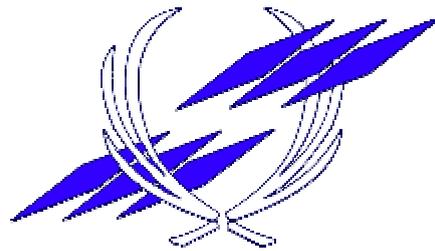


Mastère Spécialisé HEC – Mines de Paris
Management des Systèmes d'Information et des Technologies



Thèse Professionnelle

**Méthodes d'Estimation
de Charges**
dans le cadre d'un projet xNet

Georges Zadrozynski

*Cette thèse professionnelle a été réalisée à la Direction des Systèmes d'Information de la **Caisse des Dépôts et Consignations** (Etablissement Public), dans le cadre de la formation du Mastère Spécialisé «Management des Systèmes d'Information et des Technologies», conduite conjointement par l'Ecole des Hautes Etudes Commerciales (**HEC**) et l'Ecole Nationale Supérieure des Mines de Paris (**ENSMP**).*

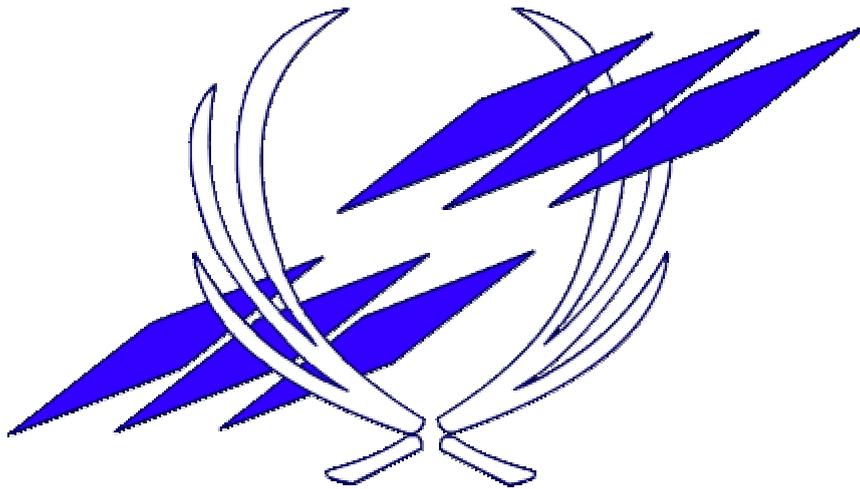
Responsable Académique : **Alain Berdugo**
(Professeur à HEC, Directeur du Mastère MSIT)

Responsable Entreprise : **André Probst**
(Responsable Méthodes de la DSI CDC - Etablissement Public)

Copyright © 2002 Georges Zadrozynski
Tous droits réservés

<http://www.gezzed.net>
gz@msit.org

Méthodes d'Estimation de Charges dans le cadre d'un projet xNet



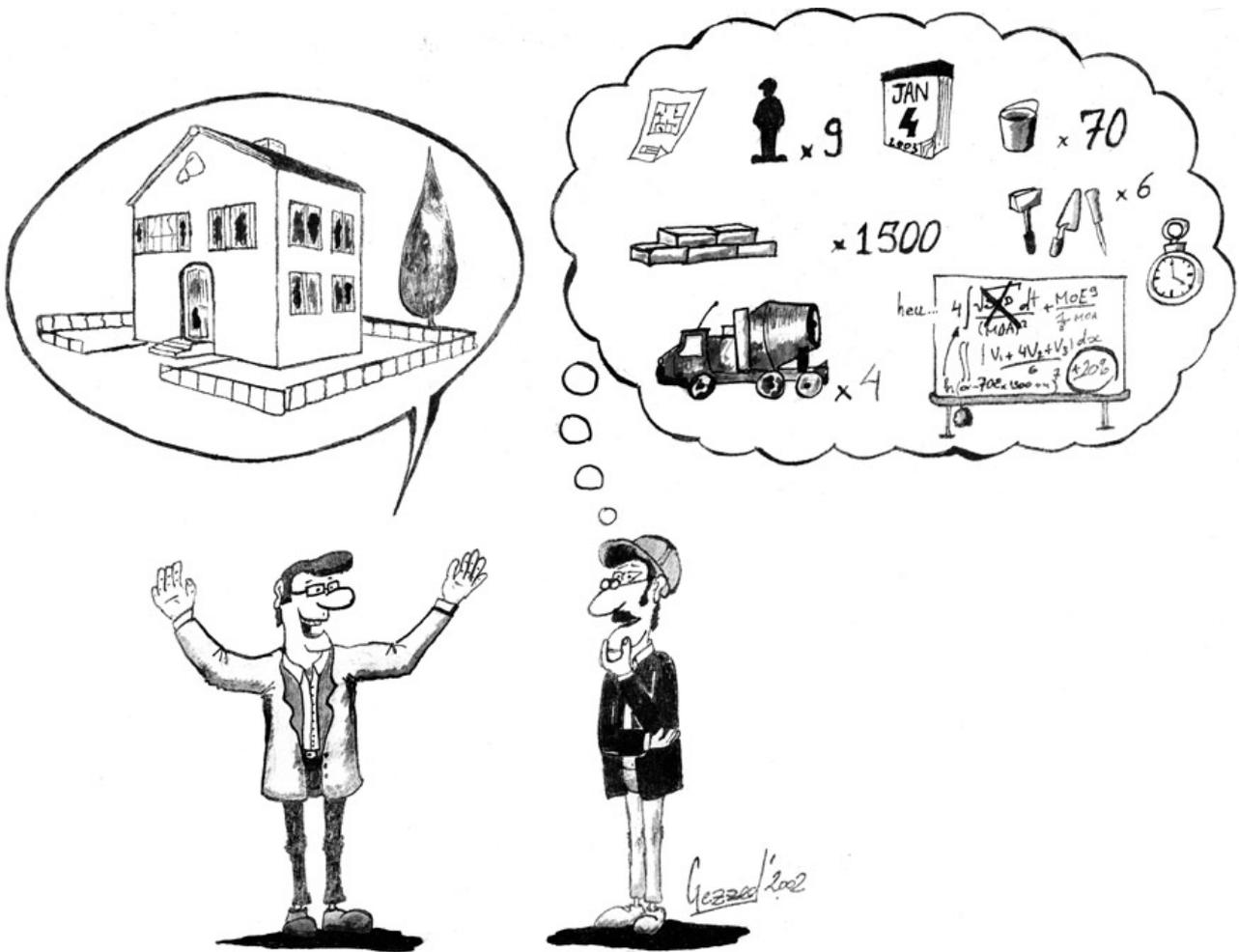


Table des Matières

TABLE DES MATIERES	5
REMERCIEMENTS	6
INTRODUCTION.....	7
I. L'EVALUATION DE CHARGES D'UN PROJET INFORMATIQUE.....	8
I.1. POURQUOI EVALUER LES CHARGES ?	8
I.2. LE CONCEPT DE "JOUR*HOMME"	8
I.3. LES PIEGES DE L'EVALUATION THEORIQUE.....	10
II. LE PROJET XNET	12
II.1. LE XNET.....	12
II.2. CARACTERISTIQUES INTRINSEQUES D'UN XNET.....	16
III. ETAT DE L'ART DES METHODES D'EVALUATIONS.....	20
III.1. METHODE DES POINTS DE FONCTIONS.....	20
III.2. METHODE COCOMO.....	26
III.3. EVALUATEUR-RAD (J-P. VIKOFF).....	36
III.4. METHODE INTERNE CDC (A. PROBST)	42
III.5. METHODE INTERNE ICDC (ESTIMAT)	51
IV. EVALUATION D'UN XNET	56
IV.1. INTRODUCTION.....	56
IV.2. CREATION D'UN XNET & AJOUT DE MODULES.....	57
IV.3. ALORS COMMENT EVALUERLE XNET ?.....	58
V. UN EXEMPLE CONCRET : RPI.....	59
V.1. PRESENTATION	59
V.2. ESTIMATIONS THEORIQUES DE LA CHARGE DE RPI.....	65
V.3. DEROULEMENT CONCRET	71
V.4. CONCLUSION.....	72
CONCLUSION	74
LA BONNE METHODE ?	74
QUELQUES BONNES PRATIQUES.....	74
ANNEXES	76
BIBLIOGRAPHIE.....	76

Remerciements

Je tiens à remercier particulièrement :

Alain Berdugo

Robert Mahl

... tout d'abord pour leur précieux soutien, constant, et cordial tout au long de l'année, mais aussi, pour avoir eu un jour, l'extraordinaire idée de créer le Mastère MSIT.

André Probst

... de m'avoir confié un sujet aussi passionnant, et d'être à tout moment disponible.

Catherine Ritz

Guy Audoli

... d'être toujours de bonne humeur, et d'excellents managers, tout simplement.

Josselin Requillard

Catherine Yvonnou

Mounir Khaldi

... pour leur soutien, leur chaleureuse présence au quotidien, et leur inépuisable potentiel de sympathie.

François Malliaros

Arnaud Ventre

Anne Rias

François-Xavier Bouquet

... pour leur bonne volonté et leur indispensable collaboration plus qu'encourageante à cette thèse.

Introduction

Lorsque l'on parle d'"Estimation des charges" de projets informatiques à une Maîtrise d'œuvre ou à une Maîtrise d'ouvrage, chacun a déjà plus ou moins une idée sur la question... une méthode à lui qu'il utilise, ou une vieille formule gribouillée sur une feuille (dont, d'ailleurs, on ne sait plus très bien d'où elle vient), mais qui a fait ses preuves, et que l'on continue à utiliser... ou alors, la bonne vieille méthode du "nez" du chef de projet, que l'on retrouve tout aussi fréquemment lors de l'estimation de charges de certains projets.

Evidemment, une entreprise qui a déjà fait un certain nombre de projets informatiques, connaît plus ou moins les charges et les délais dont elle aura besoin pour faire un projet, dont la plateforme technique et les règles fonctionnelles lui sont familières. Ces "méthodes", d'une efficacité relative, n'en restent pas moins intuitives, et par là, perdent une certaine crédibilité lors d'une justification officielle de l'estimation.

Il existe pourtant des méthodes reconnues et efficaces d'estimation de charges de projet informatiques. Ces méthodes ont eu le temps de faire leur preuve dans divers domaines. La plupart d'entre elles se basent sur l'expérience d'un certain nombre de projets-types analysés.

Jusqu'à présent, ces méthodes étaient relativement fiables, vis-à-vis des projets informatiques "classiques", qui pouvaient se faire jusqu'au milieu des années 1990 – en l'occurrence, les applications monopostes, ou en client-serveur.

Mais depuis moins d'une dizaine d'années, et avec l'évolution exponentielle des technologies et l'apparition des xNet, les règles du jeu ont changé, une fois encore.... Toutes les entreprises ont aujourd'hui leur internet, tous les grands comptes leur intranet, et la majorité de commerces leur extranet.

La mesure des charges de ces xNet est une "science" bien trop récente pour pouvoir en parler avec certitude, et sans risquer de se tromper. Mais cette mesure devient aujourd'hui plus qu'indispensable, et l'utilisation d'une méthode sûre, et reconnue dans le cadre des xNet, pourrait aider à poser des bases concrètes de comparaison, pour les MOE, les MOA, et, pourquoi pas, les SSII.

Nous allons donc nous plonger dans l'univers des xNet, tout d'abord, puis dans celui de l'évaluation des charges, ensuite. Et grâce à un exemple concret, nous verrons s'il est possible de les faire cohabiter, et avec quelle fiabilité.

I. L'Evaluation de Charges d'un projet Informatique

I.1. Pourquoi évaluer les charges ?

Dans le cadre des projets de la nouvelle économie, le concept de l'xNet est devenu quasi-incontournable : toutes les grandes entreprises possèdent aujourd'hui un intranet, et la conception (et le développement) de modules s'y greffant est le lot quotidien des responsables informatiques de ces entreprises.

Une fois le "projet" défini par les responsables du métier, ces modules devront être réalisés (en passant par des étapes successives, telles que les spécifications fonctionnelles globales, les spécifications fonctionnelles détaillées, le cahier des charges, la conception de la base de données, la réalisation, la recette, etc.).

Il est bien entendu que ces étapes prennent du temps, aussi bien du côté de la MOE que celui de la MOA, et même de l'exploitation... et parfois beaucoup plus (ou beaucoup moins) que les métiers et la maîtrise d'ouvrage ne pourraient l'estimer.

Pour pouvoir accorder un budget, il est nécessaire d'avoir une vision, même globale, des différents coûts engendrés, et du temps nécessaire à la réalisation.

Ainsi, ces méthodes d'évaluation des charges aident les responsables métiers à apprécier non seulement la charge, mais aussi la quantité de temps nécessaire à la réalisation, la taille de l'équipe, etc.

I.2. Le concept de "Jour*Homme"

Le "Jour*Homme"¹ est l'unité de mesure de la charge de travail dans le contexte d'un projet².

Concrètement, si un projet est estimé à "cent jours pour une personne à plein temps" pour arriver à son terme, on pourra considérer qu'il s'agit d'un projet estimé à "100 jours hommes"... comme l'indique l'intitulé du résultat de l'évaluation, si l'on confie la réalisation de ce projet à un unique homme, ce dernier devrait –sauf erreur d'estimation, et sauf événements extérieurs perturbant la réalisation du projet– passer exactement cent jours à le réaliser.

Cependant, nous pouvons nous permettre de supposer que tout projet, à un certain niveau, est "scindable" : c'est-à-dire que l'on va pouvoir le décomposer (dans la mesure où ce projet est

¹ La syntaxe Jour/Homme est admise, mais erronée sur la forme: Il ne s'agit pas, comme pourrait le laisser penser cette écriture, d'un nombre d'homme par jour. La syntaxe Jour*Homme est beaucoup plus près de la réalité, puisqu'en effet, le nombre obtenu résulte bien –théoriquement- de la multiplication d'un nombre de jours par un nombre d'homme. Jour/Homme, Jour*Homme, et Jour-Homme, sont la plupart du temps utilisés indifféremment en Entreprise et dans de nombreux ouvrages. Pour rester cohérents, nous utiliserons la syntaxe qui semble la plus adaptée, donc "Jour*Homme".

