

ÉVALUATEUR



Evaluation de charge
Négociation de fonctionnalités
Optimisation d'environnement

Jean-Pierre Vickoff, RAD.fr

1. Évaluation, les multiples paramètres

Le produit *Évaluateur* considère l'ensemble de la problématique d'un projet sous ses aspects méthodologiques, technologiques, organisationnels et humains, pour :

- **estimer la charge** de développement ;
- **négocier la réalisation** des fonctionnalités ;
- **évaluer la performance** de l'environnement afin de **l'optimiser**.

A partir d'un **recensement** basé sur les éléments principaux du système (tables, écrans, rapports, interfaces) déterminés lors de la phase d'INITIALISATION puis affiné lors des phases suivantes, le produit *Évaluateur*¹ :

- détermine un nombre d'éléments techniques à produire (**Métrieque**) ;
- prend en compte des facteurs de pondération liés à l'environnement (**Effort**) ;
- estime l'ampleur en jours (**Charge**) et la durée du travail à réaliser (**Délai**) en fonction des ressources.

Principes basiques de l'évaluation

Évaluateur couvre plus de soixante paramètres qui disposent d'une aide contextuelle immédiate et sont répartis en 12 groupes (onglets). Dès le début de la saisie des composants de l'application, les informations de métrieque, d'effort et de charge apparaissent dans la « barre de résultat immédiat » (figure 1).



Figure 1. — *Évaluateur*, barre de résultat immédiat

La notion de **Métrieque**² est proportionnelle au nombre d'éléments et à leur complexité.

La notion d'**Effort**³ prend en considération les paramètres représentatifs de la performance de l'environnement.

Le nombre de **Jours** représente une charge de travail basée sur l'effort et intégrant la variable « compétence des ressources⁴ ».

Un indicateur « **Effet de seuil** » est situé entre l'effort et le nombre de jours. Le changement de couleur (de la plus claire à la plus foncée) de cet indicateur informe d'un accroissement sensible des ressources ou des délais (donc des coûts et des risques).

L'effet de seuil est lié à la charge, par rapport à la productivité optimale, d'une équipe dans un délai fixe.

¹ *Évaluateur* est un logiciel d'évaluation et de planification de scénarios de projets. Freeware, il est disponible sur le « NET » (www.RAD.fr). Il est téléchargé à ce jour en dizaines de milliers d'exemplaires. Il passe « on the fly » du français à l'anglais.

² Principes des Points de Fonctions et des Points d'Objets.

³ L'effort diminue si, par exemple, les outils sont performants.

⁴ La compétence des ressources réduit le temps de réalisation.

Onglet 1 – La métrique « applicative »

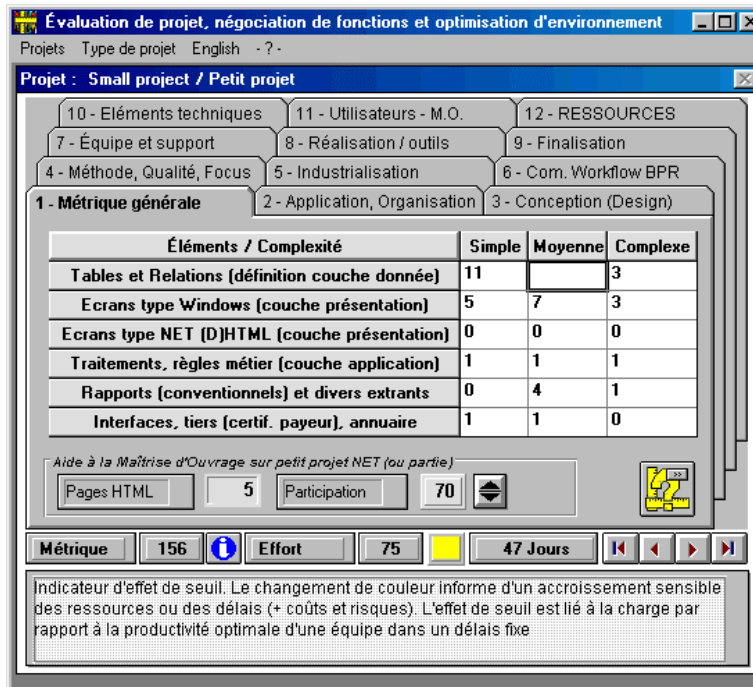


Figure 2. — Onglet 1 : métrique de l'application

La première et la plus conséquente partie du travail réside dans le recensement et l'estimation (nombre, niveaux de complexité) des éléments qui composent l'application. Pour remplir l'onglet Métrique, il faut en effet avoir une vision minimale de la taille de l'application et de sa complexité. Voici les étapes de cette recherche :

- Commencez par chercher le nombre d'éléments distincts qui composent le discours utilisateur. Vous découvrirez des « objets » (des données et des traitements associés). Cette étape permettra une première estimation des **Tables** et des relations. En observant les documents supports de ces informations vous obtiendrez une appréciation relative de leur importance (**simple, moyenne, complexe**). Ne négligez pas les relations dont les cardinalités sont de type N, elles génèrent de la complexité et doivent être considérées comme des tables simples.
- Toujours à partir du discours utilisateur, déterminez les règles de gestion ou d'organisation, ainsi que les traitements à effectuer (si possible, sous la forme d'une hiérarchie de fonctions). Comptez alors un **Écran** (Windows ou HTML) par élément ainsi recensé et un **Traitement** (*Business Process*) pour chaque élément dont la complexité applicative (fonctionnelle ou technique) est évidente.
- Le nombre de **Rapports** et d'**Interfaces** est ensuite généralement plus facile à circonvier, généralement en étudiant le modèle de flux.
- Pour les applications décisionnelles ou la maintenance d'applications, ne comptez pas les composants existants mais seulement ceux qui seront à ajouter ou à modifier. Dans ce cas, évaluez seulement la complexité de la modification.

Au-delà de l'oubli d'un élément connu, le risque principal se situe dans l'éventuelle explosion des exigences en cours de Cadrage ou de Design. Le rôle d'un animateur RAD expérimenté est d'obtenir une première estimation fiable et stable lors de l'immersion dans le domaine fonctionnel ; puis à la fin de chaque phase, d'affiner la planification.

1.1 Tables et Relations (couche donnée)

- Une entité (table) simple (définie ou modifiée) comprend au maximum une douzaine d'attributs. Ajoutez ici les tables issues des cardinalités de type N.
- Une entité (table) de complexité moyenne comprend plus d'une dizaine d'attributs, des clés étrangères et la gestion de l'intégrité référentielle.
- Une entité (table) complexe comprend plus d'une dizaine d'attributs, de nombreuses clés étrangères et implique des triggers et des procédures cataloguées.

1.2 Ecrans type Windows (couche présentation)

- Un écran Windows simple comprend :
 - de 5 à 6 objets (en plus des contrôles standards),
 - des procédures simples (la cinématique d'écran standard, les actions de recherche, de lecture, de création, de mise à jour) et les accès aux données sont inclus dans la notion d'écran.

Pour prendre en compte une complexité ergonomique exceptionnelle avec peu d'objets, utilisez le type d'écran « moyen ou complexe ».

- Un écran Windows de complexité moyenne comprend moins d'une douzaine d'objets + des contrôles standards + des procédures classiques.

Toute forme de réelle complexité applicative doit être recensée dans la couche « Traitements / Règles de métier ».

- Un écran Windows complexe comprend plus de 12 objets, des « customs controls » et des procédures complexes.

1.3 Ecrans type NET (D)HTML (couche présentation)

- Un écran (D)HTML simple comprend du code HTML sans accès aux données. Le contenu textuel et ergonomique est figé.
- Un écran (D)HTML moyen comprend une visualisation dynamique de données ou des objets simples ou un contenu instable.
- Un écran (D)HTML complexe comprend du code HTML qui encapsule divers types d'objets et des accès à des données de diverses sources.

1.4 Traitements / règles de métier (couche application)

- Un processus ou traitement simple comprend la programmation d'une règle de gestion ou des calculs semblant simples (ce qui correspond à moins d'une demi-journée de codage-test et se matérialise en général par moins d'une page de code, soit environ 50 lignes en incluant les commentaires).
- Un processus ou traitement moyen comprend la programmation d'une règle de gestion ou des calculs de complexité moyenne (ce qui correspond à moins d'une journée de codage-test et se matérialise en général par 2 ou 3 pages de code).
- Un processus ou traitement complexe comprend la programmation d'une règle de gestion ou des calculs paraissant complexes (ce qui correspond à plus d'une journée de codage-test et se matérialise en général par plus de 3 pages de codes). Décomposez en plusieurs éléments si nécessaire.

1.5 Rapports conventionnels et divers extraits

- Un rapport simple est du style « liste » avec des restrictions simples, un tri et un ou deux champs cumulés.
- Un rapport de complexité moyenne comprend plusieurs niveaux de rupture et nécessite des jointures entre les diverses tables impliquées.
- Un rapport complexe comprend de multiples ruptures, jointures, restrictions et calculs.

