.

La Maîtrise d'Ouvrage doit s'investir aussi régulièrement dans la validation de sa future application que la Maîtrise d'Œuvre le fait dans sa production. Il est illusoire de vouloir développer avec des ressources à temps partiel un projet stratégique ou sous la contrainte de temps. Seul le respect de ces conditions de communication permettra aux développements modernes de sortir de leur enlisement actuel. Lorsque la charge de travail ainsi dédiée à la spécification et au prototypage est trop importante pour la Maîtrise d'Ouvrage et déséquilibre sa production opérationnelle, des ressources supplémentaires lui sont affectées. Leur coût est inclus dans le budget global du projet. Le retour sur investissement de ce recouvrement de fonction est chiffrable et positif.

Une des missions de l'animateur est de justifier l'impact de chaque action en termes de retour sur investissement. L'animateur, en plus de son rôle de facilitateur, est le garant du respect de la méthode. Il informe les maîtrises des écarts observés et de leur conséquence sur la stratégie initialement décidée. Il forme les intervenants, contrôle la planification, l'ordonnancement des tâches et le suivi de leur exécution. Il s'assure de l'efficacité des entretiens

(participation, progression, centrage des thèmes, respect des priorités) et de la performance de l'environnement technologique, méthodologique.

Afin de soutenir l'évolution positive des motivations et la dynamique d'équipe, des moyens matériels sont à sa disposition : budget pour actions incitatives au renforcement de la cohésion interpersonnelle, amélioration du cadre de travail, assouplissement des contraintes matérielles, prise de décision démocratique, prime sur atteinte d'objectifs. Il est « neutre » vis-à-vis des deux maîtrises et cette neutralité doit le faire dépendre de la direction générale [Boehm].

1.12 Phase INITIALISATION

La phase d'INITIALISATION informe les intervenants des contraintes d'un projet RAD. Après une courte IMMERSION dans le domaine fonctionnel, les chefs de projets et l'animateur présentent aux dirigeants et à la maîtrise d'ouvrage les contraintes de la méthode et le plan d'action à respecter. Cette étape à pour nom l'ENTRETIEN PROPRIETAIRE.

Un modèle de communication est produit. Si nécessaire, une opération de réingénierie des processus précède l'automatisation du domaine. Une RÉUNION DE LANCEMENT du projet est ensuite organisée. Elle regroupe tous les acteurs recensés et donne lieu à l'individualisation des travaux préparatoires à la modélisation du système. 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 le principe des groupes de travail de celui des 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.

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 doivent être en grande partie les réels exécutants des tâches à automatiser.

Le choix des utilisateurs participant au projet est fonction de leur motivation affirmée. Ce point est essentiel au RAD, les futurs clients du système représentent une force de proposition fonctionnelle. L'entretien de groupe est la meilleure forme de réunion pour déterminer les fonctionnalités générales d'un système. Lorsque celui-ci est conséquent, un individu isolé a difficilement une vue d'ensemble objective et complète.

Parfois, pour réaliser des fonctions identiques, les méthodes de travail sont différentes d'un participant à l'autre. L'analyse commune des diverses pratiques aboutit à un consensus sur la plus efficace. L'interview de groupe, par la communication qu'il suscite, éveille naturellement la réflexion, l'évolution, le progrès dans la recherche d'amélioration de la qualité. La réunion de lancement (d'une demi-journée à une journée) est souvent considérée comme le lancement officiel du projet RAD.

A l'issue de la réunion de lancement, les participants disposent de quelques jours pour réaliser la préparation de la première réunion de Cadrage dont la date est fixée.

Phase CADRAGE

Il est inutile d'engager des ressources humaines de qualité et coûteuses si les ressources matérielles (salle, rétro, micros, logiciels) ne sont pas disponibles. Le coût direct et la démotivation résultant de ce genre de situation sont tels que le qualificatif « RAD » peut être oublié.

Lors du CADRAGE auquel participent les décideurs, l'animateur obtient le verrouillage des besoins, des budgets, des délais et de la solution globale sur les plans stratégique, fonctionnel, technologique et organisationnel.