² Il est également possible d'utiliser des concepts alternatifs de mesure, tel que le "Mois-Homme" (appelé également l' "Homme-Mois"), que l'on pourra considérer équivalent, de façon linéaire, à 21 Jour*Homme.

de taille intuitivement suffisamment élevée) en un ensemble de tâches. Et ceci est aussi vrai dans les projets informatiques "classiques", que dans les projets de la nouvelle économie.

Si ce projet est partitionnable (et ce sera quasiment toujours le cas), il est raisonnablement possible d'effectuer plusieurs de ces tâches en parallèle – donc de les faire effectuer par plusieurs hommes simultanément.

Partant de ce postulat, et dans un cadre idéal (figure I.2.1), on peut donc considérer qu'un projet de 36 J*H (planifié pour 12 hommes sur 3 jours) pourra être effectué en 9 jours par 4 hommes.

Cette répartition idéale suit une parabole. Nous verrons cependant un peu plus tard que cette théorie est loin d'être exacte dans le domaine qui nous intéresse, c'est à dire celui des projets informatiques de la nouvelle économie.

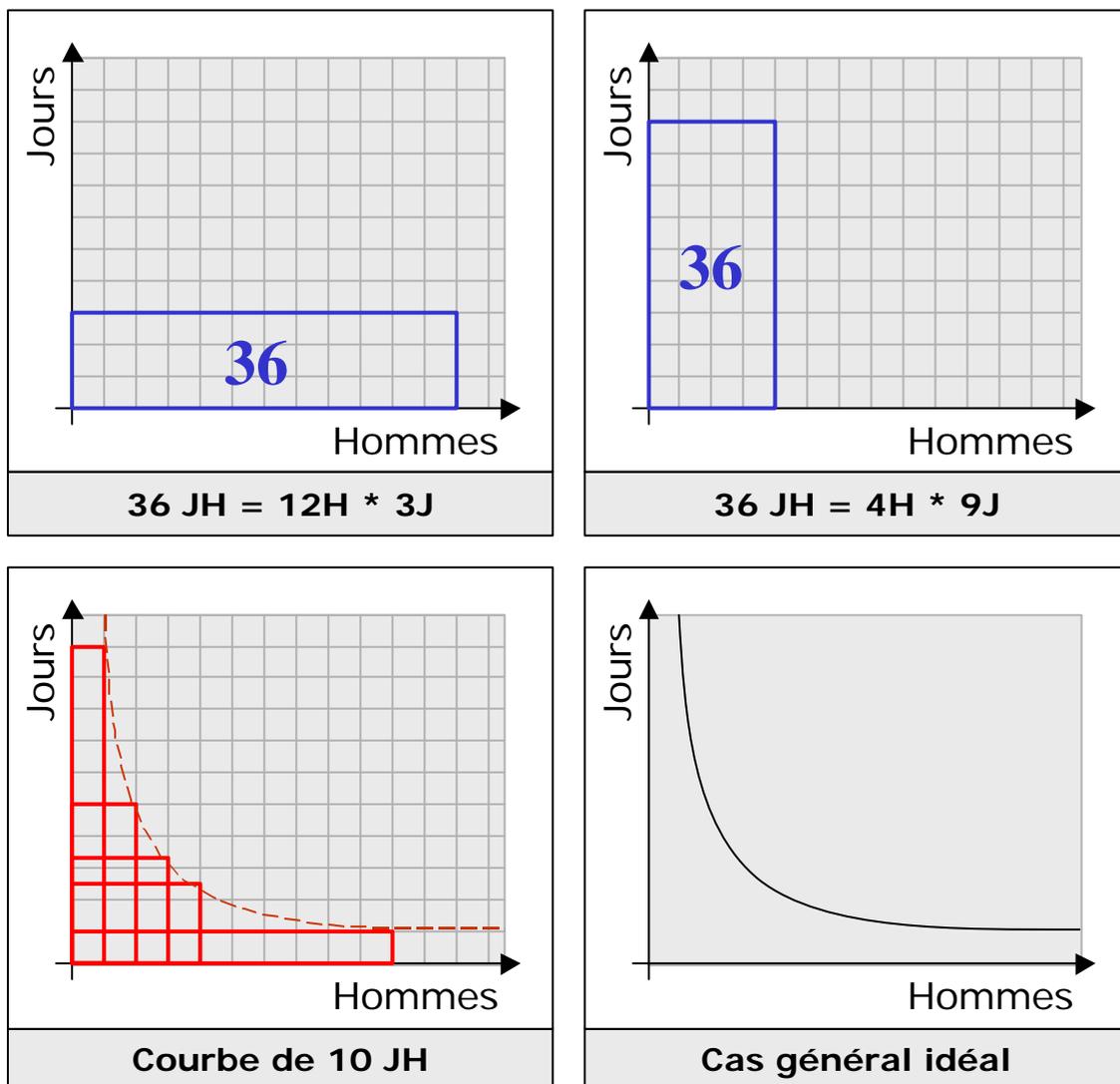


figure I.2.1 – Le principe du "Jour*Homme"

1.3. Les pièges de l'évaluation théorique

1.3.1. La loi de Parkinson sur le temps de travail

Une des lois de C.N.Parkinson³ stipule que "les programmes sont comme les gaz parfaits : Ils prennent toute la place qu'on leur donne". L'interprétation de ce postulat est extrêmement simple : Si l'on donne à une équipe un projet de 50 Jours*Homme (valeur "réelle") à faire sur 100 Jours*Homme, le projet sera fait en 100 Jours*Homme, et pas un de moins. (Et donc, vous l'imaginez, facturé 100 Jours*Homme, bien entendu.)

Cela fonctionne également dans l'autre sens : il est possible de réduire sensiblement la durée théorique d'un projet, et obtenir un résultat dans le délai escompté. Ce n'est cependant pas une méthode recommandée par la plupart des manuels de gestion de projet, et par les MOE eux-mêmes.

Si l'on part de ce principe d'analogie avec les gaz parfait, on peut supposer qu'à peu près toutes les estimations de charge possibles (dans la mesure de l'intuitivement raisonnable) sont correctes.

Il s'agit cependant bel et bien d'un piège dans lequel il est nécessaire d'éviter de tomber, en effectuant les estimations de façon, à ce que la charge supporte une "pression" raisonnable et suffisamment confortable pour la maîtrise d'œuvre et la maîtrise d'ouvrage.

De plus, la plupart des responsables MOE (étant tout à fait conscients de cette loi, et en tenant compte quotidiennement), avancent l'argument suivant : il est beaucoup plus indiqué de laisser un peu plus de temps à la MOE qu'on ne pourrait le juger nécessaire suite à une estimation.

En effet, la plupart des informaticiens (qui, donc, ont tendance à effectuer le travail dans le temps imparti) consacrent le temps supplémentaire accordé à l'élaboration d'une ergonomie beaucoup plus poussée, et qui –de fait- satisfiera davantage l'utilisateur final, ou bien dans la construction d'un code plus facilement maintenable, etc.

1.3.2. Le mythe du partage des "Jours*Hommes"

Si nous reprenons l'exemple du paragraphe sur le concept de Jours*Hommes, (le projet de 100 J*H), et que nous nous basons sur ses conclusions, nous pourrions effectuer le postulat que ce projet de 100 J*H pourrait être effectué par 200 Hommes en une demi-journée. Bien que ce soit mathématiquement exact, c'est une aberration évidente : Les projets de l'informatique, en général, ne peuvent pas être scindés de cette manière.

Ce type de raisonnement s'applique par contre, parfaitement, à des "projets" très spécifiques, comme la peinture en bâtiment, la récolte agricole, etc.

A l'extrême opposé, il existe également des projets que ne sont pas partitionnables.

³ Dans l'ouvrage *Parkinson's Law* (C.N. Parkinson), Penguin Books. ISBN : 0141186852

L'exemple revenant le plus souvent est celui –il est vrai, assez simpliste, mais très illustratif- de la femme enceinte, qui va mettre neuf mois à mettre au monde un enfant... Il est, de façon évidente impossible de partitionner, en plus d'une tâche. – Même trois femmes dotées de la meilleure volonté du monde ne pourront faire un enfant en trois mois.

Nous nous devons alors de faire des distinctions en plusieurs types de projets (*figure I.2.2*), dont l'attribution des ressources devra différer de façon significative (et ce, en fonction du nombre de personnes supposées travailler sur ce projet.)

Après avoir abordé les extrêmes (*figure I.2.2.A* : Projet totalement partitionnable, et *figure I.2.2.B* : Projet non partitionnable), nous pouvons nous attacher à analyser une modélisation plus réaliste. (*figures I.2.2.C & D*)

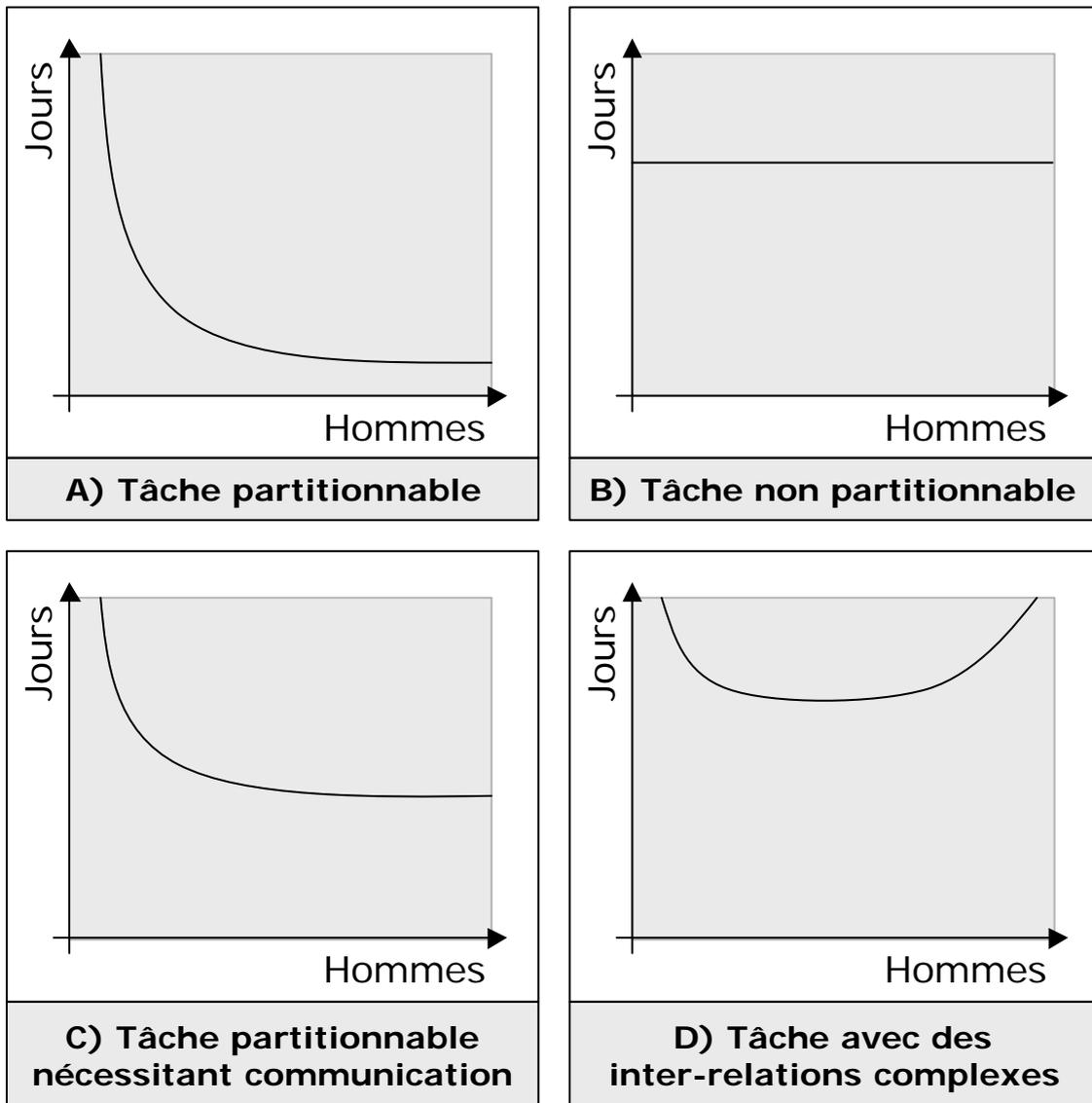


figure I.2.2 – Affectation des charges aux tâches

II. Le projet xNet

II.1. Le xNet

II.1.1. La philosophie xNet

Le "xNet" (ou *Net⁴) est une appellation générique du Système d'Information de front-office de l'entreprise. Il peut désigner un *Internet*, un *Extranet*, ou encore un *Intranet*, voire une combinaison de ces modèles de réseaux.

La différence fondamentale entre ces **Internet**, **Extranet**, et **Intranet**, est principalement liée au public (aux "visiteurs") visé par ces réseaux, et par voie de conséquence au type d'accès de ces utilisateurs à ce que nous appellerons le "moteur de distribution de contenu"⁵.

La plupart du temps, la technologie utilisée dans les xNet est la suivante :

Un serveur HTTP⁶ fournissant du contenu HTML, indifféremment statique ou dynamique, et interactif. A cela s'ajoutent divers services⁷ tournant derrière les serveurs HTTP, tels que des serveurs de données, ou de fichiers, et plus généralement quasiment n'importe quelle application pouvant potentiellement être assurée en back-office par le système d'information.

La frontière entre les trois concepts d'*Intranet*, d'*Internet* et d'*Extranet* s'avérant parfois extrêmement ténue, certains professionnels préfèrent utiliser le terme d'"**Application xNet**" pour désigner un système de "site" interactif mêlant indifféremment un ou plusieurs des trois concepts.