1.6 Interfaces (autres applicatifs, annuaires, tiers payeurs, etc.)

- Un interface (interne ou externe) de complexité simple est du style rapatriement de table de références (accès simple). Traitez comme des interfaces les tâches de reprise de données et les conversions.
- Un interface de complexité moyenne comprend plusieurs champs (accès de complexité moyenne). Traitez ici la gestion d'annuaire et accès tiers (certIFICATEURS, payeurs, etc.) standards ou de complexité moyenne.
- Un interface (interne ou externe) complexe comprend de nombreux champs et la nécessité de procéder à des vérifications d'intégrité référentielle. Traitez ici les gestions complexes d'annuaires et les accès difficiles à des tiers (certIFICATEURS, payeurs, etc.).

1.7 Aide à la maîtrise d'ouvrage sur petit projet NET (ou partie de projet)

- Indiquez le nombre de pages HTML structurellement différentes (99 maxi) à l'élaboration desquelles vous participerez (en ce qui concerne le contenu : texte, graphique, ergonomie). Utilisez cette option pour estimer un petit projet NET de type simple présentation de site WEB. Indiquez les pages STRUCTURELLEMENT différentes et manuellement programmées.

Onglet 2 – Application, Organisation

Figure 3. — Onglet 2 : qualité et organisation

2.1 Niveau applicatif (règles de gestion, complexité)

La difficulté applicative correspond soit au nombre et à la complexité des règles de gestion, soit à la sophistication de la demande. Il faut, pour ce paramètre, prendre en compte une complexité globale et non la présence de règles de gestion particulièrement complexes, mais isolées qui seront définies dans l'onglet -1- Métrique à la ligne « traitements, règles de métier, couche applicative ».

Cette évaluation de la complexité est subjective, elle correspond en fait à notre propre vision d'un problème. Pour certains un problème précis est complexe alors qu'il ne le sera pas pour une personne disposant de connaissances ou d'expériences propres à lui en faciliter la résolution. Pour cette raison, il est souhaitable que l'informaticien chargé de l'évaluation envisage la complexité applicative au regard des connaissances raisonnablement prévisible de la globalité de l'équipe chargée du développement. Le RAD

préconise dans le cadre d'un SWAT que l'évaluation soit effectuée par la personne ressource en planification ou par une approche collégiale de l'engagement de réalisation.

2.2 Standardisation exigée (normes)

Pour certains types d'applications (isolées et à durée de vie courte par exemple), exigez un haut niveau de normalisation est un choix coûteux et sans intérêt économique. La standardisation, le respect de normes a un coût non négligeable qui doit être modulé en fonction du ROI et de la durée de vie de l'application.

Un développeur habitué à respecter systématiquement des normes et des standards de développement pourra les mettre en œuvre sans perte de temps quelle que soit la taille de l'application. Par contre, pour un développeur ne disposant pas naturellement de ces réflexes, il faut prévoir de nombreux contrôles et travaux de corrections.

2.3 Recherche de réutilisabilité (objets)

Il faut choisir entre développer vite en respectant des contraintes stratégiques de délais ou de coûts et investir dans des objets réutilisables.

Le développement d'objets ou de composants avec pour finalité une forme de réutilisation représente un choix à la fois technique et économique. La surcouche de modélisation qu'il faut définir, ainsi que le haut niveau d'abstraction avec lequel il est nécessaire de concevoir, a pour contrepartie un coût de développement prohibitif (sauf s'il est envisagé dans une stratégie justifiée le plus souvent à long terme).

2.4 Qualité documentation utilisateur

La meilleure documentation est contextuelle. Le papier doit être réservé aux principes généraux. Le niveau de qualité de la documentation d'utilisation fait l'objet d'un choix justifié techniquement et économiquement. La qualité et la simplicité sont indispensables si l'utilisateur est un néophyte. La concision et la complétude en regard des problèmes complexes sont nécessaires si l'utilisateur est un expert du domaine.

Le niveau et le type de la documentation d'utilisation est une affaire de spécialiste. Usuellement les développeurs sont de mauvais « documenteurs », il est donc souhaitable de faire appel à des spécialistes de cette discipline. Une autre option est de faire rédiger le cahier des charges de l'application dans l'optique d'une documentation utilisateur par un rapporteur issu de la maîtrise d'ouvrage travaillant en partenariat avec un animateur expérimenté. Dans ce cas, les avantages en terme de délais et de budget sont de disposer simultanément et immédiatement de spécifications validées et d'une documentation adaptée.

2.5 Stabilité des spécifications (ou fiabilité)

En général les applications opérationnelles ont des spécifications relativement stables et les applications d'aide à la décision ont des spécifications fluctuantes.

La stabilisation des spécifications est un des éléments clés d'un développement réussi et performant. La compétence de l'animateur est sur ce point cruciale, particulièrement lorsque l'instabilité des exigences est liée à des relations interpersonnelles difficiles. Le rôle de l'animateur (neutre) sera alors de préciser à chaque modification le coût du changement en termes de budget et de délais. En général, la prise de conscience des interlocuteurs face à un chiffrage régulier et systématique, débouche assez rapidement sur un consensus.

2.6 Culture organisationnelle

- Grande entité ou administrative ou « normalisatrice »
- Organisation intermédiaire ou relativement souple
- Petite compagnie ou organisation évolutive ou SSCI
- Mode projet « haute performance » ou activité individuelle

La culture organisationnelle de l'organisation chargée du développement est certainement le point le plus important en terme d'évaluation. De nombreuses études ont démontré des temps de développement variant dans des fourchettes allant de 1 à 50. Ce point à lui seul représente toute la problématique de l'évaluation.

2.7 Administration projet (en %)

La norme est de 15 % à 20 % de l'effort global pour les tâches transversales (gestion avancement, réunions, qualité etc.). RAD de 5 % à 10 %. Le temps consacré à l'administration de projet peut varier dans un rapport de 5% à plus de 30% suivant les exigences des instances dirigeantes. Dans le cas de l'évaluateur cette notion recouvre principalement le temps passé à l'organisation, au recrutement, à la planification, au suivi de l'engagement des ressources, au pilotage, à la production de rapports de pilotage et aux éventuels comités techniques, de suivi ou de pilotage. Sont exclus les temps passés sur les aspects techniques et fonctionnels (la préparation et la réalisation des Focus, par exemple).

La planification et l'ordonnancement de l'engagement des ressources sont la base d'une gestion de projet efficace. 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. Il doit permettre une gestion facile jusqu'au niveau le plus fin de détail (tâche/heure). Il fournit automatiquement des rapports d'avancement suffisamment complets pour satisfaire la direction qui doit apprendre à se contenter de l'essentiel.

Onglet 3 – Conception (Design)

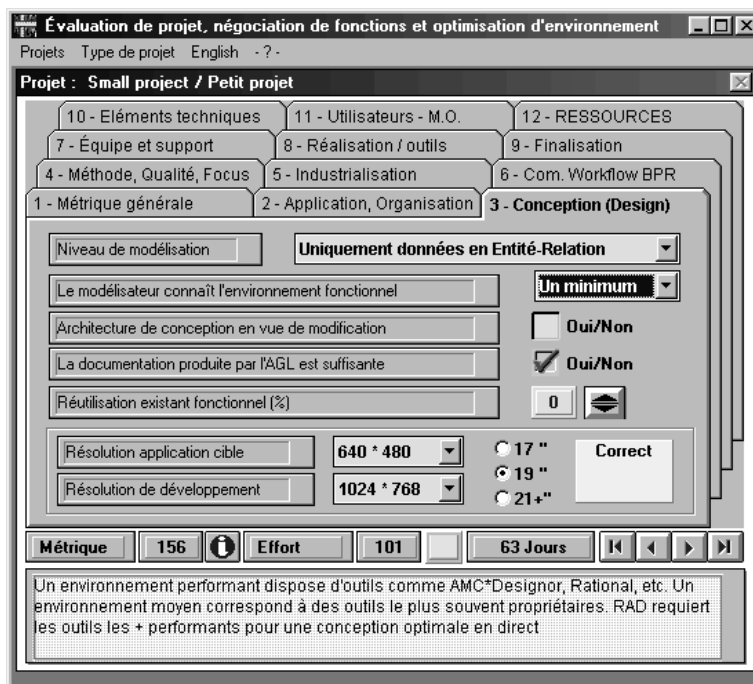


Figure 4. — Onglet 3 : modélisation et architecture

3.1 Niveau de modélisation

- Simple HTML modélisation non applicable
- Uniquement données en Entité-Relation
- E-R + Flux (SASD, Gane-Sarson, etc.)
- E-R + Flux + Traitements détaillés
- Modélisation objet complète OMT/UML

Le niveau de modélisation doit être adapté au type d'application. Il est, par exemple, inutile et disproportionné de mettre en œuvre une approche de modélisation merisienne lors d'une simple application décisionnelle requérant uniquement la définition d'un niveau de structuration des données préalablement existantes.

*Un environnement performant⁵ dispose d'outils comme AMC*Designor, Rose, etc. Un environnement moyen correspond à des outils le plus souvent propriétaires. RAD requiert les outils les plus performants pour une conception optimale en direct.*

3.2 Le modélisateur connaît l'environnement fonctionnel

Au niveau de la conception, et particulièrement pour le décisionnel ou le stratégique, les modèles de traitement merisiens sont remplacés par le principe de la hiérarchie de fonctions. (si le paramètre est non applicable, choisir l'option « Parfaitement »).

Lorsqu'un informaticien dispose d'une expérience fonctionnelle et particulièrement lorsqu'il a déjà modélisé une classe application, il lui est infiniment plus aisé d'obtenir un nouveau modèle parfaitement adapté au cas spécifique tout en bénéficiant des points forts, des détails et des exceptions révélées par son expérience antérieure.