Dans le cas où les besoins et les ressources divergeraient, il faut attribuer des priorités aux fonctionnalités en termes de retour sur investissement. Cette modélisation des traitements s'effectue sous la forme d'une hiérarchie de fonctions.

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.

Modéliser en direct avec un rétroprojecteur électronique. Dans le principe, la modélisation concrète des flux et des fonctionnalités devra être engagée le plus tôt possible avec un outil adéquat.

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 chargés de la conception et du 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 d'utilisateurs qui participent aux séances de définition de la future application sont limités à quelques personnes significatives. 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. Cette approche s'avère généralement efficace et est indispensable dans le cadre d'un management par la qualité de service (MTQS). Avant la réunion officielle, on engagera donc une réflexion sur l'ensemble des thèmes composant le domaine applicatif.

1.14 Phase DESIGN

La phase suivante, le DESIGN, repose sur la disponibilité d'un AGL de conception léger et puissant. Cet outil est utilisé « en direct », dans une salle spécialement équipée de moyens de rétroprojection et de communication. Sous la coordination de l'animateur, les utilisateurs significatifs et les concepteurs-développeurs travaillent alors en commun et en direct à la modélisation détaillée des traitements et des données de l'application. La présentation d'un premier niveau de prototype conclut cette phase.

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 ».

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. Par ailleurs, un système basé sur une structuration « métier » des données et sur une stratification des traitements dispose d'une grande capacité d'évolution d'une pérennité exceptionnelle. Cette approche s'avère plus facile à développer malgré sa généralité :

- A. Les parties les moins sujettes à évolution sont réalisées en premier;
- B. 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. Abstraction, 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.

1.15 Phase CONSTRUCTION

La phase de CONSTRUCTION affine le prototype. Elle fusionne les étapes classiques de spécification détaillée, de réalisation (codage), de tests unitaires et de tests d'intégration, ainsi que la plus grande partie des tests de cheminement fonctionnel qui constituent la recette initiale de l'application. Cette homogénéité constitue, avec la prise de décision immédiate et la validation permanente, les bases mêmes de la productivité et de la qualité RAD.

Pour atteindre cette performance, les outils de Construction utilisés doivent être choisis avec soin, une charte graphique doit avoir été validée, des normes de programmation employées, un modèle de transaction généralisé à tous les modules.

La validation permanente, comme son nom l'indique, est réellement permanente. Elle s'effectue à chaque séance de travail avec l'utilisateur.

La validation permanente 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.

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

- **SWAT** : Organisation d'une équipe de profil « concepteur-développeur » basée 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)** : Technique orientée « planning »; elle permet de contrôler l'avancement du projet en matière de visibilité et de qualité. Des jalons ZD peuvent être, si nécessaire, planifiés entre les focus. Ils correspondent à une revue du code suivie d'une intégration et d'une validation complète de cheminement par l'utilisateur de base.
- **Focus** : Réunion plénière au niveau du projet, elle correspond obligatoirement au positionnement d'un des jalons ZD.
- **État 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.

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 techniques 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.

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 base sur la notion de priorité des fonctions et impose souvent que l'ensemble des modules soient mis en chantier

simultanément, mais à un niveau de fonctionnalité limité.

La notion de Focus couvre en pratique l'enchaînement de plusieurs étapes contraignantes pour le SWAT. 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 délai précis (de 10 à 20 jours). Participent au Focus tous les acteurs du projet. 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 RAD.

Quelques jours avant la date fixée pour le Focus, en fonction de l'avancement validé, 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 d'anomalies 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. L'équipe procède alors à des revues croisées. Le niveau de qualité requis peut être très variable d'une application à une autre selon sa typologie. Cette étape prend fin lorsque la qualité de l'application atteint un niveau suffisant en robustesse, normalisation, documentation et 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 des applicatifs mettant en œuvre des données communes (externes, globales).