On peut, par extension de cette philosophie, considérer le xNet comme une application répartie dont le fonctionnement pourrait être modélisé comme suit :

D'un côté, un ou plusieurs "clients légers" (navigateurs dédiés, tels que Microsoft Internet Explorer, Mozilla, Netscape, Lynx, Opera...) sur lesquels s'affichent une interface utilisateur, très proche du principe des interfaces standard (Windows ou MacOS), et dans lesquels le

⁴ *, par analogie avec l'astérisque des informaticiens désignant «tout». xNet, ou *Net signifie donc "Tout ce qui se termine par 'Net'" (*Attention : Certaines appellations sont de façon évidente non comprises dans cette définition. I.e.: Ethernet*)

⁵ Ensemble arbitraire des technologies permettant l'identification d'un utilisateur, et la navigation et les droits d'accès aux divers documents en fonction de cette identification. (Il découle du module d'identification et est intimement lié au module de navigation)

⁶ HyperText Transfer Protocol : Protocole permettant, principalement, de délivrer des fichiers plus ou moins standardisés (statiques ou dynamique) sous forme de flux, interprétable par des clients dédiés.

⁷ Ces services peuvent être très diversifiés : interpréteurs ASP ou PHP, liaisons avec diverses bases de données, serveurs d'applications, etc.

client verra s'afficher des objets graphiques lui permettant d'effectuer des opérations. Le résultat des opérations effectuées s'affichera de même dans cette fenêtre de "client léger"

De l'autre côté, un ou plusieurs "serveurs", qui auront pour tâche de récupérer les ordres envoyés par le client, de les traiter⁸ (sans restriction de champ d'action), et de renvoyer au "client léger" le résultat des opérations.

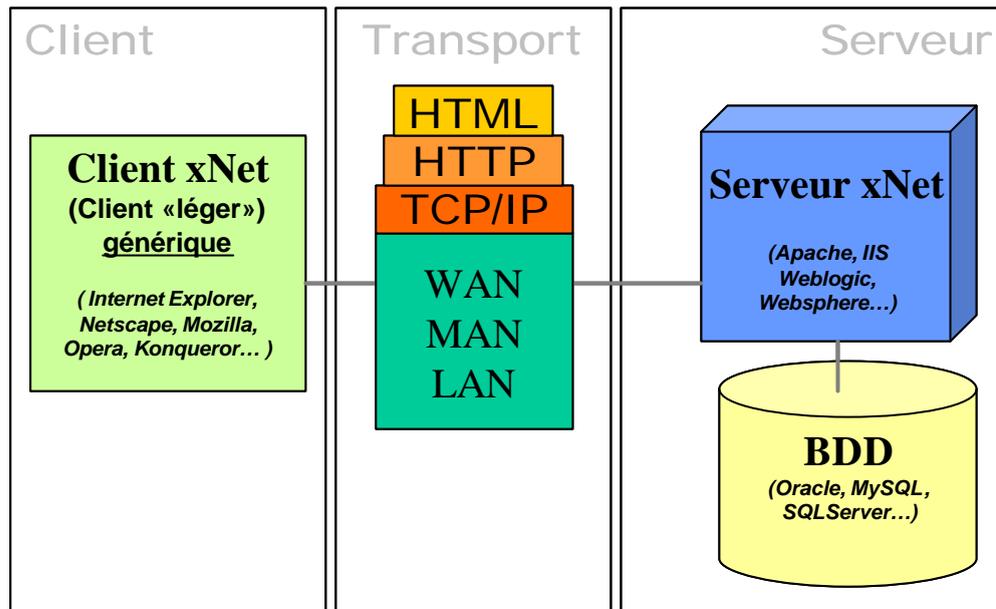


figure II.1.1 – Modèle simplifié d'architecture des xNet

II.1.2. Internet

Typiquement, un site dit "Internet" aura pour vocation de toucher toutes les personnes capables de s'y connecter physiquement (via le réseau du même nom.)

Il n'y a, à priori, aucun processus d'identification des visiteurs du site, à l'exception près (éventuelle mais rarissime) d'un mot de passe générique permettant - de façon binaire - un accès total, ou bien, aucun accès, à un site. (ou à un répertoire de ce site)

Ce type de site peut délivrer de façon indifférente des pages **statiques** ou **dynamiques**.

Il est important de noter que les visiteurs de ce site sont à priori inconnus : ce sont des clients potentiels, ou bien encore des clients actuels, voulant consulter un support technique (dans le cas d'une entreprise.) Il peut également s'agir de particuliers consultant des informations,

⁸ Notons que certains ordres ne sont pas envoyés sur le serveur, mais sont, la plupart du temps, quand la logique le permet, traités sur le client (cas des scripts de type JavaScript, ou VBScript), pour des raisons de gain de temps d'accès au serveur. Ces programmes sont implémentés directement à l'intérieur du code de l'interface destiné au client léger.

telles que les informations, la météo, ou bien tout simplement le contenu de pages construites par des particuliers.

En bref, il s'agit bel et bien de l'Internet tel qu'il est défini de nos jours par le grand public et les média : L'information en libre accès à monsieur tout-le-monde.

Quelques exemples :

<http://www.meteo.fr> (Site d'information de Météo France)

<http://www.google.fr> (Moteur de recherche)

II.1.3. Intranet

Un site "Intranet" est lié à un accès extrêmement restreint : le nombre de visiteurs potentiels est parfaitement connu, ainsi que l'identité de ces derniers. Ce type de site requiert en général des technologies dynamiques, et apporte des services conséquents à ses visiteurs, dans le cadre, par exemple, d'une organisation.

Il n'est accessible qu'à partir d'un réseau local (réel ou virtuel⁹), et donc d'un nombre de machines connu, chacune d'entre elles étant identifiée par son adresse MAC¹⁰. La plupart du temps, cet intranet est lié à un module d'identification, permettant aux utilisateurs de se faire reconnaître par le serveur de façon unique. A cette identification sont très souvent liés les profils et les droits de consultation des documents distribués sur cet Intranet.

Il est à noter que ce type de site n'est pas, à priori, accessible de l'extérieur de l'organisation.

Exemple :

<http://intradfepriv-p.dfci.cdc.fr> (Intranet de la Caisse des Dépôts et Consignations – n'est accessible qu'à partir de l'un des postes de travail à l'intérieur de la CDC)

II.1.4. Extranet

Un site "Extranet", plus restreint qu'un site Internet, sera destiné à un public possédant un mot de passe, ou une clef permettant de consulter ses données. Le plus souvent, ce type de site est dynamique, et les visiteurs sont à priori connus, ou, en tout cas, répertoriés.

On peut sans arrière-pensées simplement considérer l'Extranet comme un Intranet "ouvert".

Exemples :

<http://www.amazon.com> (Vente en ligne de produits culturels),

<http://www.theplasticsexchange.com> (Place de marché B2B pour les matières plastiques)

⁹ Soit en LAN, ou en VLAN. C'est pratiquement toujours le cas.

¹⁰ Adresse MAC, ou adresse Ethernet : Il s'agit d'une adresse physique attribuée par le constructeur à la carte réseau d'un ordinateur. A priori, on peut considérer que chacune de ces adresses est unique.

II.1.5. Conclusion sur les xNet

Par extension, certains voient l'Internet et l'Intranet comme étant des cas particuliers de l'Extranet :

- L'Intranet serait un Extranet consultable exclusivement dans le cadre d'un réseau fermé.
- L'Internet serait un Extranet dont on aurait supprimé les contraintes d'identifications appliquées au moteur de distribution de contenu.

La figure II.5.1 illustre de façon simple les différences entre les trois types xNet existants.

	Module d'Identification	Type d'Accès en Réseau
Internet	<i>NON</i>	<i>Ouvert</i>
Intranet	<i>OUI</i>	<i>Fermé</i>
Extranet	<i>OUI</i>	<i>Ouvert</i>

figure II.5.1 – Les différents types de xNet

L'analyse de cette figure peut éventuellement amener la question suivante : existe-t-il un cas de figure où l'on n'a pas de module d'identification, mais que l'on a un accès restreint ? Cette configuration existe, en effet. Il s'agit également d'un Intranet (puisque étant en réseau fermé), où l'on pourra considérer qu'un seul type d'utilisateur (un seul profil) existe : celui d'un visiteur lambda, ayant accès à toutes les informations de ce xNet.

La distinction qui est faite ici est purement théorique : il est parfaitement possible d'avoir un serveur capable d'assumer les trois fonctions de xNet sans qu'il y ait réellement de frontière. Le cas d'Amazon est d'ailleurs assez illustratif de ce principe : la limite entre l'internet et l'extranet d'Amazon est imperceptible.

En effet, sur le site internet d'Amazon, il est possible, pour un visiteur, de naviguer et de remplir un panier virtuel sans qu'il doive s'identifier. Cependant, une couche supplémentaire d'identification est disponible, permettant au visiteur du site de bénéficier de services supplémentaires (conseil d'achat, etc.), et d'accès à son "compte", offrant un récapitulatif des objets achetés, etc.

Il s'agit là typiquement d'un cas où la frontière est concrètement imperceptible.

On peut imaginer de même un site Internet qui offre en plus des informations disponibles à tous les visiteurs, des services à distance pour ses clients (Extranet) après identification, ou encore aux différents collaborateurs de l'entreprise

II.2. Caractéristiques intrinsèques d'un xNet

II.2.1. Module de Navigation

Grâce au modèle HTML et aux liens hypertextes (inventés par Théodore Nelson en 1965), on peut de façon extrêmement simple modéliser n'importe quel xNet sous forme d'un ensemble de "pages" liées entre elles par ces liens.

Trois modèles globaux (et totalement arbitraires) d'agencement de "pages" coexistent dans le monde des xNet.

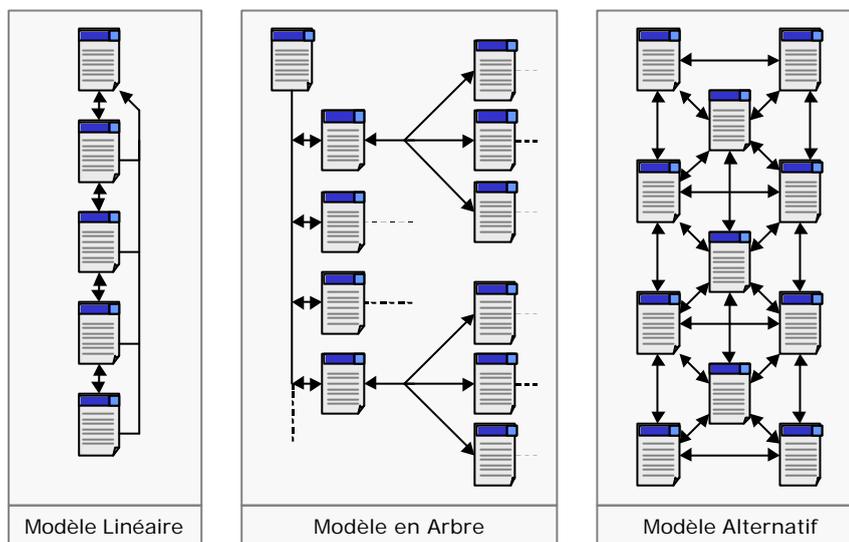


figure II.2.1 – Exemples d'architecture de xNet

Le module de navigation (intimement lié au module d'identification) permet au xNaute¹¹ de naviguer à l'intérieur.

Ce module de navigation peut être situé n'importe où sur la page, se répète le plus souvent sur toutes les pages, et peut définir un ou plusieurs niveaux de l'arborescence. (si l'on est dans le cas d'un site en arbre, ce qui est le cas la plupart du temps)



figure II.2.2 – Le module de Navigation d'Amazon

La figure II.2.2 montre le module de navigation du site Amazon.com. Il s'agit ici d'un site de type "Internet" : Il n'y a pas de droits et de restrictions particulières en fonction du profil. On peut cependant le considérer également comme un Extranet, puisque les informations et le contenu de la page changent sensiblement en fonction de l'utilisateur.

¹¹ Néologisme désignant un intranaute, un internaute, ou un extranaute. (Plus concrètement, utilisateur d'un xNet)

II.2.2. Module d'Identification

Suivant le type de xNet visité, et suivant le contexte dans lequel on se place, tous les utilisateurs n'auront pas accès de la même façon au site... (ces "droits d'accès" ayant une répercussion directe sur le module de navigation)

Pour simplifier : certaines pages pourront être visibles par certaines personnes, tandis que d'autres ne pourront pas y accéder.

Ceci étant parfaitement compréhensible si l'on considère que derrière chaque page visitée, il y a une information (pouvant être à caractère privé), ou un traitement (pouvant être à caractère sensible) qui peut être uniquement dédié à un groupe d'utilisateurs, voire à un seul utilisateur.

Un exemple très parlant pour le cas d'une authentification¹² au niveau de l'utilisateur est celui du site Web d'une banque (donc, d'un Extranet) où chaque client a accès à ses comptes et peut y effectuer des opérations. Il est – bien entendu - totalement impensable qu'une personne ait accès à tous les comptes, ou aux comptes d'autrui ; d'où la nécessité du module d'identification au moment de l'accès aux pages "sensibles".

Un autre exemple, cette fois-ci en Intranet, de l'utilité d'identification d'un groupe de personnes. Dans le cadre de l'application RPI de la Caisse des Dépôts et Consignations (que nous verrons détaillée plus loin), plusieurs groupes d'utilisateurs ont accès aux différentes pages du site.

En l'occurrence, l'application RPI permet d'effectuer un reporting des projets informatiques au sein de la DSI de la CDC. Cette application est disponible sur l'intranet de la CDC et accessible à la fois à la DSI, mais également aux MOE, au MOA, et à la production. Ces trois derniers groupes peuvent remplir les informations relatives à leurs rôles respectifs, et toutes ces données peuvent être consultées et modifiées par la DSI exclusivement.

Dans le cas général, une fois les spécifications fonctionnelles générales achevées, on définit les divers groupes d'utilisateurs, et on leur assigne ou non le droit d'accéder aux fonctionnalités de l'Intranet.

Ceci passe par deux étapes :

Première étape (*figure II.2.3*), on liste toutes les pages qui requièrent une potentielle identification. En fonction de ces pages, on crée une matrice qui liste autant de profils que nécessaire, auxquels on aura associé ou non un droit d'accès à chacune des pages.