3.3 Architecture de conception « en vue de modifications »

Cette architecture permet d'obtenir un état de livraison permanente et de réaliser des applications évolutives. Concevoir en vue de modification nécessite de connaître plusieurs techniques complémentaires qui se retrouvent généralement dans les approches objets. L'architecture de conception en vue de modification est basée sur les techniques objets : dissimulation, modularité, abstraction, encapsulation, cohésion, couplage, hiérarchie, héritage, polymorphisme, algorithmique, structuration (traitements et données).

Cette option de développement représente de la rigueur, et un investissement qui se rentabilise par la facilité de modification qui en résulte. La justification de cet investissement implique donc un type d'application avec un minimum d'espérance de vie ou un type d'application stratégique nécessitant une forte capacité d'adaptation immédiate.

3.4 La documentation technique produite par l'AGL (de conception) est suffisante

La documentation d'analyse doit être entièrement gérée par l'AGL de conception et doit être considérée comme suffisante⁶. Répondre :

- OUI si la production automatique de l'AGL est suffisante pour les besoins techniques (en général de maintenance) ;
- NON si vous devez produire une documentation de conception manuellement pour satisfaire à divers besoins (qu'il faudra faire justifier ROI à l'appui).

La documentation technique a pour seul objectif de faciliter le développement et la maintenance qui en est le prolongement naturel. On distingue plusieurs formes de documentation technique :

- *La documentation de Cadrage (spécification) inclut généralement sous une forme textuelle le cahier des charges et les divers documents destinés à faciliter la compréhension des exigences.*
- *La documentation Design (conception) inclut sous une forme graphique les modèles de données, de traitements et de communication. Ces informations doivent impérativement être gérées dans un AGL spécialisé permettant de générer automatiquement des rapports sous forme de textes à partir de l'expression graphique des modèles.*
- *La documentation de Construction (codage) doit désormais répondre impérativement aux particularités du mode "Windows" qui distribue des portions de codes dans chaque événement de chaque objet de l'IHM. Pour être réellement utilisable cette documentation doit être totalement intégrée au code qu'elle décrit et structurée au niveau de l'application, du module, de l'objet et de l'événement.*

⁵ Dans des projets de plusieurs millions, plus de 10% du temps global a été gaspillé par l'usage d'AGL de conception qui ne répondait pas aux critères de convivialité Windows. Il faut impérativement utiliser les outils les plus performants, pour concevoir et documenter.

⁶ Le plus important est de réfléchir sur l'usage réel qui sera fait de cette documentation. Rien n'est plus coûteux et démobilisant que la rédaction de documents inutiles ou inutilisables. De toute façon, personne n'aura bientôt plus les moyens de ce gaspillage qui perdure depuis trente ans. Actuellement, une application moyenne représente encore un investissement d'une année ; d'ici cinq ans ce sera l'affaire d'un mois. L'application sera montée en kit (objets de métier).

3.5 Réutilisation existant fonctionnel (%)

Cet existant peut être une ancienne application, ou un modèle de données issu d'une autre application, ou le contenu des pages HTML en format DOC propre. (Dans ce dernier cas, si vous négociez et composez le contenu des pages, choisir l'option « 0 » (zéro) pour ce paramètre).

Lors de la refonte d'une application, étudiez la partie fonctionnelle existante. Lorsqu'elle est satisfaisante (qualité, complétude) et actualisée, elle permet un gain de temps appréciable et évite de nombreux tâtonnements ou erreurs. Dans le cas d'une nouvelle application, il est souhaitable de réaliser un benchmarking.

3.6 Résolution application cible

La résolution CIBLE détermine la résolution de développement. Cet aspect conditionne à la fois l'ergonomie des applications, le confort et la productivité des utilisateurs. En termes de retour sur investissement, il représente la pierre angulaire de la décision (figure 90).

A d é q u a t i o n e n t r e r é s o l u t i o n e t t a i l l e d ' é c r a n					
Résolutions Standards	Taille écran	Nombre de pixels	Accroissement d'information		Types d'applications utilisant ces résolutions
640 * 480	14	307200	--	--	Gestion classique (désuet)
800 * 600	15	480000	64%	--	Gestion semi-graphique
1024 * 768	17	786432	62%	250%	Graphisme, développement
1200 * 1000	20	1200000	65%	400%	Graphisme, dev. haute résolution
1600 * 1200	21+	1920000	62%	625%	Graphisme (PAO, CAO, etc.)
640 * 480 ou 800*600	9 à 12				(Micros portables)

<p>Portables (cas particulier)</p>	<p>Micros de bureau</p>	<p>La superficie de l'écran doit évoluer avec la résolution pour que la taille des caractères reste stable</p>
----------------------------------------	-------------------------	----------------------------------------------------------------------------------------------------------------

Figure

Figure 5. — Optimisation résolution / taille d'écran

3.7 Résolution de développement

La résolution de développement doit permettre un affichage simultané de l'écran cible et des palettes d'outils. Cette possibilité conditionne la performance de développement. Développer professionnellement une application en mode VGA requiert une résolution d'au moins 1024*768 et un écran de 17 pouces. Si l'application cible est SVGA, la résolution de développement doit être XGA appliquée à un écran de 20 ou 21 pouces. De nombreuses études démontrent la rentabilité à court terme (2/3 mois) du couple haute résolution et écran adéquat..

Onglet 4 – Méthode, Qualité, Focus

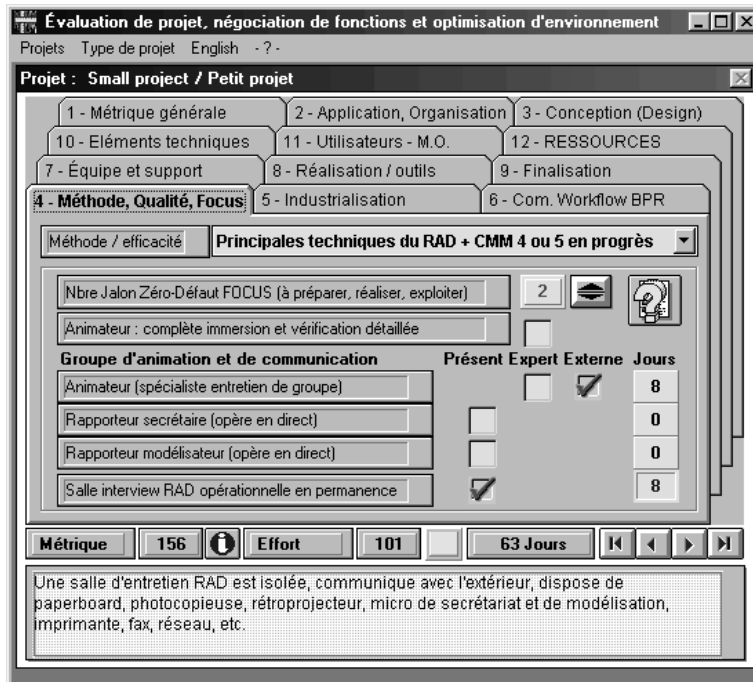


Figure 6. — Onglet 4 : Méthode et groupe d'animation

4.1 Méthode / efficacité

Dans ce domaine, les références d'Évaluateur sont :

- RAD (cycle semi-itératif) ou DSDM, RUP, Iteor (cycle totalement itératif) avec une forme de modélisation adaptée à la problématique (Objet, Flux, Merise « allégé »),
- Merise dans le cycle de vie qui lui est habituellement associé et sa modélisation typique,
- SDM/S (cycle en cascade généralement associé en France avec une modélisation merisienne).

Afin de les situer sur une échelle normalisée Évaluateur répertorie les pratiques mises en œuvre par ces méthodes, par rapport aux 5 niveaux de CMM⁷ ou de SPICE :

- Empiriques, ayant déjà conduit à débordement (correspond à CMM-1bas),
- « Maison », mais n'ayant jamais conduit à débordement de projet (correspond à CMM-1haut),
- Classique (V ou W) Merise, SDMS (correspond à CMM niveaux 2 et 3),
- Principales techniques du RAD (correspond à CMM niveaux 4 ou 5 en progrès),
- Toutes les techniques du RAD (correspond à CMM niveaux 4 et 5).

La meilleure méthode de conduite de projet est sans conteste celle que l'on pratique. Cela dans la mesure où cette méthode n'a jamais conduit au retard ou à l'échec et qu'il est possible de la comparer objectivement avec les résultats obtenus par d'autres.

⁷ Pratiques de planification, d'ingénierie et de gestion qui améliorent la capacité de l'organisation à atteindre des objectifs de coût, de délais, de qualité et de fonctionnalité.

4.2 Nombre de jalons zéro-défaut et de Focus (à préparer, réaliser, exploiter)

Une application légère nécessite un ou deux Focus plus le Focus « final » de fin de recette. Une application moyenne comprend 3 ou 4 Focus.

Le Focus est une présentation de l'application en cours de développement à un niveau réduit mais stabilisé de fonctionnalités. En préalable à un Focus, il est nécessaire d'effectuer un point d'avancement, des revues de code, une intégration des modules, des tests minimums de cheminement et de créer une version de l'application en « état de livraison permanente ». Toutes ces opérations ont un prix qui se justifie par la visibilité offerte sur l'état réel d'avancement des travaux autant que sur la validation générale réalisée au cours de la présentation.