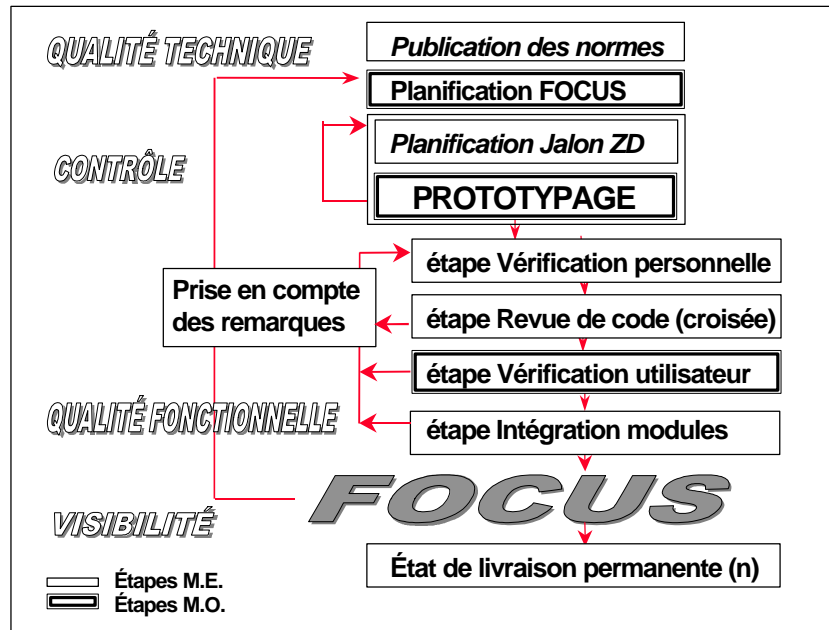


Figure 3. Architecture de réalisation

Les utilisateurs ayant participé aux 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 et ils sont susceptibles de déclencher des événements que le développeur n'aurait pas imaginés. A ce point, seules les anomalies bloquantes et majeures sont corrigées. Cette opération implique un minimum de tests de non-régression et une étape complète d'intégration.

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 (même réduit). Le Focus est une affaire de visibilité externe

ponctuelle et non un moyen de validation interne.

Entre deux Focus, les concepteurs-développeurs sont en contact avec les utilisateurs qui participent en direct à la spécification de détail. Lors d'un Focus, l'utilisateur manipule la partie de l'application à laquelle il a contribué. Les spectateurs observent et critiquent. Le groupe d'animation note les observations. Sur le plan des communications, l'organisation d'un Focus est identique à celle d'une session de travail. Le nombre de Focus dépend de la durée du projet et de sa complexité¹. Il est en général planifié 3 ou 4 Focus pour un « petit » projet et de 4 à 8 Focus pour un projet intermédiaire.

Lors du focus, la discrétion de l'informaticien est proportionnelle à son efficacité technique et à la puissance de sa méthode.

¹ 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 est alors dépendante du nombre d'équipes et du style de projet.

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

Tableau 2. Nombre de Focus

1.16 Phase FINALISATION

Pour de multiples raisons dont notamment les techniques de validation permanente et de Jalons Zéro-Défaut, la recette RAD est beaucoup moins conséquente qu'une recette classique. La phase de FINALISATION comprend, entre autres, la formation, les activités liées au déploiement, le bilan de projet, le retour de connaissance. Elle clôture le projet RAD.

1.17 Conclusions

Le processus RAD2 représente l'aboutissement formalisé d'une maîtrise organisationnelle et méthodologique de la conduite de projet. Cette avancée significative de l'Etat de l'Art se voit consacrée en mai 1999 par la reconnaissance du Gartner Group dans le rapport « Ré-ingénierie du Développement d'Applications ».

Au-delà des principes théoriques, lors du démarrage de chaque projet, il faut conserver à l'esprit les réalités suivantes : la méthode RAD ne remplace pas la modélisation du système par du prototypage, mais utilise cette technique pour associer pleinement les utilisateurs dans une recherche de consensus et d'efficacité.

La méthode RAD s'applique à toutes les tailles de projets et n'est pas uniquement dédiée à des petits projets isolés. Au contraire, il faut une organisation raisonnablement conséquente pour bénéficier pleinement des avantages offerts par cette approche rigoureuse et formelle des développements. Plus le projet est complexe, important, sensible ou nécessite une bonne communication et plus le RAD démontre la pertinence et l'efficacité de son cycle de vie face aux méthodes totalement en cascade ou totalement itératives.