La *figure II.2.4* illustre une matrice d'authentification pour un site d'une arborescence à trois niveaux. On applique à cette arborescence un ensemble d'affectation de droits, (en fonction du cahier des charges fonctionnel) d'où découle une liste de huit profils.

¹² Dérivé du mot anglais signifiant "identification". Les deux mots peuvent être utilisés indifféremment.

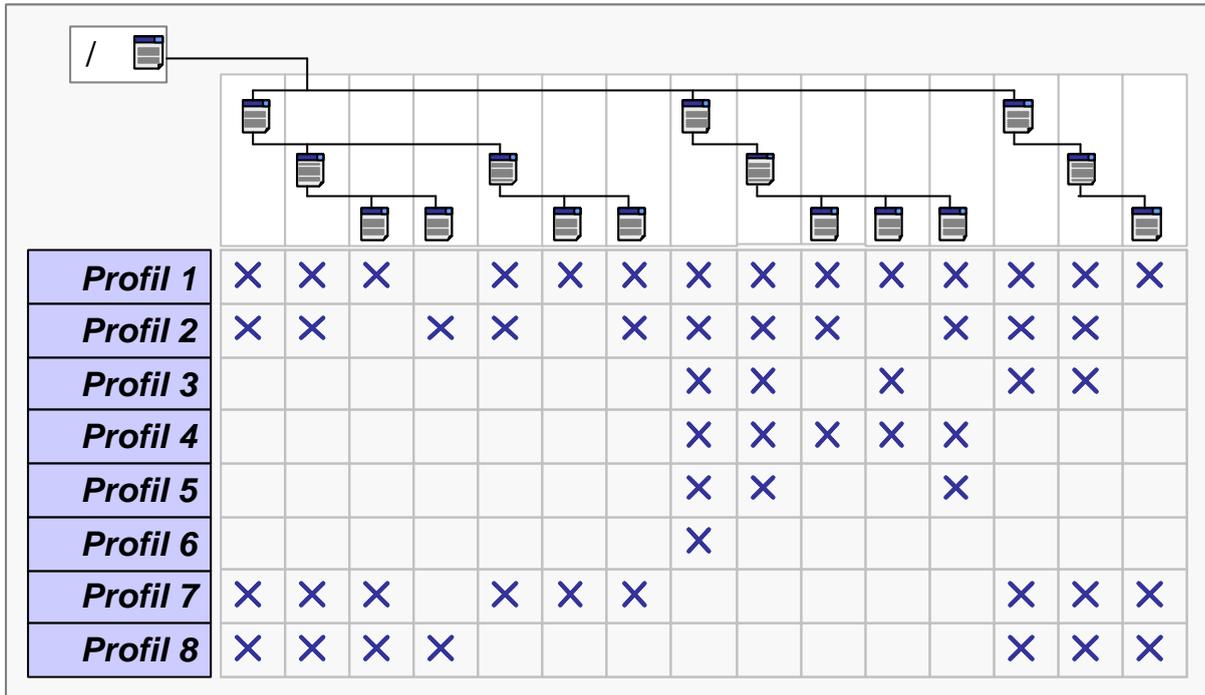


figure II.2.3 – Droits des profils sur les pages

La seconde étape consiste à affecter les profils ainsi définis à chacun des utilisateurs potentiels de l'intranet. Cette affectation passe également par une matrice (figure II.2.4).

	Profil 1	Profil 2	Profil 3	Profil 4	Profil 5	Profil 6	Profil 7	Profil 8
Utilisateur 1			X				X	
Utilisateur 2	X					X		
Utilisateur 3		X	X					X
Utilisateur 4					X			
Utilisateur 5	X	X		X		X		X

figure II.2.4 – Affectation de profils aux utilisateurs

Par la suite, lorsqu'un utilisateur viendra s'identifier sur le site, son "profil utilisateur" sera déduit, et le menu de navigation propre à son profil sera ainsi calculé, puis affiché à l'écran, pour lui permettre de naviguer sur le site. A priori, aucune autre page que celles qui lui sont proposée ne sera disponible, puisqu'elles auront été protégées en prévention de ce cas.

La figure II.2.5 montre la première page du site "The Plastics Exchange" (Place de Marché B2B d'échange de matières plastiques, réservée aux professionnels, sur le web.)

Il s'agit ici typiquement d'un extranet : La première page est une page de présentation de l'entreprise, destinée au grand public, ou aux clients potentiels. On y trouve une zone d'identification, qui va permettre au visiteur de s'identifier vis-à-vis du site.

Une fois cette identification faite, le système connaissant l'utilisateur et son profil, il va pouvoir lui proposer l'accès à un ensemble de pages auxquelles il a un droit d'accès.

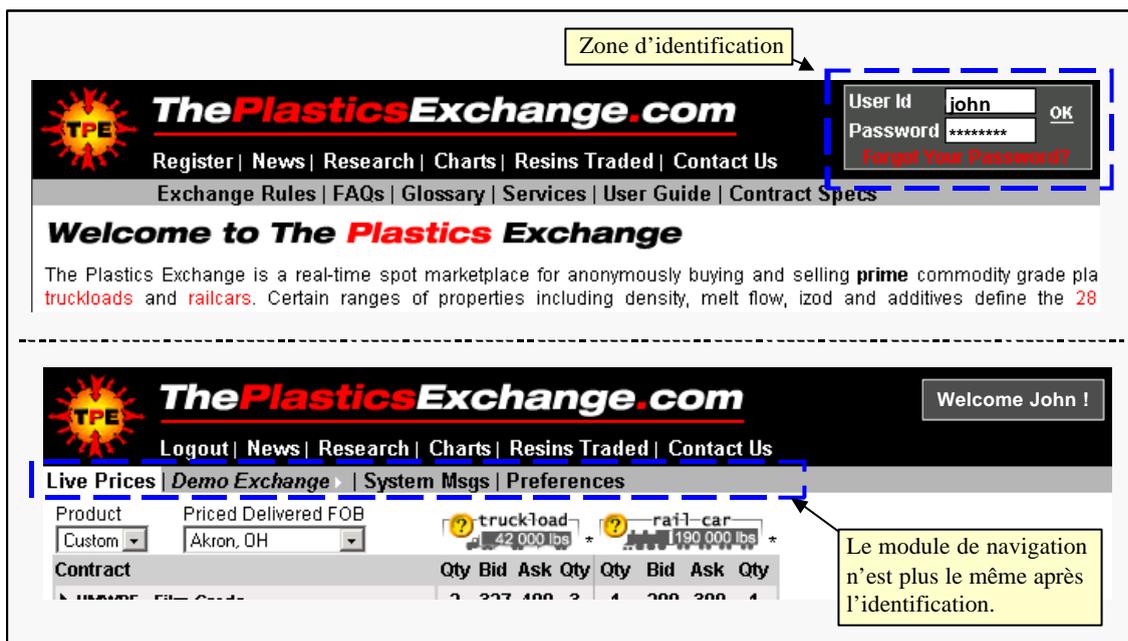


figure II.2.5 – Identification et navigation sur un Extranet

II.2.3. Contenu d'un xNet : Interface, Données, et les Traitements

Il faut garder en tête qu'un xNet n'est pas un simple "site web". Un xNet est une application à part entière : en plus de l'interface utilisateur, chaque xNet possède son traitement et ses données.

La partie "visible" du xNet, l'interface, est en fait un contenu dynamique affiché sur un client... ce contenu est le résultat d'une suite d'opérations effectués sur un ou plusieurs serveurs distant, grâce à des données situées elles aussi, sur un ou plusieurs serveurs distants. La seule différence significative par rapport au Client-Serveur, est que l'interface client est normalisée, et que les transactions liant l'interface aux traitements le sont aussi.

Il est donc tout à fait légitime, dans une certaine mesure, de considérer le xNet comme une application Client-Serveur, et –en terme de charges- de l'évaluer de façon similaire.

Les différentes méthodes que nous allons analyser ne prendront pas en compte le contenu textuel statique du site, qui devra faire l'objet d'une spécification à part, du côté de l'utilisateur.

III. Etat de l'art des méthodes d'évaluations

III.1. Méthode des points de Fonctions

III.1.1. Philosophie des Points de Fonction

La méthode des Points de Fonctions est une méthode destinée à évaluer la taille d'un projet, et ce, indépendamment de la technologie utilisée pour le réaliser.

Son avantage réside principalement dans le fait qu'il est possible d'effectuer une évaluation grâce à cette méthode très en amont dans le projet – en même temps que les spécifications fonctionnelles.

On s'affranchit, de plus, du comptage de lignes de code – comme pour la méthode COCOMO, qui devient d'une fiabilité tout à fait relative en fonction du langage utilisé... Ceci étant d'autant plus vrai dans le domaine des xNet ou une grosse partie des lignes de "code" définit l'interface.

III.1.2. Paramètres de Points de Fonction

Les points de fonctions reposaient, à l'origine, sur un calcul basé sur quatre entités (entrée, sortie, interrogation, fichiers), et ce, sans catégorisation de complexité. (Albrecht, 1979)

Depuis le milieu des années 80, avec l'IFPUG¹³, et la normalisation AFNOR, le comptage des points de fonctions se fait à partir des entités suivantes :

- **Données internes (GDI)**
Relatif à l'aspect statique du Système d'Information.
Groupe de données logiquement liées, ou de groupe de paramètres de contrôle, et identifiables par l'utilisateur. Ces données sont mises à jour et utilisés à l'intérieur de la frontière de l'application.
Notons que certaines entités reliées par une cardinalité (1,1) sont susceptibles de former un seul et même GDI.
- **Données externes (GDE)**
Relatif à l'aspect statique du Système d'Information.
Groupe de données logiquement liées, ou groupe de paramètre de contrôle, et identifiables par l'utilisateur. Ces données sont utilisés par l'application, mais mises à jour par une autre application. Le GDE est un GDI dans un autre domaine.

¹³ IFPUG: *International Function Points User Group*, association fédérant plusieurs grandes entreprises, ayant pour but d'actualiser le guide de comptage par les points de fonctions.

- **Entrées (ENT)**
Relatif à l'aspect dynamique du Système d'Information.
Ce sont les données, ou les paramètres de contrôle, qui entrent dans l'application considérée. Ces entrées maintiennent un ou plusieurs GDI, initialisent ou contrôlent un traitement, et font l'objet d'un traitement unique. Une Entrée correspond donc à un écran de saisie, ou à une réception de données. Il est à noter qu'à chaque GDI doit correspondre au moins une entrée, permettant sa mise à jour.
- **Sorties (SOR)**
Relatif à l'aspect dynamique du Système d'Information.
Ce sont les données, ou les paramètres de contrôle qui sortent de l'application. Ces sorties sont le résultat d'un traitement unique. (Ce traitement unique doit être différent d'une simple extraction de données). Ce peut être un écran de visualisation, ou un message vers une autre application.
- **Interrogations (INT)**
Relatif à l'aspect dynamique du Système d'Information.
Ce sont des données élémentaires qui correspondent à une extraction de données. L'INT ne met à jour aucun GDI.

Ces entités peuvent être définies sur trois acteurs (*figure III.1.1*) :

- L'application
- L'utilisateur
- Les autres Applications

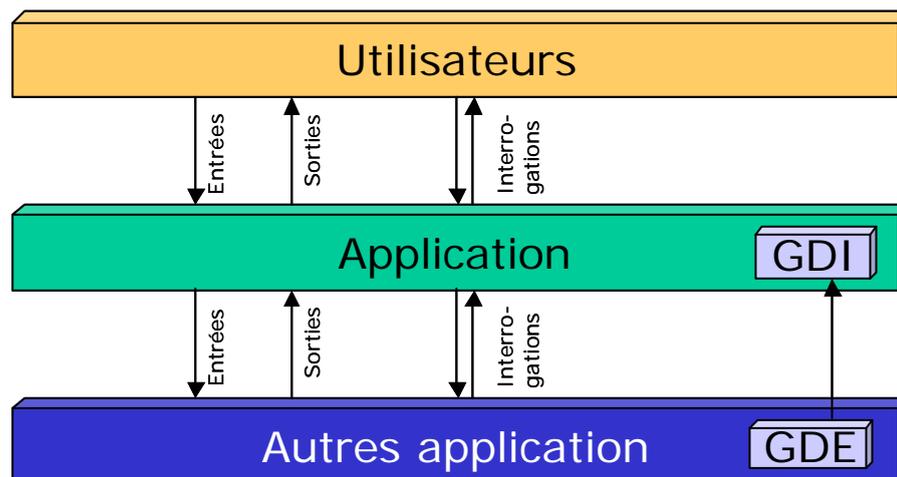


figure III.1.1 – Principe des Points de Fonctions

On associe à ces cinq entités deux paramètres supplémentaires, qui vont permettre de déterminer par la suite le niveau de complexité de chaque élément.

Ces paramètres sont les suivants :

- Données Élémentaires (DE)**
 Chaque GDI ou GDE est composé de données élémentaires. Une DE équivaut à un champ de données. On compte un seul DE par champ répétitif dans les entrées, les sorties, et les interrogations.
- Sous-ensemble logique de données (SLD)**
 Relatif aux GDE et aux GDI.
 D'un point de vue fonctionnel, ce sont les groupements logiques de GDI ou de GDE qui sont traitées simultanément dans l'application.
- Groupe de données référencées (GDR)**
 Relatif aux ENT, SOR ou INT.
 D'un point de vue fonctionnel, ce sont les groupements logiques de GDI ou de GDE qui sont mis à jour, ou consultés simultanément par les différentes ENT, SOR ou INT.

III.1.3. Calcul des Points de Fonction Bruts

Une fois toutes les entités isolés (ENT, SOR, INT, GDI, GDE) et leurs paramètres respectifs calculés (DE, SLD ou GDR), on pourra donc, pour chacun des composants, déterminer son niveau de complexité (*figure III.1.2*).