4.3 Animateur (immersion complète et vérification détaillée)

L'animateur (ou facilitateur) est neutre et peut devoir arbitrer entre les maîtrises d'ouvrages et d'œuvre. Pour ce faire, il doit prendre connaissance du périmètre avant le lancement du projet. Choisir cette option a un coût dont la contrepartie s'évalue en termes de sécurité et de qualité (ce ROI n'est pas pris en compte par l'évaluateur). Si l'immersion complète est choisie, l'animateur réalise généralement une validation de l'opportunité du projet.

L'animateur RAD commence sa mission par une immersion minimum dans le secteur fonctionnel (de quelques jours à quelques semaines suivant la taille du projet). Durant cette immersion, l'animateur répertorie les acteurs, élabore un plan de communication, détermine le besoin et réalise une première évaluation.

L'option qui consiste à investir dans une immersion complète de l'animateur semble initialement coûteuse. Dans la réalité ce choix est fortement conseillé et offre en contrepartie une grande sécurité d'opération. Cela se matérialise concrètement par une meilleure visibilité, une qualité accrue, tant fonctionnelle que technique et surtout l'assurance d'une parfaite formalisation et maîtrise des communications du projet.

4.4 Groupe d'animation et de rapport (Présent, Expert, Externe, Jours)

Groupe d'Animation et de Rapport

Le RAD nécessite un animateur spécialisé en entretien de groupe et de rapporteurs travaillant « en direct ».

Les règles internes à l'Évaluateur qui régissent la présence, l'origine et l'expertise des membres du GAR sont les suivantes :

- Plus les rapporteurs et l'animateur sont expérimentés et plus leur efficacité rentabilise leur présence. Mais, si les deux rapporteurs sont des experts, la performance de l'animateur sera moins significative.
- Lorsque les ressources sont externes au projet, elles ne sont plus calculées dans la charge estimée par l'Évaluateur et devront, si nécessaire, être réintégrées dans le budget alloué au projet. En principe, l'animateur « neutre » devrait dépendre d'une instance « neutre » (Direction Générale, Direction Qualité & Méthode, etc.).

Tout cela peut sembler paradoxal ou coûteux, mais dans la réalité, ces investissements concrétisent l'aboutissement d'un pragmatisme à tout épreuve dont dépend directement la performance d'un projet, lorsque ce n'est pas de sa réussite ou de son échec qu'il est question.

Précisez la présence, l'origine et l'expérience des membres de l'équipe d'animation et de rapport :

4.4.1 Animateur (spécialiste entretien de groupe)

4.4.2 Rapporteur secrétaire (opère en direct)

4.4.3 Rapporteur modélisateur (opère en direct)

4.4.4 Salle d'interview RAD opérationnelle en permanence

L'emploi de ressources externes est conseillé (l'animateur doit être neutre).

- La présence de rapporteurs spécialisés réduit l'effort à produire (le savoir-faire est toujours économique). Plus le groupe d'animation est efficace et moins les utilisateurs ou la maîtrise d'ouvrage devront consacrer de temps aux entretiens.
- Il en est de même si, en plus d'être présents, les membres d'un groupe d'animation bénéficient d'une grande expérience, voire d'une expertise en synthèse et modélisation directes du discours utilisateurs.
- Si la ressource membre du groupe d'animation est externe, sa charge n'est pas prise en compte par l'Évaluateur. Ce peut être le cas d'un animateur neutre engagé et rémunéré par la direction générale. Dans le cas où sa facturation incombe au projet, il faudra l'ajouter dans le budget réel. Il faut aussi tenir compte de l'origine de certains rapporteurs qui peuvent être issus de la maîtrise d'ouvrage et qui pourraient alors ne pas être rémunérés directement sur le projet. Tous ces éléments doivent être précisés dans la note de pilotage.

Le groupe dispose en permanence d'une salle équipée pour le RAD. Une salle d'entretien RAD est isolée, communique avec l'extérieur, dispose des outils suivants : *paperboard*, photocopieuse, vidéoprojecteur, micro de secrétariat et de modélisation, imprimante, fax, réseau, etc.

Onglet 5 – Industrialisation

Évaluation de projet, négociation de fonctions et optimisation d'environnement

Projets Type de projet English - ? -

Projet : Small project / Petit projet

1 - Métrique générale 2 - Application, Organisation 3 - Conception (Design)
 10 - Eléments techniques 11 - Utilisateurs - M.O. 12 - RESSOURCES
 7 - Équipe et support 8 - Réalisation / outils 9 - Finalisation
 4 - Méthode, Qualité, Focus 5 - Industrialisation 6 - Com. Workflow BPR

Pondération expérience précédente (ou contexte organisationnel)

Premier projet de ce type pour la Maîtrise d'Ouvrage Oui/Non

Dérive % (+/-) 0 Premier projet RAD prévoir +15%

Outils d'infocentre (MSAccess, BD, Cognos, etc.) Oui/Non

Progiciel / Composants métier (% couvert) 0 100

Réutilisation existant code/objets/données (%) 0 Jours Paramétrage 0

Métrique 156 Effort 101 63 Jours

Ce projet ne nécessite qu'une ressource (ou deux par sécurité ou délais). (calculé pour une équipe de type SWAT performant) 63 jours de travail (selon les ressources précisées)

Figure 7. — Onglet 5 : Industrialisation, progiciel, composants

5.1 Premier projet de ce type pour la maîtrise d'œuvre (O/N)

Après un premier projet RAD, le retour d'expérience permet des gains de productivité importants. C'est en général après le premier ou deuxième projet qu'une équipe se « rode » parfaitement aux principes et outils du RAD. L'accroissement de productivité entre les projets est particulièrement évident dans le cas de réutilisation de templates de transactions ou de rapports.

5.2 Ajustement (%) (pourcentage de pondération en fonction des expériences précédentes)

Pour un premier projet RAD, prévoir +15%.

Le paramètre d'ajustement permet de tenir compte des retours d'expérience. Le retour d'expérience pour un projet ou une organisation similaire est un paramètre d'évaluation intéressant mais à considérer avec précaution. L'expérience bien que représentative d'une réalité objective ne présage en rien de sa « reproductibilité » dans un contexte en évolution (prise de conscience, formation, etc.).

La notion de contexte organisationnel peut aussi être affinée avec ce paramètre. Il est particulièrement utile lorsqu'un modèle d'évaluation a été réalisé (paramètres organisationnels) et est réutilisé systématiquement. Dans ce cas, si des écarts réguliers apparaissent dans la réalisation, il est possible d'en tenir compte :

- soit en modifiant les paramètres semblant incriminés dans la divergence ;
- soit en incrémentant la valeur d'ajustement.

Théoriquement, dans un contexte organisationnel similaire, quelle que soit la métrique, si les aspects fonctionnels et techniques sont correctement pris en compte, cette variable d'ajustement peut correspondre au différentiel de productivité des équipes. En pratique, je déconseille d'utiliser un produit pour juger de la valeur d'une équipe dans un domaine aussi complexe que le développement d'applications.

5.3 Outils d'infocentre (BO, Cognos, MSAccess, etc.)

Par défaut le paramètre est à « OUI ». Passez le paramètre à « NON » dans le cas où vous devez livrer des états et que vous ne disposez pas d'un outil de ce type pour les réaliser.

Un outil d'infocentre réduit le temps de production des rapports et facilite les vérifications à effectuer sur les données. L'utilisation d'outils d'infocentre ou même simplement d'assistants comme ceux de MS Access ou de Crystal Report permet de réaliser très rapidement des états sophistiqués.

5.4 Progiciel / Composants métier

5.4.1 Pourcentage de couverture des fonctionnalités

Dans le cas d'une solution impliquant un progiciel ou des composants métiers pour partie de l'application, indiquez le pourcentage de fonctionnalités qui seront couvertes par la solution préfabriquée et indiquez aussi le nombre de jours de paramétrage, d'assemblage ou de personnalisation incombant à l'équipe projet.

En général, une solution progiciel rentable couvre 80% minimum des fonctionnalités.

5.4.2 Paramétrage

Nombre de jours de paramétrage à la charge de l'équipe projet.

Le recours à un expert « produit » est généralement souhaitable.

5.5 Réutilisation existant (code/objets/données) en %

Procédez comme pour le paramètre progiciel et tenez compte du nombre de jours prévu pour l'intégration ou la personnalisation incombant à l'équipe projet.

Onglet 6 – Communications, Workflow, BPR

Évaluation de projet, négociation de fonctions et optimisation d'environnement

Projets Type de projet English - ? -

Projet : Intermediare project / Projet moyen

1 - Métrique générale 2 - Application, Organisation 3 - Conception (Design)

10 - Éléments techniques 11 - Utilisateurs - M.O. 12 - RESSOURCES

7 - Équipe et support 8 - Réalisation / outils 9 - Finalisation

4 - Méthode, Qualité, Focus 5 - Industrialisation 6 - Com. Workflow BPR

Un courrier électronique (mail) est opérationnel Oui/Non

Un agenda électronique est opérationnel Oui/Non

Une boîte vocale filtre les appels téléphoniques Oui/Non

Une ré-ingénierie des procédés du domaine précède ce projet Oui/Non

Mise en oeuvre d'un workflow procédural Non

La modélisation des communications est déjà effectuée

Développement administration workflow

Métrique 509 Effort 504 336 Jours

Indicateur d'effet de seuil. Le changement de couleur informe d'un accroissement sensible des ressources ou des délais (+ coûts et risques). L'effet de seuil est lié à la charge par rapport à la productivité optimale d'une équipe dans un délais fixe

Figure 8. — Onglet 6 : Outils de communication et réingénierie

6.1 Communications interpersonnelles (logistique et facilités)

Ces outils sont indispensables à la performance d'un SWAT.