La méthode RAD a pour objectif principal d'améliorer la qualité des applications par l'organisation d'une participation accrue des utilisateurs dans la réalisation de leur futur outil de travail. La méthode RAD privilégie donc la recherche de fonctionnalités utiles et non pas la production de code ou d'écrans. La mise en œuvre systématique des techniques RAD permet de réduire les coûts et les temps de développement de l'ordre de 30% à 50% par rapport à un projet identique où toutes les conditions de la performance n'auraient pas été réunies. Rencontrer l'ensemble des prérequis de la méthode, puis respecter un processus formel et l'esprit d'un plan d'assurance qualité représente une pratique qui conduit systématiquement au succès des projets.

BIBLIOGRAPHIE

- [Boehm (B.), ...], *A Collaborative Spiral Software Process Model*, USC, 1994
- [Bouchy (S.)], *Ingénierie des systèmes d'information évolutifs*, Eyrolles, 1994
- [Henry (A.), ...], *Rédiger les procédures de l'entreprise*, Edition d'Organisation, 1995
- [Jacobson (I.)], *Le génie logiciel orienté objet*, Addison-Wesley, 1991
- [Martin (James)], *Rapid Application Development*, Macmillan, 1991
- [Mc Carty (J.)], *54 Règles d'or pour un grand logiciel*, Microsoft Press, 1997
- [Mc Connell (S.)], *Stratégie de développement rapide*, Microsoft Press, 1996
- [Mucchielli (R.)], *L'Interview de groupe*, ESF, 1987.
- [Muller (P. A.)], *Modélisation Objet avec UML*, Eyrolles, 1997.
- [Nanci (D.), ...] *Ingénierie des systèmes avec Merise*, Sybex, 1993.
- [Paulk], *The Capability Model : Guidelines*, SEI, 1995
- [Rumbaugh (J.)], *OMT Modélisation et conception OO*, Prentice Hall (Masson), 1996
- [Sary (P.)], *Stratégie de la programmation neurolinguistique*, Edition Retz, 1990
- [Vickoff (J-P)], *RAD - Développement Rapide d'Applications*, Macmillan, 1996
- [Vickoff (J-P)], *Reengineering, Le Paradigme du futur immédiat*, Qualité Ingénierie, 1998
- [Vickoff (J-P)], *Ré-ingénierie du Développement d'Applications, RAD2*, Gartner Group, 1999
- [Vickoff (J-P)], *Pratique d'Assurance Qualité RAD2*, www.RAD.fr, 1999



Sociétés

Alliance entre Rational et IBM

Rational et IBM ont annoncé la conclusion d'une alliance stratégique. Son premier résultat est la nouvelle intégration entre la suite VisualAge d'IBM et l'environnement de conception Rose de Rational. Selon IBM, cela permettra la génération complète de code à partir des modèles contenus dans Rose, et vice-versa.

Si la réalité des promesses de cette annonce reste à vérifier, Rational confirme néanmoins sa position d'acteur majeur dans le monde du génie logiciel au vu de ses liens privilégiés avec IBM et Microsoft. C'est aussi l'occasion de se conforter dans l'idée que les environnements de conceptions semblent encore avoir un rôle à jouer, même à l'époque d'Internet.

Citations & Commentaires

Formation ? FormationS !

"Selon un rapport du gouvernement des Etats-Unis, seulement un quart des personnes qui travaillent comme développeurs de logiciels aux Etats-Unis dispose d'un diplôme en informatique. Comparons avec les docteurs, avocats ou ingénieurs : presque chaque personne active dans ces domaines a un diplôme lié à sa profession. [...] Chaque communauté est composée de membres jeunes et âgés; certains disposent de plus de capacités que d'autres. [...] Il faut répondre aux besoins de développeurs qui ont une solide éducation orientée sur l'informatique, le génie logiciel et les sujets liés, mais aussi à ceux des programmeurs formés "sur le tas", ces ingénieurs, comptables, enseignants ou

autres qui se retrouvent aujourd'hui à gagner leur vie en développant du logiciel, même s'ils n'avaient pas planifié de devenir des développeurs."