Points de Fonctions				
Complexité des Composants				
GDI		1 à 19 DE	20 à 50 DE	= 51 DE
	1 SLD	Faible	Faible	Moyen
	2 à 5 SLD	Faible	Moyen	Élevé
GDE	= 6 SLD	Moyen	Élevé	Élevé
ENT		1 à 4 DE	5 à 15 DE	= 16 DE
	0 ou 1 GDR	Faible	Faible	Moyen
	2 GDR	Faible	Moyen	Élevé
	= 3 GDR	Moyen	Élevé	Élevé
SOR		1 à 5 DE	6 à 19 DE	= 20 DE
	0 ou 1 GDR	Faible	Faible	Moyen
	2 à 3 GDR	Faible	Moyen	Élevé
INT	= 4 GDR	Moyen	Élevé	Élevé

figure III.1.2 – Complexité des Composants

Une fois le niveau de complexité de chaque composant déterminé, il suffit de sommer le nombre de chacun des composants, pondéré par une valeur déterminée en fonction de sa complexité.

Points de Fonctions Grille de Complexité			
k(i,j)	Faible	Moyen	Élevé
GDI	7	10	15
GDE	5	7	10
ENT	3	4	6
SOR	4	5	7
INT	3	4	6

figure III.1.3 – Affectation de PdF aux complexités

$$PFB = \sum (C_{i,j} * k_{i,j})^{14}$$

- $C_{i,j}$ correspond au nombre de composants calculés.
- $k_{i,j}$ correspond au coefficient à attribuer à chacun des composants suivant son type et sa complexité. (figure III.1.3)
- PFB correspond au nombre de Points de Fonctions bruts calculés pour l'application.

On peut, pour le calcul, utiliser également le directement le tableau de la figure III.1.4.

Points de Fonctions Valeur des Composants				
GDI		1 à 19 DE	20 à 50 DE	= 51 DE
	1 SLD	7 PdF	7 PdF	10 PdF
	2 à 5 SLD = 6 SLD	7 PdF 10 PdF	10 PdF 15 PdF	15 PdF 15 PdF
GDE		1 à 19 DE	20 à 50 DE	= 51 DE
	1 SLD	5 PdF	5 PdF	7 PdF
	2 à 5 SLD = 6 SLD	5 PdF 7 PdF	7 PdF 10 PdF	10 PdF 10 PdF
ENT		1 à 4 DE	5 à 15 DE	= 16 DE
	0 ou 1 GDR	3 PdF	3 PdF	4 PdF
	2 GDR = 3 GDR	3 PdF 4 PdF	4 PdF 6 PdF	6 PdF 6 PdF
SOR		1 à 5 DE	6 à 19 DE	= 20 DE
	0 ou 1 GDR	4 PdF	4 PdF	5 PdF
	2 à 3 GDR = 4 GDR	4 PdF 5 PdF	5 PdF 7 PdF	7 PdF 7 PdF
INT		1 à 5 DE	6 à 19 DE	= 20 DE
	0 ou 1 GDR	3 PdF	3 PdF	4 PdF
	2 à 3 GDR = 4 GDR	3 PdF 4 PdF	4 PdF 6 PdF	6 PdF 6 PdF

figure III.1.4 – Affectation de PdF aux composants

¹⁴ i décrit l'ensemble {GDI, GDE, ENT, SOR, INT}, j décrit l'ensemble {Faible, Moyen, Elevé}

III.1.4. Calcul des Points de Fonction Ajustés

L'étape suivante (facultative) consiste à ajuster ces points de fonctions. Le principe est le suivant : on corrige le PDF grâce à un ensemble de 14 DI (degrés d'influence).

Chaque DI aura une valeur entre 0 et 5 en fonction de l'importance du degré d'influence sur l'application.

Les degrés d'influence sont les suivants :

- Degré d'influence de la communication des données
- Degré d'influence de la distribution des données ou des traitements
- Degré d'influence de la performance
- Degré d'influence de l'intensité d'utilisation de la configuration matérielle
- Degré d'influence du taux de transition
- Degré d'influence du taux de transaction
- Degré d'influence de la saisie interactive
- Degré d'influence de la convivialité
- Degré d'influence de la mise à jour en temps réel des GDI
- Degré d'influence de la complexité des traitements
- Degré d'influence de la réutilisation
- Degré d'influence de la facilité d'installation
- Degré d'influence de la facilité d'exploitation
- Degré d'influence de la portabilité
- Degré d'influence de la facilité d'adaptation

On calcule ainsi le facteur d'ajustement, en fonction des degrés influence :

$$\text{" FA = 0.65 + S DI}_i / 100 \text{"}$$

Puis on en déduit le nombre ajusté de points de fonctions :

$$\text{" PFA = PFB * FA "}$$

On en déduit que le Facteur d'Ajustement va nous permettre de corriger le nombre de points de fonctions bruts avec un taux allant de 65% à 135% (allant donc du simple au double.)

III.1.5. Calcul de la Charge avec les Points de Fonction

Le calcul de la charge (le point qui nous intéresse en particulier) est une fonction linéaire de PFB. (ou de PFA si on l'a calculé)

Cette évaluation se fait grâce à un facteur multiplicatif.

Ce facteur multiplicatif est d'environ 3 Jours*Homme par Point de Fonction en moyenne (2 J*H si le projet est petit, 4 s'il s'agit d'un grand projet)

A la fin de l'étude détaillée, ce facteur peut varier de 0,1 (pour les projets en L4G, dans le meilleur des cas) jusqu'à 2 pour les projets plus complexes, en passant par un facteur 0,5 pour les projets de type RAD, où la productivité est plus élevée.

III.1.6. Conclusion sur les Points de Fonction

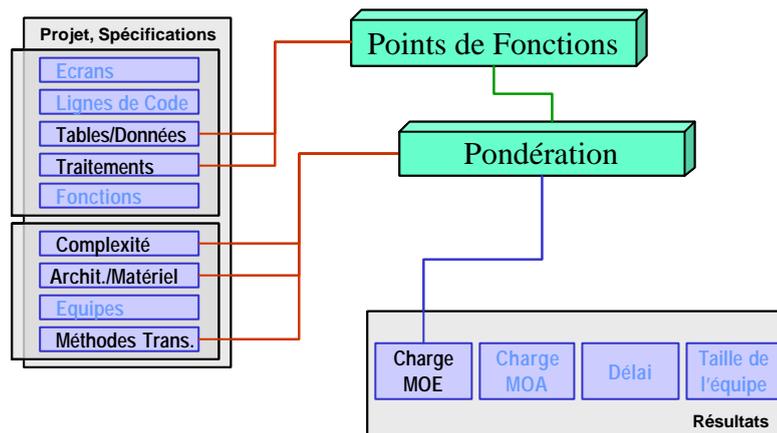


figure III.1.5 – Vue d'ensemble de la méthode

Inconvénients :

- Le comptage est assez subjectif.
- Le comptage est très difficile à automatiser.
- Même si l'estimation de l'effort est juste, l'estimation de la charge (étant sujette à une multiplication par une valeur subjective) est d'une exactitude toute relative.

Avantages :

- L'estimation est disponible assez tôt dans le projet. (Seules des spécifications détaillées sont nécessaires)
- L'estimation est indépendante du langage, de la plateforme, et des diverses technologies utilisées.
- On peut, avec l'expérience et les statistiques, adapter facilement le modèle à l'estimation des xNet.

III.2. Méthode COCOMO

III.2.1. Philosophie de COCOMO

"Pour estimer la charge d'un projet, il faut en estimer la taille"

...Ainsi pourrait être résumée la philosophie de la méthode COCOMO¹⁵.

En effet, la principale hypothèse de cette méthode est qu'un informaticien saura mieux évaluer la "taille" d'un projet informatique que l'"effort" à fournir pour le développer.

COCOMO a été défini par B.W.Boehm¹⁶ sous la forme d'équations simples, admettant quatre paramètres.

Ces divers paramètres (qui sont, en fait, des coefficients) ont été trouvés de façon expérimentale, Boehm se basant sur l'estimation et la réalisation de plus d'une soixantaine de projets divers d'informatique classique. (chaque projet ayant une taille située entre 2000 et 100.000 lignes de code)

La probabilité d'obtenir un résultat proche de la réalité grâce à COCOMO dépend du fait que le projet que l'on analyse soit proche ou non du panel de projets analysés par Boehm.

III.2.2. La "Ligne de Code"

Le premier –et le plus important– de ces quatre paramètres est la "ligne de code". Le nombre de milliers de ligne de code représente la "taille" du projet. C'est principalement en fonction de cette "taille" que l'on estimera le temps nécessaire à la réalisation du projet.

On considère la définition de la ligne de code donnée par N.E.Fenton (*Software Metrics: A Rigorous and Practical Data*, 1998) :

"Une ligne de code est toute ligne du texte d'un programme qui n'est pas une ligne de commentaire, ou une ligne blanche, sans considération du nombre d'instructions ou de fragments d'instructions dans la ligne. Sont incluses toutes les lignes contenant des en-têtes de programmes, des déclarations, et des instructions exécutables et non exécutables."

Cependant, dans le contexte des xNet, contrairement aux langages modernes, orientés GUI¹⁷, on est en droit de se demander si le concept de "nombre de lignes de code" reste fiable, en raison de la nécessité d'utilisation d'un grand nombre d'entre elles pour définir uniquement l'interface de l'application xNet. (C'est typiquement le cas du HTML).

¹⁵ COCOMO : **C**onstructive **C**ost **M**odel (Modèle Constructif De Coût)

¹⁶ Dans l'ouvrage *Software Engineering Economics* (Barry W. Boehm, 1981). ISBN : 0138221227

¹⁷ GUI : *Graphical User Interface*. Il d'agit là de langages de type "Visual", où les lignes de code ne servent pas, ou peu à la définition de l'interface.

La taille d'un projet est estimée en milliers de lignes de code. On parle généralement de KDSI (*Kilo Delivered Source Instruction*), ou de KLOC. (*Kilo Lines Of Code*)

Il est possible, théoriquement, d'évaluer la taille (le nombre de KDSI) en fonction du nombre de points de fonctions du projet, grâce à la formule suivante :

$$\boxed{T = 0.1187 * PFA - 6.49}$$

Une autre méthode¹⁸ d'estimation de T peut être déterminée grâce aux facteurs suivants, en fonction du langage utilisé :

- Assembleur : 0,32 KDSI par Point de Fonction
- C : 0,15 KDSI par Point de Fonction
- COBOL : 0,106 KDSI par Point de Fonction
- Pascal : 0,091 KDSI par Point de Fonction
- Prolog : 0,064 KDSI par Point de Fonction
- Basic : 0,064 KDSI par Point de Fonction
- SQL : 0,040 KDSI par Point de Fonction
- Smalltalk : 0,021 KDSI par Point de Fonction

III.2.3. Equations COCOMO d'estimation d'Effort

Equation pour les projets "simples" : (Projets nécessitant une équipe de trois personnes, ou moins)

$$\boxed{E = 21 * A * T + B}$$

Equation pour les projets "complexes" : (Projets nécessitant une équipe de plus de trois personnes)

$$\boxed{E = 21 * A * T^B * M}^{19}$$

Le modèle COCOMO admet les paramètres suivants :

- E : effort. (mesuré en Jours*Hommes) Il s'agit du résultat de l'équation

¹⁸ Dans l'ouvrage *Function Point Analysis* (B.J. Dreger, 1989). ISBN : 0133323218

¹⁹ Le facteur 21 du début de l'équation est dû au fait que l'équation de Boehm est, à l'origine, destinée à évaluer le projet en Mois*Homme.

- A, B : Coefficients de "Type de logiciel".
- T : Taille, exprimée en KDSI (*Kilo Delivered Source Instruction*, Nombre de milliers de lignes de code source du projet)
- M : Facteur d'ajustement de l'effort

III.2.4. Equation COCOMO d'estimation de Durée

Il est possible d'estimer la durée d'un projet en fonction de l'effort à fournir.

$$\text{" DUREE = 21 * C * (E/21)^D \text{"}^{20}$$

Les paramètres sont les suivants :

- E : effort. (mesuré en Jours*Hommes)
- C, D : Coefficients de "Type de logiciel"
- DUREE : Durée prévue du projet. Il s'agit du résultat de l'équation

On peut, par la suite, calculer le nombre de personnes nécessaires à la réalisation de ce projet :

$$\text{" S = E / DUREE \text{"}$$

Le paramètre est les suivant :

- S: Staffing. (Effectifs) Il s'agit du résultat de l'équation

NB : Les coefficients A, B, C et D diffèrent, non seulement, selon le type de projet, mais également selon le type d'estimation utilisé.

Il existe trois types d'estimation :

- COCOMO Basique
- COCOMO Intermédiaire
- COCOMO Détaillé

Une fois le type de projet décelé, il suffira d'appliquer les précédentes équations. Nous verrons en détails tous les paramètres pour les deux premiers types d'estimation.

²⁰ Il est nécessaire dans cette équation de diviser l'Effort par 21, puisque l'unité d'origine de E dans le modèle COCOMO est le Mois*Homme.

III.2.5. Types de projets

La première chose à faire lorsque l'on décide d'utiliser COCOMO pour évaluer le projet, est donc de déterminer à quel "type" de projet l'on a affaire.

Boehm propose trois types de projet:

Projet Organique : Projet simple, ou de routine, effectué par une équipe ayant déjà travaillé ensemble, dans lequel il y a peu de "surprises" et où une bonne anticipation est possible.

Projet Semi-Détaché : Entre organique et imbriqué. Le Projet n'est ni trop simple, ni trop complexe, l'équipe de développement se connaît un peu, et les technologies peuvent être mal connues, mais pas d'une grande difficulté d'appréhension.

Projet Imbriqué : Techniques innovante, organisation complexe, beaucoup d'interactions. Projet difficile, ou dans un domaine inconnu par l'entreprise, équipe de développement n'ayant pas encore travaillé ensemble, ou projet impliquant des technologies encore peu connues des développeurs.

III.2.6. COCOMO Basique

Le modèle de base de COCOMO est relativement simple. Il permet d'estimer non seulement l'effort (le nombre de Jours*Homme nécessaires à la réalisation d'un projet) en fonction de la Taille (nombre de lignes de code), mais aussi la durée nécessaire, par conséquent le nombre de personnes nécessaires à sa réalisation.