6.1.1 Un courrier électronique (mail) est opérationnel

Un courrier électronique utilisé avec rigueur est un outil indispensable aux communications entre le groupe projet et les autres acteurs. Il permet de formaliser des questions « en cours de réflexion » tout en évitant de déranger sans cesse les interlocuteurs. Il garantit des réponses formelles et traçables.

6.1.2 Un agenda électronique est opérationnel

La capacité d'utiliser un agenda électronique partagé (GroupWyse, Lotus) pour organiser automatiquement les rendez-vous et les entretiens est l'outil indispensable du travail de groupe moderne. La contrepartie de cette facilité, lorsqu'elle est mal maîtrisée, c'est qu'elle conduit à des excès et à la « réunionnite ». C'est un problème d'évolution, de culture et d'organisation.

6.1.3 Une boîte vocale filtre les appels téléphoniques

La boîte vocale répond à des préoccupations identiques, mais en sens inverse, car elle protège le concepteur-développeur des dérangements intempestifs. Pour comprendre l'importance de ce point, il faut savoir que dans le cas de réflexion profonde liée à la modélisation ou à la formalisation de règles complexes, une interruption de quelques secondes provoque une perte de temps pouvant atteindre plus de dix minutes. Il suffit d'imaginer la perte de productivité induite par une interruption directe ou indirecte à une fréquence moyenne d'un quart d'heure pour comprendre la nécessité d'isolement de l'équipe projet.

6.2. La modélisation des communications est déjà effectuée

Un modèle de communication du domaine (ou de l'entreprise) peut exister et faciliter le travail de l'animateur du pilote et des coordinateurs.

6.3 Une réingénierie des procédés du domaine précède ce projet (BPR⁸)

Une réorganisation permet d'optimiser les procédés à automatiser et facilite la maîtrise des flux. Une remise à plat des procédés du domaine à automatiser devrait être effectuée avant tout projet informatique. Lorsque cette opération d'optimisation est effectuée, l'informaticien dispose généralement d'une formalisation des informations qui lui sont nécessaires et travaille sur une organisation stabilisée et optimisée. Les développements seront économiques, rapides et parfaitement adaptés.

6.4 WORKFLOW

6.4.1 Mise en œuvre d'un workflow procédural

La notion de Workflow implique dans ce contexte une automatisation de l'enchaînement des procédés.

Le WORKFLOW réserve une place primordiale au BPR. Le raisonnement en termes de données et de fonctions est insuffisant. Une vision étendue, matricielle et globaliste de l'organisation et de ses communications s'impose. Une nouvelle génération d'applications, plus dynamiques dans l'atteinte des objectifs qualité des organisations, a commencé à émerger. Par exemple : sur la base de règles de gestion ou d'organisation, les traitements sont automatisés partiellement ou totalement dans le cas où un acteur impliqué dans le cheminement d'un document fait défaut, une procédure de reroutement attribue le travail à un autre acteur en fonction de diverses priorités. La plupart des applications classiques gagneraient à intégrer quelques procédures de ce type.

6.3.3 Développement administration du workflow

La notion d'automatisation des processus (flux et événements) est la réponse aux besoins de réactivité des organisations. Cette dimension implique une surcouche de procédures (coût de modélisation et de développement).

Très fréquemment les fonctions techniques de suivi de l'évolution des données sont oubliées au début des projets.

⁸ *Business Process Reengineering* se traduit par « Bien Penser avant de Réaliser ». Le primordial est la recherche du meilleur service au moindre coût. Il ne faut plus faire l'erreur de reproduire un existant automatisé, sans tenir compte des nouvelles réalités économiques et techniques.

Onglet 7 – Equipe et Support (technique)

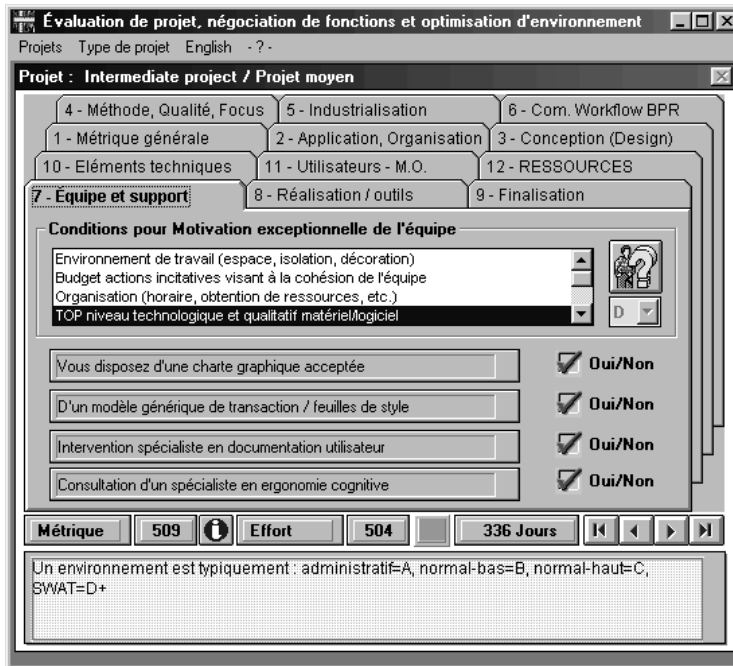


Figure 9. — Onglet 7 : Equipe, motivation et support

7.1 Conditions pour motivation exceptionnelle de l'équipe

- environnement de travail apprécié (espace, isolation, décoration),
- budget pour actions incitatives visant à la cohésion de l'équipe,
- organisation⁹ souple du travail (horaires, obtention de ressources, etc.),
- haut niveau technologique en termes de matériel, de logiciel, de formation, etc.
- coordination démocratique plate de l'équipe projet (choix, risques, décisions, engagement),
- prime correspondant à l'atteinte d'objectifs ambitieux,
- prime significative sur atteinte d'objectifs très ambitieux,
- prime très significative et divers avantages matériels en récompense d'un engagement exceptionnel lié à une exigence exceptionnelle.

7.2 Vous disposez d'une charte graphique acceptée (O/N)

Une charte graphique acceptée par les utilisateurs permet d'uniformiser les applications (facilite la réutilisation et l'apprentissage).

Les bases théoriques de la charte graphique la plus utilisée du futur immédiat s'expriment par un sigle WIMP (Windows, Icônes, Menus, Pointing devices). Pour le généralisable, en termes d'ergonomie et de stylistique, on respecte les normes initiées par Windows¹⁰.

⁹ Une équipe RAD est avant tout autonome et responsabilisée. Autonome dans les choix techniques des solutions et responsabilisée dans la performance de réalisation qui en découle. Ces éléments sont la base de l'identification technique des individus et de l'équipe avec son application.

¹⁰ Le sujet mériterait un exposé complet. Microsoft offre, avec Visual Basic, un guide de « stylistique ». Il détermine, au pixel près, la taille idéale des objets et de leurs espacements.

7.3 Vous disposez d'un modèle générique de transaction (O/N)

Un modèle générique de transaction est nécessaire à un développement fiable, rapide et économique (répondre OUI par défaut pour une application en simple HTML).

La décomposition d'une application en modules (dans le sens Visual Basic) est la base des développements parallèles. Les diverses fonctions organiques de l'application sont normalisées. Schématiquement, on retrouve les modules suivants :

- *initialisation, gestion de la sécurité, personnalisation,*
- *menus de navigation,*
- *aide en ligne, contextuelle,*
- *modules fonctionnels, gestion du spécifique,*
- *gestion généralisée des événements, des erreurs,*
- *gestion des formes d'accès aux données,*
- *interfaces d'intégration.*

Les modules génériques font l'objet d'un développement préalable, de tests et d'une intégration dans un modèle standardisé de transaction. Si l'on ne dispose pas d'un tel modèle, il ne faut pas hésiter à acheter "clé en main" une transaction type à une société de services spécialisée. Avec des outils comme Visual Basic, il est possible de développer séparément, puis d'intégrer des modules complexes sans difficulté.

7.4 Intervention spécialiste en documentation utilisateur (O/N)

Un spécialiste en documentation dispose d'un savoir-faire technique comme ergonomique et il maîtrise les outils qui assurent sa performance. Si non applicable (pas de documentation spécifique à produire) répondre OUI par défaut.

En général, les experts en développement sont de piètres rédacteurs, leur vision est trop technique. Il est plus efficace et moins coûteux de faire appel à une ressource spécialisée. Le secrétariat de projet n'est pas non plus la solution idéale, mais représente un moindre mal.

7.5 Consultation d'un spécialiste en ergonomie cognitive (O/N)

Si l'application concerne un grand nombre d'utilisateurs (utilisation fréquente ou utilisateurs inexpérimentés) l'ergonomie doit être soignée, et le recours à un spécialiste est indispensable.