Source : "From the Editor", Steve McConnell, IEEE Software, September-October 1998.

Si les chiffres varient, la même situation se retrouve dans nos pays. Une bonne partie des développeurs n'a pas de diplôme en informatique. Pour ces personnes, mais également pour une partie des informaticiens diplômés, il existe une équivalence simplificatrice entre le développement de logiciel et la programmation. Programmer est considéré un art "intuitif" basé sur la logique et le seul apprentissage nécessaire est celui du langage ou de l'environnement de programmation. Cette vision simplifiée existe aussi dans la culture de nombreux services informatiques où la programmation est la seule vraie "activité productrice".

Et pourtant, la conception, la documentation, la structuration, les tests ou la gestion de la maintenance d'un programme sont des éléments qui contribuent plus que la programmation, souvent assez simple dans les environnements de gestion d'entreprise, au succès et à la qualité des projets de développement informatique. Mais combien d'entreprises enverront les nouveaux développeurs suivre des cours de spécification et de choix d'algorithmes ? Dans ces conditions, faut-il encore s'étonner des difficultés rencontrées par la majorité des projets ?

Mesurer avec mesure

- "Ne jamais mesurer que pour mesurer. Si vous ne savez pas quoi faire avec les résultats, pourquoi mesurer ?

- Ne cherchez pas la perfection. Trop d'organisations essaient de trouver la mesure parfaite ou de concentrer leur programme sur une seule mesure. L'expérience montre que différentes mesures sont meilleures pour différentes phases ou niveaux de management.
- Commencez en utilisant peu de mesures. Il est facile d'en adopter trop au lieu de se concentrer sur la douzaine de métriques qui sont significatives pour les gens.
- Impliquez tous les participants. Un programme réussi est compris par tous les participants. Il faut le soutien des programmeurs et des chefs de projet.
- Evitez le sabotage. Il existe un sabotage intentionnel ou non des programmes de mesure par de mauvaises actions, une résistance passive ou un manque de coopération.
- Intégrez les mesures à l'activité. Les métriques sont un mécanisme de feedback qui doit être reconnu par votre organisation. Les métriques doivent influencer les évaluations des projets."

Source : "Using Metrics to Justify Investment in IT", E. Chikofsky et H. Rubin, UT Professional, March/April 1999.

Selon les auteurs, seul un programme de mesures sur six résiste après deux ans...

Conférences

QWE'99 - 3rd International Software Quality Week Europe

8-12 novembre 1999, Bruxelles, Belgique

Software Research Institute, 901 Minnesota Street, San Francisco, CA 94107

Tél. +1 (415) 957-1441

Fax +1 (415) 957-0730

e-mail : qw@soft.com

<http://www.soft.com>

SPI 99- Software Process Improvement **30 novembre – 3 décembre 1999,**

Barcelone, Espagne

SPI 99, Meetings Management, Hart House, 4 The Hart, Farnham, Surrey, GU9 7HA, UK

Tél. +44 (0) 1252 726066

Fax +44 (0) 1252 723303

e-mail : spi@meetingmgmt.u-net.com

<http://www.meetingsmanagement.com>

Bientôt dans Forum Logiciel

- Introduction des outils de tests automatiques dans un projet.
- Base de données objet-relationnelle ou objet pure : comment choisir ?

FORUM LOGICIEL (ISSN 1023-0114) est une publication de **Martinig & Associés**, Avenue Nestlé 28, CH-1800 Vevey, Suisse Tél. +41 21 922 13 00 Fax +41 21 921 23 53 www.martinig.ch
Editeur : Franco Martinig E-Mail : franco@martinig.ch

Abonnement annuel : Europe 190,- Euros; Canada 290,- C\$ (par avion); Suisse et autres pays 290,- FS
Six numéros par année. Pour la Suisse : TVA 2,3% incluse.
Tirage de ce numéro : 500 exemplaires

La reproduction des textes n'est autorisée qu'avec l'accord préalable de l'éditeur et la mention de la source.
Copyright © 1999, Martinig & Associés