COCOMO Basique	a	b	c	d
Organique	2.4	1.05	2.5	0.38
Semi-Détaché	3.0	1.12	2.5	0.35
Imbriqué	3.6	1.20	2.5	0.32

figure III.2.1 – Coefficients COCOMO Basique

Les coefficients du COCOMO Basique (*figure III.2.1*) suffisent pour avoir une estimation rapide des projets Simples (*figure III.2.2*) et des projets complexes (*figure III.2.3*).

NB : Notons que pour le modèle COCOMO Basique, le facteur d'ajustement **M = 1**.

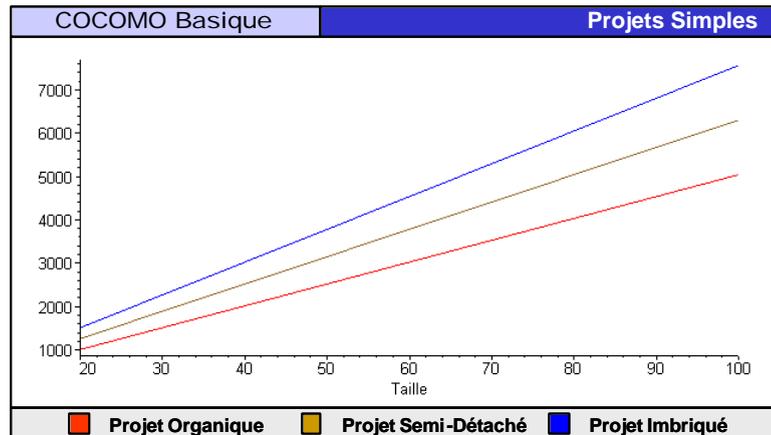


figure III.2.2 – COCOMO Basique, Projets Simples

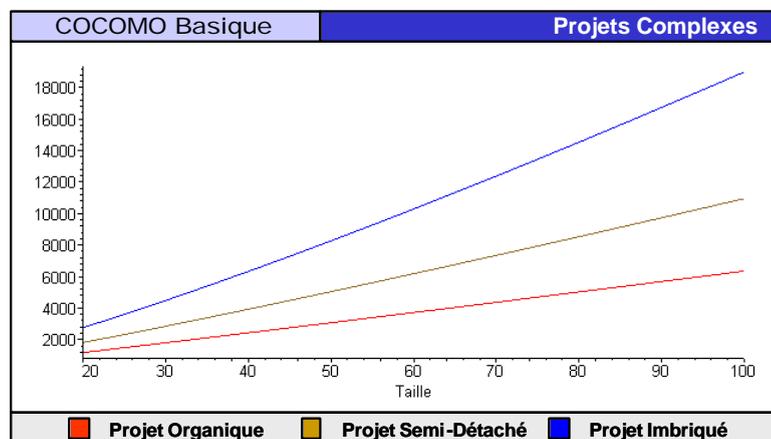


figure III.2.3 – COCOMO Basique, Projets Complexes

Il est même possible de comparer l'estimation pour ces deux types de projet (*figure III.2.4*). On constate cependant une rapide divergence pour les projets qui ne sont pas de type "organique". Il n'est donc pas recommandé d'utiliser l'équation "projets simples" du modèle Basique.

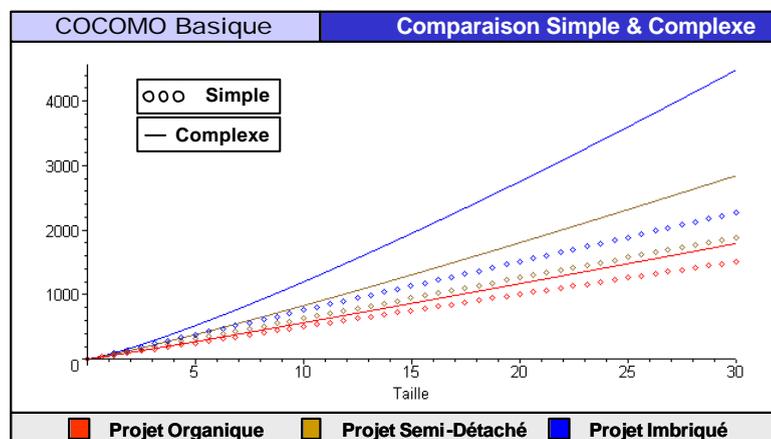


figure III.2.4 – Divergence des Projets Simples et Complexes

III.2.7. COCOMO Intermédiaire

Le Modèle Intermédiaire de COCOMO est relativement similaire au modèle Basique, à la différence près d'un changement des coefficients a et b (*figure III.2.5*), et de la présence d'un "Facteur d'Ajustement" appelé M, dépendant d'une quinzaine de différents paramètres.

La *figure III.2.6* représente les courbes de COCOMO intermédiaire avec un facteur d'ajustement égal à 1.

COCOMO Intermédiaire	a	b	c	d
Organique	3.2	1.05	2.5	0.38
Semi-Détaché	3.0	1.12	2.5	0.35
Imbriqué	2.8	1.20	2.5	0.32

figure III.2.5 – Coefficients COCOMO Intermédiaire

Le facteur d'ajustement **M** est égal au produit des 15 facteurs consignés dans le tableau de la *figure III.2.7*.

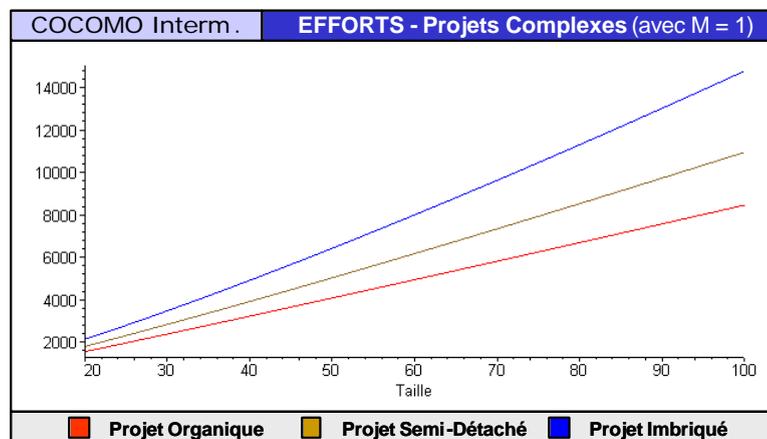


figure III.2.6 – COCOMO Intermédiaire, projets complexes

Les quinze facteurs doivent être classés suivant une échelle à six degrés, allant de "Très Bas" à "Extrêmement haut"

COCOMO Intermédiaire Facteurs d'Ajustement		Très Bas	Bas	Normal	Haut	Très Haut	Extr. Haut
Attributs du produit							
Fiabilité Requise	RELY	0,75	0,88	1,00	1,15	1,40	
Taille de la Base de Données	DATA		0,94	1,00	1,08	1,16	
Complexité du produit	CPLX	0,70	0,85	1,00	1,15	1,30	1,65
Attributs du Matériel							
Contraintes de temps d'exécution	TIME			1,00	1,11	1,30	1,66
Contraintes de taille mémoire principale	STOR			1,00	1,06	1,21	1,56
Instabilité de la Machine Virtuelle	VIRT		0,87	1,00	1,15	1,30	
Temps de Retournement	TURN		0,87	1,00	1,07	1,15	
Attributs de l'équipe							
Compétence des Analystes	ACAP	1,46	1,19	1,00	0,86	0,71	
Expérience du domaine d'application	AEXP	1,29	1,13	1,00	0,91	0,82	
Compétence des Programmeurs	PCAP	1,42	1,17	1,00	0,86	0,70	
Expérience de la Machine Virtuelle	VEXP	1,21	1,10	1,00	0,90		
Expérience du langage	LEXP	1,14	1,07	1,00	0,95		
Méthodes et Outils							
Pratique des Méthodes Modernes	MODP	1,24	1,10	1,00	0,91	0,82	
Utilisation d'outils logiciels	TOOL	1,24	1,10	1,00	0,91	0,83	
Contraintes de planning	SCED	1,23	1,08	1,00	1,04	1,10	

figure III.2.7 – Facteurs d'ajustement COCOMO

Notons que le facteur d'ajustement peut varier de **0.09** à **72.38** (un rapport de plus de 800 !!!), sa valeur étant égale à 1 pour un projet normal.

Les quinze composants du facteur d'ajustement M sont séparés en quatre catégories :

Attributs du produit

- Fiabilité Requise (RELY)
Quelles sont les conséquences de la défaillance du logiciel ?
De "très basse" (Pas de conséquences), à "très haute" (Pertes de vies humaines).
Le "Normale" équivaut à des pertes récupérables.
- Taille de la Base de Données (DATA)
Ratio entre la taille de la base de données et la taille du programme (en octets sur nombre d'instructions). Le "Normale" équivaut à un ratio compris entre 10 et 100.
- Complexité du produit (CPLX)
En terme de structure de contrôle, opérations et calculs, opérations dépendant d'entrées/sorties sur périphériques, gestion de données.

Attributs du Matériel

- Contraintes de temps d'exécution (TIME)
En terme de temps d'exécution machine. (en % en fonction du temps disponible)
- Contraintes de taille mémoire principale (STOR)

En terme de temps de taux d'utilisation de la place. (en % en fonction de la place disponible)

- Instabilité de la Machine Virtuelle (VIRT)
(NB: Machine virtuelle, dans l'appellation de Boehm, correspond à l'environnement de programmation.)
- Temps de Retournement (TURN)
Temps qui s'écoule entre la soumission d'une donnée au système, et la récupération du résultat. Typiquement dans les xNet, ce temps varie de Bas à Normal.

Attributs de l'équipe

- Compétence des Analystes (ACAP)
- Expérience du domaine d'application (AEXP)
Expérience pouvant varier de "Très Bas" (moins de 4 mois) à "Haut" (Plus de six ans)
- Compétence des Programmeurs (PCAP)
- Expérience de la Machine Virtuelle (VEXP)
Expérience pouvant varier de "Très Bas" (moins d'1 mois) à "Haut" (Plus de trois ans)
- Expérience du langage (LEXP)
Expérience pouvant varier de "Très Bas" (moins d'1 mois) à "Haut" (Plus de trois ans)

Méthodes et Outils

- Pratique des Méthodes Modernes (MODP)
Dépend de la méthode utilisée, du degré d'utilisation, et de la connaissance et de l'expérience dans ces méthodes.
- Utilisation d'outils logiciels (TOOL)
Dans l'environnement de programmation, utilisation d'outils aidant au développement/débogage du logiciel, et contribuant à la communication entre les membres de l'équipe
- Contraintes de planning (SCED)
Adéquation entre le temps estimé par l'équipe de réalisation et le temps estimé par COCOMO. "Très Bas" correspond à 75% du temps, "Très Haut" à 160%.

Il est également possible grâce à COCOMO Intermédiaire, de calculer la Durée du projet (*figure III.2.8*), et les Effectifs nécessaires à sa réalisation (*figure III.2.9*).

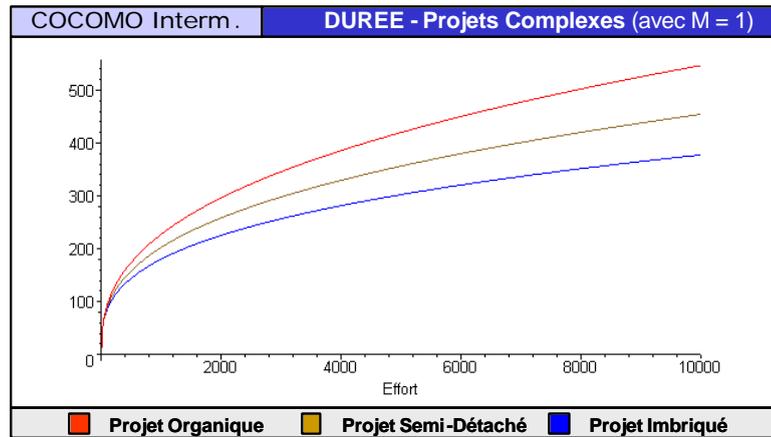


figure III.2.8 – Durée, en fonction de l'effort

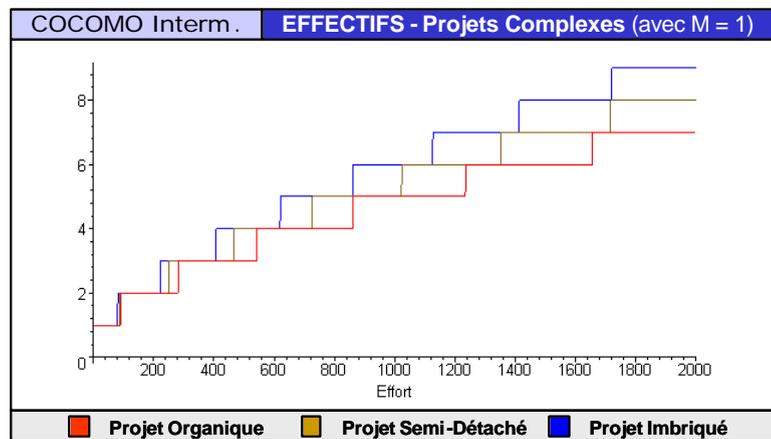


figure III.2.9 – Effectifs, en fonction de l'effort

III.2.8. COCOMO Détaillé

Le dernier niveau de COCOMO (COCOMO Détaillé), fonctionne de la façon suivante : le projet doit être découpé en "phases" distinctes. Chacune des phases doit être évaluée avec COCOMO Intermédiaire (Le facteur d'ajustement M est donc différent pour chaque phase), puis les résultats doivent être sommés à la fin.