Depuis l'apparition des écrans, les spécialistes des sciences cognitives et les psychologues se sont penchés sur les principes de l'interface universel¹¹. Certains principes concernant la mémoire et la perception¹² sont à respecter scrupuleusement. Les bases de cette expertise reposent sur la détermination de trois processus fondamentaux : sensoriel (perception), cognitif (interprétation), moteur (action). Pour faciliter la perception immédiate, on respectera des règles basiques de construction :

- *simplicité (nombre et type de composants),*
- *proximité (regroupement des éléments),*
- *continuité (logique de déplacement),*
- *homogénéité (similarité des applications),*
- *guidage (dévoilement progressif, aide contextuelle),*
- *flexibilité (adaptation à la dynamique utilisateur),*
- *contrôle (capacité d'intervention sur action),*
- *feed-back (réaction visible aux commandes).*

Parmi les facilités offertes, la plus simplificatrice¹³ est certainement la manipulation directe¹⁴ de fonctionnalités exprimées sous forme d'objets dont le graphisme respecte des métaphores universelles. L'affichage permanent du statut des opérations en cours réalise un "feedback" permanent vers l'utilisateur. On utilise à cette fin du texte dans une « barre d'état » ou le polymorphisme du curseur. La flèche caractérisant le pointeur se transforme en sablier durant les périodes d'attente, ou en tout autre objet en fonction du contexte d'opération. De nombreux concepts comme : le dévoilement progressif, le droit à l'essai et à l'erreur¹⁵, les liens inter-applications, ou la saisie sans clavier, posent de réels problèmes d'adaptation au programmeur conventionnel. La somme de ces détails constitue la qualité de l'interface et conditionne la productivité des utilisateurs¹⁶. Des dizaines, voire des centaines d'autres recommandations existent. D'une manière générale, la logique prime, et il est nécessaire de les faire valider par un expert en ergonomie à l'écoute des utilisateurs.

¹¹ De nombreux spécialistes ont étudié l'ergonomie cognitive ; il est néanmoins difficile de généraliser. Une application qui concerne des néophytes n'a pas la même allure qu'un outil dédié à des professionnels. L'usage des couleurs sera plus subtil que les simples règles de contraste. Pour un usage occasionnel, les couleurs vives sont recommandées, en revanche dans un écran destiné à un usage permanent, le pastel s'impose. L'utilisateur occasionnel préfère un écran respectant les règles de mémorisation immédiate (sept objets simultanés). Découvrant les enchaînements, il navigue d'écran simple en écran simple. Le professionnel préfère un écran complexe offrant un accès direct et performant aux fonctionnalités qu'il maîtrise.

¹² Organisation « spatiale » dans la distribution des objets, de la gauche vers droite et du haut vers le bas.

¹³ La dynamique de l'interface Windows est événementielle, il ne faut pas en sous-estimer la complexité sous-jacente. La généralisation du multi-fenêtrage pose d'ailleurs de sérieux problèmes d'intégration de modules. Imaginez le bouclage d'une transaction complexe lorsque l'utilisateur peut à tout moment changer de fenêtre, voire éteindre l'ordinateur.

¹⁴ Manipulation directe : voir, pointer, agir. Impact immédiat, un click = une action.

¹⁵ Mode exploratoire, avec avertissement / confirmation des opérations dangereuses. Principe d'opération dit indulgent à l'erreur rendant certaines modifications automatiquement réversibles.

¹⁶ Prise d'option par défaut pour accélérer les choix et la saisie (interface à dévoilement progressif).

Onglet 8 – Réalisation / Outils

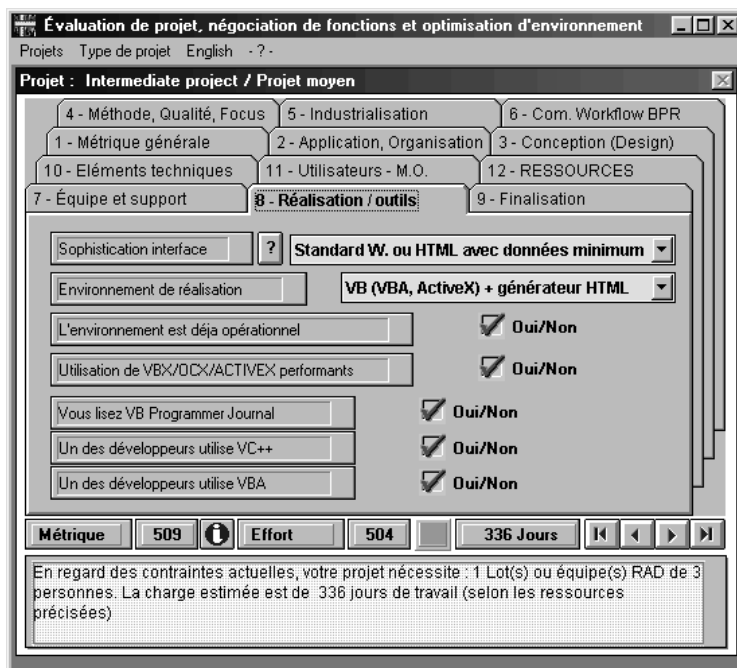


Figure 10. — Onglet 8 : IHM, outils de réalisation

8.1 Sophistication de l'interface

- Basique Windows ou simple document HTML : écrans avec les fonctionnalités minima (génériques) de mise à jour.
- Standard Windows ou HTML avec données minimum : écrans avec, en plus des fonctionnalités génériques, des éléments sophistiqués (objets, boutons, procédures).
- Élevée Windows ou HTML avec accès SGBD : écrans avec des objets offrant à l'utilisateur plusieurs fonctions (tableur, calendrier, aide, onglets, etc.).
- Extrême Windows ou DHTML, Applet, ActiveX : écrans avec plusieurs objets offrant à l'utilisateur de nombreuses fonctions sophistiquées (tableur, calendrier, aide, onglet, import-export, couper-coller, glisser-déposer, etc.).

8.2 Environnement de réalisation (choix des outils)

La performance de développement lors de réalisations complexes ou soignées est dépendante d'un langage standardisé et ouvert aux *add-in* et *add-on*.

- Procédural / pas d'outils spécifiques,
- AGL intégré (*Oracle, PB, etc.*),
- *Delphi* + générateur HTML,
- *VB (VBA, ActiveX) + générateur HTML*.

Le RAD est avant tout une synergie méthodes / outils. Sans les avancées technologiques des quatre dernières années, le RAD n'existerait pas. Au niveau du poste CLIENT, la question Windows ne se pose plus. Après avoir oscillé entre divers outils de front end SQL, il est évident que la productivité se décline maintenant en Visual Basic¹⁷ (VB, VBA, VBScript).

¹⁷ Contrairement à des idées préconçues, les applications en Visual Basic sont robustes. Des centaines de compagnies « tiers » produisent des custom-controls VBX et maintenant des objets OCX. Cette ouverture consacre VB comme l'environnement de développement fonctionnellement le plus riche. D'ailleurs, Oracle Object for Windows, dans sa vision ouverte du client-serveur, confirme la prédominance

8.3 L'environnement est déjà opérationnel (O/N)

Répondre NON si chaque développeur doit installer son poste de travail (environnement de développement) ou s'il est prévu une mise à niveau des logiciels en cours de projet.

8.4 Si vous utilisez *Visual Basic* (cette section s'applique uniquement dans ce cas) :

8.4.1 Utilisation de *ActiveX* performants

(Vous en avez la nécessité). La qualité de l'interface et la performance de réalisation sont liées à la puissance des *add-in* disponibles. Leur choix est un facteur déterminant de productivité.

VB s'avère être un excellent outil RAD (homogénéité avec les applications de Bureautique communicante, universalité de *Windows*), mais il faut être conscient que pour des applications sérieuses, les objets inclus, même dans la version professionnelle, sont insuffisants.

Pour la plupart, leur origine n'est pas Microsoft et ils sont « bridés ». Il suffit pour connaître l'éditeur de déposer un objet dans une forme et d'appeler sa feuille de propriété (F4). Un clic dans le premier champ (About) renseigne immédiatement sur l'éditeur et sur une offre d'achat du produit d'origine complet et de sa documentation. Ce point qui est le gage d'une fantastique ouverture vers le reste du monde désavantage parfois Visual Basic car il est souvent incompris ou assimilé à un danger potentiel en cas d'acquisition de produits tiers dont les fabricants pourraient disparaître. S'il est vrai qu'il faut prendre un minimum de précautions dans le choix de ses fournisseurs, il ne faut pas pour autant rester paralysé. Pour composer un atelier de réalisation performant autour de VB il suffit de faire appel à deux ou trois fournisseurs d'ActiveX, ce qui est un risque acceptable et indispensable (Sheridan, MicroHelp, Visual Component, Crescent, Apex, Intersolv, BlueSky). D'ailleurs la plupart des produits sont à peu près identiques et proposent des méthodes d'utilisation semblables.

8.4.2 Vous lisez *VB Programmer Journal* (www.VBPJ.com) (O/N)

La lecture de revues spécialisées (VBPJ) fournit des solutions (*tips and tricks*) à des problèmes techniques (si non applicable, répondre OUI).

8.5 Un des développeurs utilise *VC++* (O/N)

(Si vous en avez la nécessité). Dans certains il est utile de maîtriser les *API Windows*¹⁸ (si non applicable, répondre OUI).

8.6 Un des développeurs utilise *VBA* (O/N)

(Vous en avez certainement la nécessité). La connaissance de *VBA* permet des liens inter-applications et des solutions remplaçant des développements très lourds (si non applicable, répondre OUI).

de ce langage et des liens Objets OLE. Visual Basic est aussi l'outil de développement graphique le plus répandu. Il séduit aussi bien l'expert que le néophyte. Il représente le maillon entre l'informaticien et l'utilisateur professionnel. VB se retrouve maintenant derrière toutes les fonctions de bureautique communicante. Des applications entières apportent alors leur richesse fonctionnelle en devenant les puissants objets d'une réalisation économique.

¹⁸ Aux Etats-Unis, 83 % des développeurs VB se connectent à un serveur SQL et 44% à un mainframe, 40 % connaissent Visual C++ et 27% écrivent des DLL, 71% utilisent des customs-controls, 7% ont déployé des applications stratégiques sur plus de 500 postes clients simultanés. En France, dans les entreprises, 70 % des développeurs sous Windows produisent des applications Client-Serveur. La dimension moyenne des équipes est de 5 personnes. Dans un proche avenir, 90 % des développements se réaliseront dans un langage de type Visual Basic ou Delphi.

Onglet 9 – Finalisation

Projet : Intermediare project / Projet moyen

4 - Méthode, Qualité, Focus 5 - Industrialisation 6 - Com. Workflow BPR
 1 - Métrique générale 2 - Application, Organisation 3 - Conception (Design)
 10 - Eléments techniques 11 - Utilisateurs - M.O. 12 - RESSOURCES
 7 - Équipe et support 8 - Réalisation / outils 9 - Finalisation

Spécialiste en STRESS d'application événementielle Confirmé

Chemins fonctionnels modélisés par le DESIGN Oui/Non

Les utilisateurs disposent d'un correspondant informatique Oui/Non

Complexité de déploiement Non applicable

Vous utilisez un logiciel de télédistribution (ou NET) Oui/Non

Développement d'outils d'audit de l'évolution des données Aucun(e)

Métrique 509 Effort 504 336 Jours

En regard des contraintes actuelles, votre projet nécessite : 1 Lot(s) ou équipe(s) RAD de 3 personnes. La charge estimée est de 336 jours de travail (selon les ressources précisées)

Figure 11. — Onglet 9 : Validation et déploiement

9.1 Spécialiste en STRESS d'application événementielle

Répondre OUI si vous ne développez que des pages HTML basiques ou des écrans extrêmement simples. Les développeurs ne sont pas généralement aptes à tester leurs applications. Cette expertise doit être développée chez une ressource spécialisée (Source : Microsoft).

9.2 Chemins fonctionnels modélisés par le DESIGN

Parfois, les menus, les écrans et les cheminements fonctionnels ont été modélisés lors de la phase de DESIGN. Dans ce cas, ou si ces travaux ont été effectués par des ressources externes (groupe qualité ou maîtrise d'ouvrage), répondre OUI.

9.3 Les utilisateurs disposent d'un correspondant informatique

La présence d'un correspondant informatique facilite les spécifications et le transfert d'informations. Si l'application est excessivement simple ou si les utilisateurs sont expérimentés et impliqués, répondre OUI.

9.4 Complexité de déploiement

La complexité de déploiement est fonction du nombre de postes, de sites, et d'autres facteurs propres à votre organisation. Si ces tâches sont effectuées par des ressources externes au projet, répondre N.A. (non applicable).

9.5 Vous utilisez un logiciel de télédistribution (ou NET)

Les distributions manuelles sont coûteuses. Si votre application est disponible sur un réseau (local, NET), ou si la mise en exploitation est externe, répondre OUI.

9.6 Développement d'outils d'audit de l'évolution des données

En plus des contraintes d'intégrité référentielle, ce développement lié aux spécificités fonctionnelles ou volumétriques offre à l'utilisateur principal des moyens de contrôle sur ses données (qualitative et quantitative). Bien souvent le développement de ces outils n'est pas prévu dans le projet alors que leur nécessité devient cruciale dès la mise en exploitation de l'application.

Onglet 10 – Eléments techniques

The screenshot shows the 'Onglet 10 - Eléments techniques' window. At the top, it displays the project name 'Intermédiaire project / Projet moyen'. Below this, there are several tabs for different project phases: 7 - Équipe et support, 8 - Réalisation / outils, 9 - Finalisation, 4 - Méthode, Qualité, Focus, 5 - Industrialisation, 6 - Com. Workflow BPR, 1 - Métrique générale, 2 - Application, Organisation, 3 - Conception (Design), 10 - Eléments techniques (selected), 11 - Utilisateurs - M.O., and 12 - RESSOURCES. The '10 - Eléments techniques' section contains several input fields and dropdown menus:

- Équilibre technique équipe:** Value 18.
- Support technique:** Radio buttons for Minimum, Soutenu (selected), and Permanent.
- Complexité transactionnelle:** Dropdown menu set to Moyenne.
- Vous disposez d'un administrateur de SGBD:** Checkmark and dropdown menu set to Oui/Non.
- Les développeurs maîtrisent l'OS Serveur et Client:** Dropdown menu set to La moitié.
- Client-Serveur de troisième génération (N-Tier):** Dropdown menu set to Oui/Non.
- Présence sous-projet de communications:** Dropdown menu set to Simple.
- Haute Disponibilité - Sécurité - Performances:** Dropdown menu set to Non.

At the bottom, there is a summary bar with the following values: Métrique: 509, Effort: 504, 336 Jours. Below this bar, a text box states: 'En regard des contraintes actuelles, votre projet nécessite : 1 Lot(s) ou équipe(s) RAD de 3 personnes. La charge estimée est de 336 jours de travail (selon les ressources précisées)'.

Figure 12. — Onglet 10 : Eléments techniques

10.1 Equilibre technique équipe

- évaluation planification,
- conduite de projet,
- techniques de modélisation,
- langage AGL,
- SGBDR SQL PLS/QL,
- bureautique VBA OLE,
- test et stress d'application,
- documentation.

10.2 Support technique

- MINIMUM, correspond à la présence dans l'équipe d'un spécialiste utilisant depuis plusieurs mois l'environnement de réalisation, les divers outil et *add-in* et *add-on*.
- SOUTENU, correspond à la présence et la disponibilité dans l'organisation d'une équipe RAD (SWAT) complète.
- PERMANENT, correspond à la présence d'une équipe RAD (SWAT) complète et à la disponibilité d'un expert en nouvelles technologies maîtrisant les *add-in* et *add-on* et divers outils.

10.3 Complexité transactionnelle

La complexité transactionnelle est liée à divers éléments (complexité des relations et des tables, niveau de sécurité, données réparties, procédures cataloguées, etc.).

10.4 Vous disposez d'un administrateur de SGBD

Un administrateur de données facilite les manipulations du SGBDR, les « tunings », les affectations de droits et toutes les subtilités des nouveaux SGBDR, etc. (si non applicable, répondre OUI par défaut).

10.5 Les développeurs maîtrisent l'OS Serveur et Client

Exemple Serveur = UNIX, Client = *Windows* (si non applicable, répondre : TOUS).

10.6 Client-Serveur de troisième génération (*Three-Tier*)

N-Tier = Architecture de service distribuée sur plusieurs niveaux (habituellement 3 niveaux : présentation (*Browser*), application (serveur de procédure), données (serveur de données).

L'usage étendu des réseaux engendre un nouveau mode de développement d'applications. On assiste actuellement à une transition du client-serveur classique avec son middleware le plus souvent propriétaire, au client-serveur Intranet ou Internet caractérisé par l'émergence d'un véritable standard ouvert de communication et la notion de client universel. Transparente en termes de type d'application, cette évolution implique néanmoins une logique de conception spécifique. Les éléments principaux de l'application gérés par des processus comme OLE Automation se concentrent alors sur les serveurs (données, services applicatifs), le poste client utilise une couche de présentation standardisée et intégrée dans un seul logiciel appelé navigateur (Netscape, Internet Explorer). Dans cette nouvelle expression de l'entreprise communicante, le modèle Client-Serveur ne peut plus être réduit à l'association de micros et de réseaux, il englobe l'ordonnancement de tâches distribuées, le contrôle de la logique corporative, la gestion de composants applicatifs partagés, la sécurité et l'intégrité des informations. Techniquement, cette approche consiste à séparer les divers éléments qui composent une application en couches de services. Selon la dénomination de Microsoft, les couches se répartissent entre services utilisateurs, services d'entreprise et services de données. Les services utilisateurs fournissent la couche de présentation qui permettra de visualiser et de manipuler les informations à sa disposition. Les services d'entreprise sont les dépositaires des règles de gestion et d'organisation. A la demande d'un utilisateur par l'intermédiaire de sa couche de présentation, ils accèdent aux informations en respectant les règles de cohésion fonctionnelle. Les services de données assurent l'accès, la sécurité et l'intégrité des données.