III.2.9. Conclusions sur COCOMO

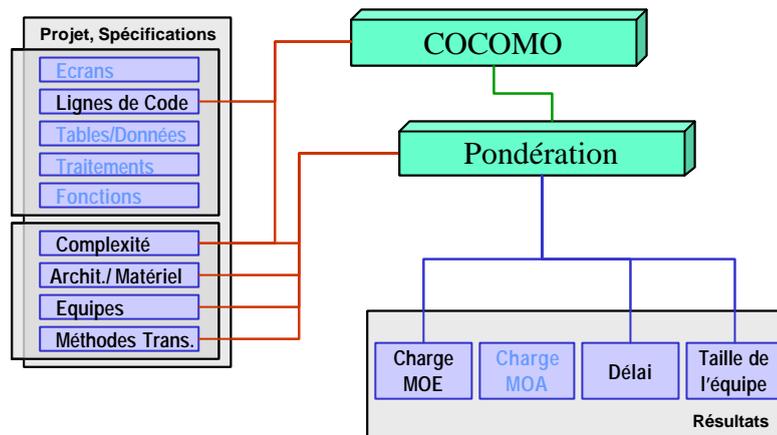


figure III.2.10 – Vue d'ensemble de la méthode

Inconvénients :

- "Pour savoir combien de temps il va falloir pour construire ta maison, compte le nombre de briques dont tu as besoin !"
- Il est assez ardu de calculer de façon fiable la "taille" du projet (en KDSI) lorsque l'on est peu avancé dans le projet. Cette difficulté s'accroît d'autant plus avec l'Effort qu'il conviendrait de fournir.
- Il est peu aisé de choisir de façon correcte le "Type de projet".
- La fiabilité du résultat obtenu dépend énormément du fait que le projet à réaliser se rapproche fortement de ceux qui ont servi à créer le modèle.
- Le fait d'utiliser la "ligne de code" en tant qu'unité de calcul amène quelques failles dans le système : on ne tient pas compte, par exemple, des commentaires dans le code (Qui, pour la maintenance ou le débogage s'avèrent primordiaux).

Avantages :

- COCOMO est transparent, il est facile d'ajuster ses paramètres (d'autant qu'il y en a peu – ce qui peut être aussi bien un avantage, que la mise en valeur d'un « manque ») et de recréer son propre modèle.

III.3. *Evaluateur-RAD (J-P. Vikoff)*

III.3.1. Philosophie d'Évaluateur-RAD

Évaluateur-RAD est un logiciel de Jean-Pierre Vikoff²¹ qui se base, selon l'auteur, sur une évolution des travaux de Albrecht, Ashby, Boehm, Clark, Egyed, Gacek, Hoh, Lehman, Martin, Maxwell, Pareto, Parkinson, Rayleigh et Shannon... autant dire un mélange de al plupart des méthodes d'évaluation et de statistiques du domaines.

Évaluateur-RAD n'est donc pas, à proprement parler, une méthode. Cependant, ce logiciel reste néanmoins très intéressant dans le contexte de cette thèse, pour les raisons suivantes :

- Évaluateur-RAD prend en compte plusieurs paramètres, dont la nature a totalement été occultée dans les autres méthodes analysées ici.
- Il s'agit bel et bien du produit la plus "adapté" aux xNet, les autres méthodes répondant aux demandes d'évaluation, pour la plupart, du client-serveur, ou bien de l'informatique classique.

III.3.2. Paramètres

En guise de présentation, nous allons passer en revue les 12 écrans d'Évaluateur-RAD, et passer en revue les paramètres intervenant dans le calcul de la charge finale.

La métrique Générale

Le premier tableau est le plus significatif : il va servir à mesurer d'entrée le "contenu" de l'application, et son "niveau de complexité". Tout comme la méthode des Points de Fonctions, le résultat obtenu en fin d'évaluation de "contenu" sera altéré par des paramètres d'environnement.

1 - Métrique générale			
Éléments / Complexité	Simple	Moyenne	Complexe
Tables et Relations (définition couche donnée)	0	0	0
Ecrans type NET (D)HTML (couche présentation)	0	0	0
Traitements, règles métier (couche application)	0	0	0
Rapports (conventionnels) et divers extrants	0	0	0
Interfaces, tiers (certif. payeur), annuaire	0	0	0

figure III.3.1 – Métrique Générale

²¹ Fourni dans l'ouvrage "Piloter les projets informatiques de la nouvelle économie (J-P. Vickoff, Editions d'Organisation), ISBN : 2708124870, mais également disponible sur <http://mapage.noos.fr/rad/eval2000.htm>

2 - Application, Organisation

Niveau applicatif (règles de gestion, complexité)	Moyenne
Standardisation exigée (normes)	Moyenne
Recherche de réutilisabilité (objets)	Non applicable
Qualité documentation utilisateur	Réduite
Stabilité des spécifications (ou fiabilité)	Très Stables
Culture organisationnelle	Organisation intermédiaire ou relativement souple
Administration projet (en %)	10

figure III.3.2 – Application, Organisation

3 - Conception (Design)

Niveau de modélisation	Uniquement données en Entité-Relation		
Le modélisateur connaît l'environnement fonctionnel	Un minimum		
Architecture de conception en vue de modification	<input type="checkbox"/> Oui/Non		
La documentation produite par l'AGL est suffisante	<input checked="" type="checkbox"/> Oui/Non		
Réutilisation existant fonctionnel (%)	0		
Résolution application cible	800 * 600	<input type="radio"/> 17"	Correct
Résolution de développement	1024 * 768	<input checked="" type="radio"/> 19"	
		<input type="radio"/> 21+"	

figure III.3.3 – AGL Conception

4 - Méthode, Qualité, Focus

Méthode / efficacité	Toutes les techniques du RAD / CMM niveaux 4 ou 5			
Nbre Jalon Zéro-Défaut FOCUS (à préparer, réaliser, exploiter)	3	?		
Animateur : complète immersion et vérification détaillée	<input type="checkbox"/>			
Groupe d'animation et de communication		Présent	Expert Externe	Jours
Animateur (spécialiste entretien de groupe)	<input type="checkbox"/>	<input checked="" type="checkbox"/>		0
Rapporteur secrétaire (opère en direct)	<input type="checkbox"/>			0
Rapporteur modélisateur (opère en direct)	<input type="checkbox"/>			0
Salle interview RAD opérationnelle en permanence	<input checked="" type="checkbox"/>			0

figure III.3.4 – Méthode, Qualité, Focus

5 - Industrialisation

Pondération expérience précédente (ou contexte organisationnel)

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

Ajustement % (+/-) Premier projet RAD prévoir +15%

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

Progiciel / Composants métier (% couvert) 100 Jours Paramétrage

Réutilisation existant code/objets/données (%)

figure III.3.5 - 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

La modélisation des communications est déjà effectuée

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

Mise en oeuvre d'un workflow procédural

Développement administration workflow

figure III.3.6 – Communication, Workflow, BPR

7 - Équipe et support

Conditions pour Motivation exceptionnelle de l'équipe

Environnement de travail (espace, isolation, décoration) 

Budget actions incitatives visant à la cohésion de l'équipe B

Organisation (horaire, obtention de ressources, etc.)

TOP niveau technologique et qualitatif matériel/logiciel

Vous disposez d'une charte graphique acceptée Oui/Non

D'un modèle générique de transaction / feuilles de style Oui/Non

Intervention spécialiste en documentation utilisateur Oui/Non

Consultation d'un spécialiste en ergonomie cognitive Oui/Non

figure III.3.7 – Equipe et Support

8 - Réalisation / outils

Sophistication interface ? Standard W. ou HTML avec données minimum

Environnement de réalisation VB (VBA, ActiveX) + générateur HTML

L'environnement est déjà opérationnel Oui/Non

Utilisation de VBX/OCX/ACTIVEX performants Oui/Non

Vous lisez VB Programmer Journal Oui/Non

Un des développeurs utilise VC++ Oui/Non

Un des développeurs utilise VBA Oui/Non

figure III.3.8 – Réalisation et Outils

9 - Finalisation

Spécialiste en STRESS d'application événementielle Débutant

Chemins fonctionnels modélisés par le DESIGN Oui/Non

Les utilisateurs disposent d'un correspondant informatique Oui/Non

Complexité de déploiement 10 - 50 Postes

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

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

figure III.3.9 - Finalisation

10 - Éléments techniques

 Équilibre technique équipe 0 Support technique Minimum Soutenu Permanent

Complexité transactionnelle Moyenne

Vous disposez d'un administrateur de SGBD Oui/Non

Les développeurs maîtrisent l'OS Serveur et Client Tous

Client-Serveur de troisième génération (N-Tier) Oui/Non

Présence sous-projet de communications Non

Disponibilité - Sécurité - Performances - Pilotage risques Non

figure III.3.10 – Éléments techniques

11 - Utilisateurs - M.O.

Nombre de personnes impliquées en parallèle	< 5
Nombre de divisions utilisatrices impliquées	1
La prestation se réalise sur le site utilisateur	Entièrement
En terme de recette, la Maîtrise d'Ouvrage est	Excellente
Les utilisateurs sont habitués à l'informatique	<input checked="" type="checkbox"/> Oui/Non
Les utilisateurs sont habitués à Windows	<input checked="" type="checkbox"/> Oui/Non
Disponibilité des utilisateurs	Excellente
Motivation des utilisateurs	Bonne
Engagement utilisateur	
1 à 1 Jours	

figure III.3.11 – Utilisateurs, Maîtrise d'Ouvrage

12 - RESSOURCES

Proposition

	Débutant(s)	Confirmé(s)	Expert(s)
Nbre	0	1	0

Productivité équipe de base par jour en Unités d'Effort 1.6

Ressources nécessaires - optimisées -

Lot(s) ou équipe(s) * Personne(s) (idéalement)

1 * 1

Durée minimum (lot) 7 Jours

Charge à produire 7 Jours

figure III.3.12 - Ressources

III.3.3. Conclusion sur Evalueur-RAD

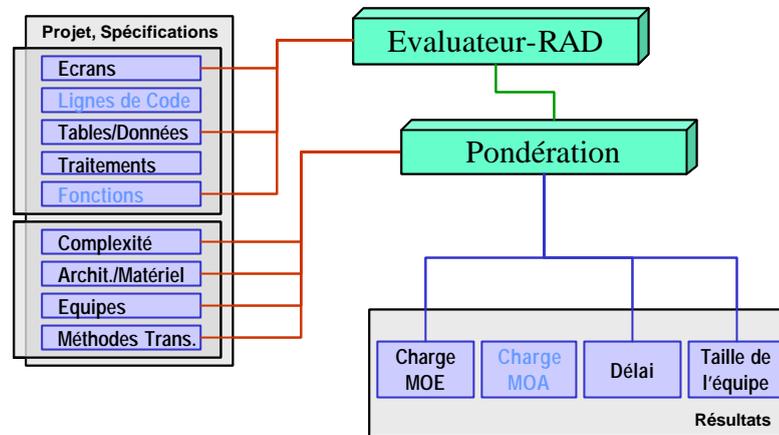


figure III.3.13 – Vue d'ensemble de la méthode

Pour un projet nominal de 100 Jours*Homme, les paramètres d'environnement peuvent varier de 8% à plus de 35000% de la taille prévue. (Où, d'ailleurs, le programme déclare un dépassement de capacité.)

Plusieurs personnes ayant utilisé Evalueur-RAD estiment que la charge jaugée en fin de remplissage des écrans était légèrement supérieure à la réalisation réelle. Nous verrons par la suite si pour l'estimation du projet testé (RPI), la valeur finale était ou non comparable à la réalité.

Avantages :

- Un logiciel complet est fourni pour mettre en application la méthode. Son utilisation est intuitive, et on obtient un résultat rapidement.

Inconvénients :

- La méthode Evalueur-RAD semble donner des chiffres beaucoup trop élevés pour des petits projets.
- Certains coefficients de pondération sont difficilement explicables ou justifiables (ex: taille de l'écran de l'application, etc...)
- La méthode est opaque, et on ne peut pas "régler" les paramètres et coefficients.

III.4. Méthode Interne CDC (A.Probst)

III.4.1. Philosophie de la méthode Interne CDC

Cette méthode a été mise au point par André Probst, Responsable Méthodes de la Direction des Systèmes d'Information de la CDC. La méthode correspond, à l'origine, à l'évaluation de projets de type Client-Serveur.

Les coefficients de cette méthode sont obtenus grâce à l'expérience, sur des essais fait sur plus d'une dizaines de gros projets de la Caisse des Dépôts et Consignations. (Il est d'ailleurs possible, de ce fait, que l'estimation soit mal adaptée à certains projets de trop petite taille.)

Cette méthode est la méthode la plus complète étudiée dans ce document : elle l'avantage d'évaluer à la fois, la charge nécessaire en terme de MOE et de MOA, en rentrant dans tous les détails de la construction de l'application.

III.4.2. Etape 1 : Décomposition fonctionnelle

La première étape consiste à décomposer le projet en macro-fonctions. Chacune de ces macro-fonctions correspond à une fonctionnalité globale de l'application. Par la suite, il s'agira de décomposer chaque macro-fonction en un ensemble de micro-fonctions. (dont le nombre peut être très variable suivant les cas)

Nom du Projet	Estimation nbre de fonctions élémentaires			
	simple	standard	complexe	Total
Fonctions recensées en étude préalable				
Libellé de la première Fonction	5	3	2	10
Libellé de la deuxième Fonction	10	28	11	49
Libellé de la troisième Fonction	20	30	10	60
Libellé de la quatrième Fonction	16	30	20	66
Libellé de la cinquième Fonction	2	4	5	11
etc.				
TOTAL Nbre Fonctions	53	95	48	196

MCD Nbre d'Objets/Relations	58	Nombre d'objets/relations à saisir
Effort Brut	390	
Effort Corrigé	600	

figure III.4.1 – Décomposition Fonctionnelle

Ces micro-fonctions seront pondérées par leur niveau de complexité que leur attribue le chef de projet grâce à sa propre expérience.

Il faut, par la suite, renseigner "Nombre d'Objets/Relations". Sa valeur correspond, dans le MCD, à la somme du nombre de tables "significatives" du modèle de données, et du nombre de relations les liant.