Cette architecture part d'un principe conceptuel fort pour permettre ensuite un découpage technique¹⁹ des développements par services indépendants qui peuvent être confiés à des développeurs spécialisés (interfaces, fonctionnalités, gestion des données). Cette segmentation représente pour les développeurs une opportunité de dépassement des contraintes engendrées par la généralité d'une conception Client-Serveur classique. Parmi les divers avantages de cette distribution on note :

- *l'émergence d'expertises spécifiques hautement productives des développeurs ;*
- *une modularité permettant la gestion simplifiée de développements parallèles ;*
- *une grande flexibilité dans la répartition de la charge de travail et la gestion du projet ;*
- *un déploiement facilité de l'application et une vision dynamique de son évolution (la logique applicative se limite à des serveurs spécialisés et n'est pas dispersée sur toutes les machines utilisateurs).*

10.7 Présence sous-projet de communications

Communications dans le sens large (multi-sites, EDI, etc.).

10.8 Haute Disponibilité - Sécurité - Performances – Pilotage Risques

Additionner le nombre d'exigences requises par le projet (ces notions méritent une étude approfondie).

¹⁹ La notion d'objets de haut niveau ou d'objets de métier repose sur cette conception. Ces objets sont composés par encapsulation de composants de plus bas niveau. L'ensemble interagit par le biais de messages à travers des interfaces standardisées et publiées (propriétés, méthodes). La structuration des frontières et des interactions entre ces divers éléments est donc un point essentiel de cette conception. Le modèle de services à trois couches est une architecture logique qui s'implante lors du déploiement dans une architecture physique qui peut comporter de multiples couches. La phase de construction doit être précédée d'une étape de recherche de solutions techniques. Réfléchissez avant de programmer !

Onglet 11 – Utilisateur (MOA), Organisation

The screenshot shows a software window titled "Évaluation de projet, négociation de fonctions et optimisation d'environnement". The main area is divided into several sections:

- Navigation tabs:** 7 - Équipe et support, 8 - Réalisation / outils, 9 - Finalisation, 4 - Méthode, Qualité, Focus, 5 - Industrialisation, 6 - Com. Workflow BPR, 1 - Métrique générale, 2 - Application, Organisation, 3 - Conception (Design), 10 - Eléments techniques, 11 - Utilisateurs - M.O. (selected), 12 - RESSOURCES.
- Configuration fields:**
 - Nombre de personnes impliquées en parallèle: > 15
 - Nombre de divisions utilisatrices impliquées: 2 - 3
 - La prestation se réalise sur le site utilisateur: Partiellement
 - En terme de recette, la Maîtrise d'Ouvrage est: Intermédiaire
 - Les utilisateurs sont habitués à l'informatique: Oui/Non
 - Les utilisateurs sont habitués à Windows: Oui/Non
 - Disponibilité des utilisateurs: Moyenne
 - Motivation des utilisateurs: Moyenne
 - Engagement utilisateur: 34 à 60 Jours
- Summary bar:** Métrique: 509, Effort: 504, 336 Jours.
- Text box:** "En regard des contraintes actuelles, votre projet nécessite : 1 Lot(s) ou équipe(s) RAD de 3 personnes. La charge estimée est de 336 jours de travail (selon les ressources précisées)."

Figure 13. — Onglet 11 : MOA et utilisateurs

11.1 Nombre de personnes impliquées en parallèle

Le total des personnes impliquées directement ou indirectement dans le développement, la formation, le déploiement, l'exploitation, la maintenance, etc.

11.2 Nombre de divisions utilisatrices impliquées

Nombre d'entités administratives ou organisationnelles en rapport avec le projet.

11.3 La prestation se réalise sur le site utilisateur

La proximité géographique entre les développeurs et les utilisateurs est primordiale en ce qui concerne les exigences de communication du RAD.

11.4 En termes de recette, expérience de la maîtrise d'ouvrage

Le RAD déporte une grande partie des validations en phase de DESIGN et de CONSTRUCTION. L'expérience des utilisateurs est sur ce point déterminante (habitude découlant de plusieurs projets).

11.5 Les utilisateurs sont habitués à l'informatique

Bien que cela soit du ressort de la maîtrise d'ouvrage, cela affectera votre projet (si non applicable, répondre OUI).

11.6 Les utilisateurs sont habitués à Windows

Si votre interface est non graphique ou si cela ne vous concerne pas, répondre OUI.

11.7 Disponibilité des utilisateurs

La qualité de la validation dépend de la disponibilité des utilisateurs (option « excellente » par défaut si ce paramètre est non applicable dans votre cas).

11.8 Motivation des utilisateurs

La motivation des utilisateurs est primordiale. L'ensemble du processus se base sur leurs connaissances et la volonté de transmettre l'information représentative du besoin tout en s'appropriant l'application (si non applicable dans votre cas, choisir l'option « excellente » par défaut).

Onglet 12 – RESSOURCES / PLANIFICATION

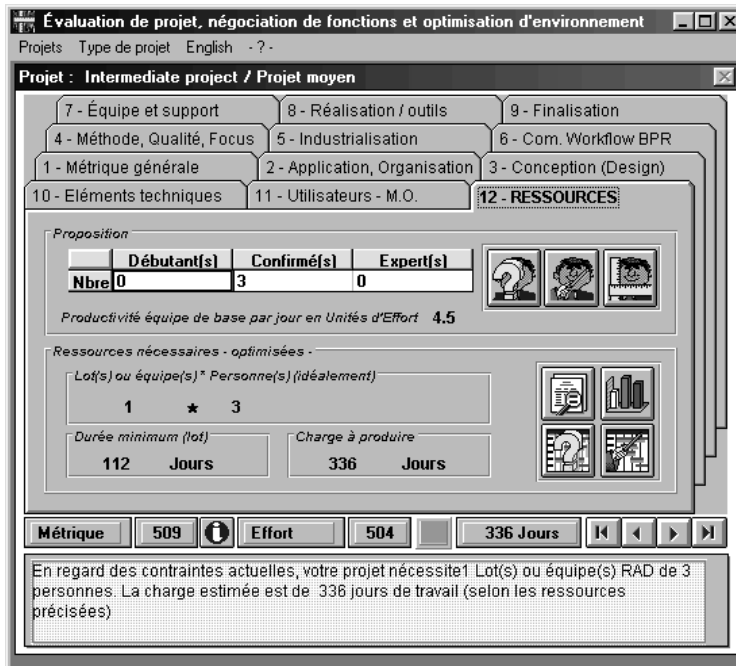


Figure 14. — Onglet 12 : Ressources et planification

- Allocation des ressources (Manuelle / Automatique).
- Application de contraintes (Délais / Budget).
- Planification / ordonnancement des phases.

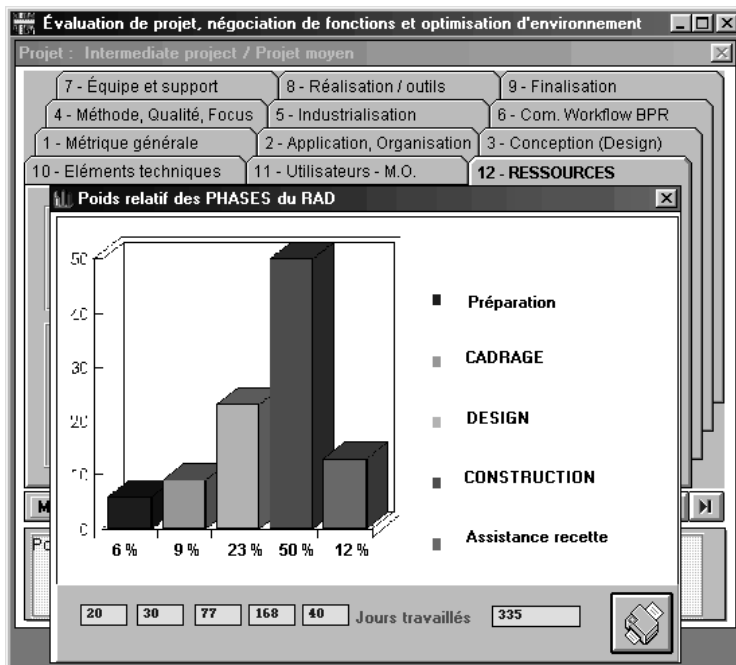


Figure 15. — Poids relatifs et théoriques des phases du RAD

Le pilote d'un projet RAD fonctionne avec un triangle ayant pour côté trois types d'éléments : des ressources (personnel, argent), des fonctionnalités (le produit et sa qualité) et le planning. Toute modification apportée à l'un des côté a des répercussions sur un autre des côté et le plus

souvent sur les deux. Un problème de planning ne dispose que de trois axes de solution pour être résolu : ajout de temps, de ressources ou réduction de fonctionnalités.

Pour rattraper un retard vous pouvez soit réduire les fonctionnalités à livrer, soit augmenter les ressources. L'ajout inopportun d'individus à un projet peut produire un ralentissement du travail en cours.

Selon les statistiques issues de projets RAD, un écran présente la distribution théorique de la charge à produire en fonction des phases de la méthode RAD (*figure 100*). Un assistant d'optimisation (*figure 101*) permet de faire évoluer des scénarios en fonction des contraintes du projet. Son intérêt principal réside dans la mise en évidence immédiate des incidences de chaque option antagoniste envisagée.

	Débutant(s)	Confirmé(s)	Expert(s)	Total	Lot(s) ou équipe(s)
Nbre	0	3	0	3	1

Figure 16. — Optimisation (budget, délais, visibilité, qualité)

Conseil : ne tombez pas dans le piège qui consiste à jouer avec les paramètres d'environnement pour tenter d'adapter l'évaluation à des désirs ou à un budget.