Soient les variables suivantes :

- F_{simples}** : Nombre de fonctions Simples
- F_{standard}** : Nombre de fonctions Standard
- F_{complexes}** : Nombre de fonctions Complexes
- NOR** : Nombre d'Objets/Relations
- EB** : Effort Brut
- EC** : Effort Corrigé

On obtient par le calcul les résultats suivants :

$$\boxed{\text{EB} = \text{F}_{\text{simples}} * 3/5 + \text{F}_{\text{standard}} * 3/2 + \text{F}_{\text{complexes}} * 4 + \text{NOR} * 2/5}$$

$$\boxed{\text{EC} = \text{EB} / 0.65}$$

L'Effort Brut (EB) correspond à une métrique : il s'agit de l'effort fournir dans le cadre de la réalisation d'un projet, à partir d'une étape avancée dans le temps, correspondant approximativement au milieu de l'étude préalable. (La méthode d'origine permettant de calculer les charges en fonction de ces informations était basée sur une durée de projet évaluée à partir de cette période).

L'Effort Corrigé est obtenu par la division de l'Effort Brut par 0.65 (la multiplication par 1.54) pour obtenir l'effort total à fournir sur le projet. Cet effort sera ensuite multiplié par un ensemble de coefficients, afin d'obtenir la charge.

III.4.3. Etape 2 : Répartition des Charges

<i>Nom du Projet</i>											
Ne rien saisir sur cette feuille											
Etapas:	Maitrise d'Œuvre				Maitrise d'Ouvrage			MOE + MOA			
	Coeff	Charge			Coeff	Charge		Coeff	Charge		
		J/H	%MOE	/tot		J/H	%MOA		J/H	%MOE+MOA	% de chacun
									MOE	MOA	
Suivi de Projet		138	14%	8%		60	4%	198	12%	70%	30%
Encadrement Projet	0,23	138			0,1	60		198		70%	30%
Initialisation		8	1%	0%		21	1%	29	2%	28%	72%
Cadrage	0,004	2			0,01	6		8		25%	75%
Protocole de Projet	0,002	1			0,01	6		7		14%	86%
Comité de Pilotage	0,005	3				6		9		33%	67%
Réunion d'initialisation	0,003	2			0,005	3		5		40%	60%
Etude Préalable		130	13%	8%		193	12%	323	20%	40%	60%
Analyse et Modélisation	0,08	48			0,15	90		138		35%	65%
Etude de l'existant	0,03	18			0,02	12		30		60%	40%
Prise en cpte des contraintes	0,03	18			0,05	30		48		38%	63%
Etude de Faisabilité	0,06	36			0,08	48		84		43%	57%
Estimation des charges	0,01	6			0,01	6		12		50%	50%
Revue Qualité		1						1		100%	0%
Comité de Pilotage		3				7		10		30%	70%
*Etude de l' Option Progiel	0	0			0	0		0			
*Comité de Pilotage		0				0		0			
Conception Détaillée		313	31%	19%		229	14%	542	33%	58%	42%
Analyse des Flux	0,1	60			0,1	60		120		50%	50%
Analyse des Données	0,1	60			0,1	60		120		50%	50%
Impacts Organisationnels	0,01	6			0,04	24		30		20%	80%
Maquettage	0,1	60			0,03	18		78		77%	23%
Architecture Applicative/Technique	0,04	24				0		24		100%	0%
Spécifications Fonctionnelles Détaillées	0,05	30			0,03	18		48		63%	38%
Retours revue d'ergonomie		6				3		9		67%	33%
Spécifications Techniques	0,1	60			0,02	12		72		83%	17%
* Reprise d'antériorité	0	0			0	0		0			
Organisation pour l'implantation					0,04	24		24		0%	100%
Estimation des Charges	0,005	3			0,005	3		6		50%	50%
Revue Qualité		1				0		1		100%	0%
Comité de Pilotage		3				7		10		30%	70%
Construction		367	37%	23%		24	1%	391	24%	94%	6%
Préparation des Tests (jeux d'essais)	0,04	24			0,04	24		48		50%	50%
Codage des composants	0,34	204						204		100%	0%
Tests unitaires	0,09	54						54		100%	0%
Tests fonctionnels	0,08	48						48		100%	0%
Qualification/Intégration.	0,06	36						36		100%	0%
Revue Qualité		1						1		100%	0%
Recette		44	4%	3%		98	6%	142	9%	31%	69%
Préparation logistique	0,01	6			0,03	18		24		25%	75%
Tests de Recette	0,02	12			0,1	60		72		17%	83%
Préparation de l'exploitation	0,04	24			0,03	18		42		57%	43%
Revue Qualité/ Comité de Pilotage		2				2		4		50%	50%
Charge Totale Projet		1000	100%	62%		625	38%	1625	100%	62%	38%

figure III.4.2 – Répartition des Charges

Nous aurons par la suite besoin d'une variable de ce tableau :

- SUIVI : % de Charge de MOE assigné au "Suivi de projet". (ici: 14%)

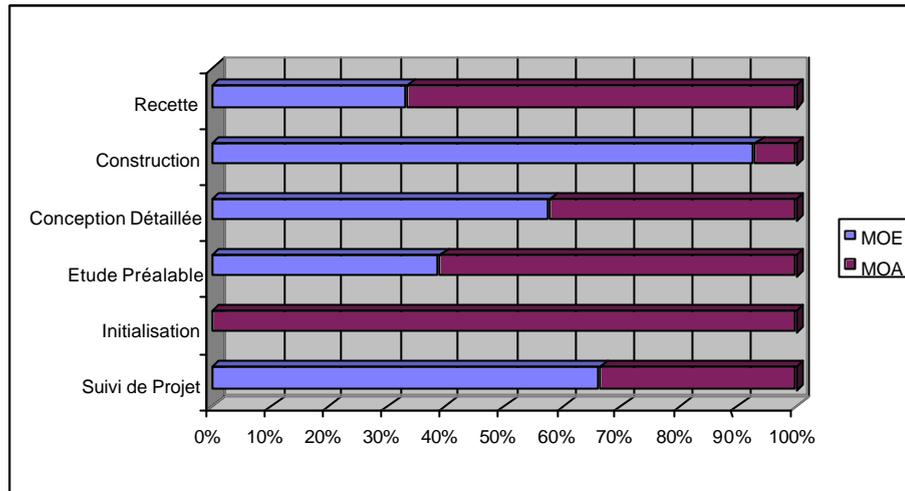


figure III.4.3 – Répartition des charges, EC < 100 JH

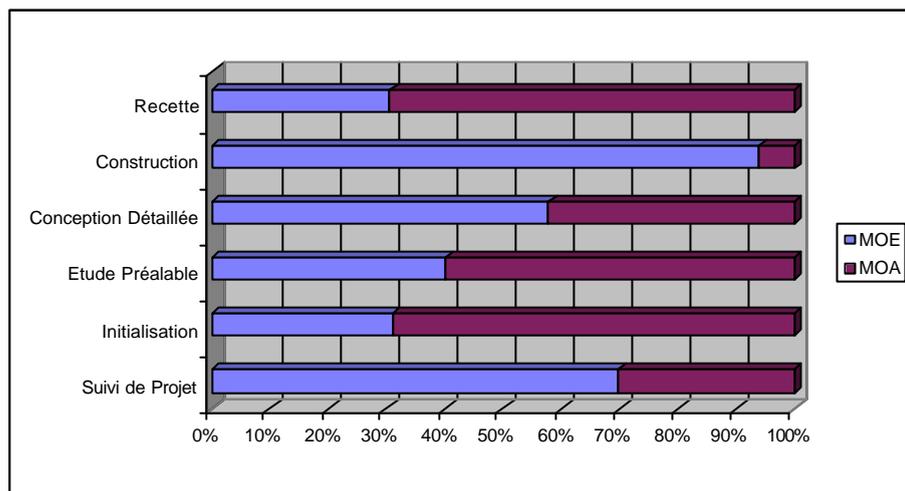


figure III.4.4 – Répartition des charges, EC > 100 JH

La répartition des charges du projet se fait pour la MOE et la MOA en fonction de diverses pondérations de la charge corrigée, suivant des coefficients affectés à certaines tâches du projet.

Il est important de noter qu'une barrière de Charge Corrigée de 100 jours a été fixée : dans le cas où cette valeur serait dépassée, les résultats escomptés diffèrent.

Les différents coefficients sont résumés dans les deux tableaux suivants.

Etape	Coefficient MOE	Coefficient MOA
Suivi de Projet		
Encadrement Projet	0,23	0,1
Initialisation		
Cadrage	0,004	0,01
Protocole de Projet	0,002	0,01
Comité de Pilotage	0,005	
Réunion d'initialisation	0,003	0,005
Etude Préalable		
Analyse et Modélisation	0,08	0,15
Etude de l'existant	0,03	0,02
Prise en cpte des contraintes	0,03	0,05
Etude de Faisabilité	0,06	0,08
Estimation des charges	0,01	0,01
Conception Détaillée		
Analyse des Flux	0,1	0,1
Analyse des Données	0,1	0,1
Impacts Organisationnels	0,01	0,04
Maquettage	0,1	0,03
Architecture Applicative/Technique	0,04	
Spécifications Fonctionnelles Détaillées	0,05	0,03
Spécifications Techniques	0,1	0,02
* Reprise d'antériorité	0	0
Organisation pour l'implantation		0,04
Estimation des Charges	0,005	0,005
Construction		
Préparation des Tests (jeux d'essais)	0,04	0,04
Codage des composants	0,34	
Tests unitaires	0,09	
Tests fonctionnels	0,08	
Qualification/Intégration	0,06	
Recette		
Préparation logistique	0,01	0,03
Tests de Recette	0,02	0,1
Préparation de l'exploitation	0,04	0,03

figure III.4.5 – Coefficients de Répartition

Etape	Charge Corrigée < 100		Charge Corrigée >= 100	
	Charge MOE	Charge MOA	Charge MOE	Charge MOA
Suivi de Projet				
Encadrement Projet	EC * 0,115	EC * 0,05	EC * 0,23	EC * 0,1
Initialisation				
Comité de Pilotage	0	0	EC * 0,005	6
Réunion d'initialisation	0	0	EC * 0,003	EC * 0,005
Etude Préalable				
Estimation des charges	0	0	EC * 0,01	EC * 0,01
Revue Qualité	0		1	
Comité de Pilotage	0	0	3	7
Conception Détaillée				
Estimation des Charges	0	0	EC * 0,005	EC * 0,005
Comité de Pilotage	0		3	7
Construction				
Qualification/Intégration	EC * 0,03		EC * 0,06	
Recette				
Revue Qualité/ Comité de Pilotage	1	1	2	2

figure III.4.6 – Coefficients en fonction de l'Effort Corrigé

Il faut garder en tête que les étapes et les coefficients calculés ici sont adaptés au Client-Serveur.

Pour pouvoir être en adéquation avec les xNet, A.Probst propose d'éventuellement diminuer la charge du projet de 20% par rapport à sa valeur d'origine (pour avoir un résultat rapide), ou mieux, ajouter des étapes, et modifier les différents coefficients des deux figures précédentes.

Nous verrons par la suite quel type d'étapes nous pourrions modifier pour arriver un résultat proche de celui escompté.

III.4.4. Etape 3 : Pondérations des charges

Parmi toutes les charges calculées précédemment, celles de certaines tâches sont directement liées au niveau d'expertise de l'équipe de développement.

1 Pondération du niveau d'expertise de l'équipe de développement					
Seules les 3 cellules bleues sont à renseigner		(le total % doit faire 100)			
		Senior (coeff.0,6)	Moyen (coeff.1)	Novice (coeff.1,4)	Total %
Exemple: Pour 1 expert, 5 bons et 4 novices, entrez : (senior=10.moyen=50.novice=40)		10	50	40	100
Tâches révisées	Charges initiales	Senior	Moyen	Novice	Pondérées
Maquettage	60	3,6	30	33,6	67
Architecture Applicative/Technique	24	1,44	12	13,44	27
Spécifications Techniques	60	3,6	30	33,6	67
Codage des composants	204	12,24	102	114,24	228
Tests unitaires	54	3,24	27	30,24	60
Total en jours	402	24	201	225	450

figure III.4.7 – Pondération des Charges

On compte parmi ces tâche le Maquettage, l'Architecture Applicative, les Spécifications Techniques, le Codage des Composants, et les test.

On applique à chacune de ces charges un coefficient de pondération en fonction du niveau d'expertise du développeur.

Ces coefficients sont respectivement de 60%, 100% et 140% du temps nominal pour des programmeurs experts, moyens, et débutants.

Les applications du monde du Client-Serveur peuvent, par ailleurs, nécessiter une intégration plus ou moins complexe.

En fonction de la complexité de cette intégration, on va affecter au chiffre donné dans la figure III.4.2 de Qualification/Intégration, un facteur de 1, 1.5 ou 2.5 suivant que cette intégration est d'une difficulté normale, complexe, ou très complexe.

2 Pondération selon la complexité à intégrer le Produit				
(architecture technique et logicielle complexe, forte intégration avec d'autres applications)				
Sélectionnez le niveau de difficulté de l'intégration du produit : →			Normal	coefficient
				1
Tâche révisée	Charges initiales	Pondérées		
Qualification/Intégration	36	36		

figure III.4.8 – Pondération due à la complexité d'intégration

Par la suite, on va estimer un besoin "transverse". Ce surcoût est donc lié à des structures transverses, s'impliquant principalement dans le projet lors de réunions, pour des besoins de suivi de projet en terme d'architecture, de qualité, et de sécurité.

Ces charges se calculent par tranche de 100 jours, en fonction des coefficients suivants :

3 Estimation de la charge Architecture/Qualité en J/H (calcul automatique)				
Nature	Barème J/H			Charges calculées
	100 premiers jours	Par tranche de 100 j supplém.	Fixe si > 100 jours	
Administration				
administration fonctions	0,5	0,25		2,75
administration données	0,5	0,5		5,00
PVCS				
validation archivage	0,5			0,50
Revue Qualité				
code NCL	0,5	0,5		5,00
code C	0,5	0,5		5,00
ergonomique	0,5	0,5		5,00
procédures stockées	0,5	0,5		5,00
architecture	2	0,5		6,50
performances	3	0,5		7,50
fin d'étape	3		7,5	7,50
TOTAL Charges				49,75

figure III.4.9 – Charge Architecture/Qualité

Notons qu'en xNet, ce besoin est beaucoup moins important qu'en Client-Serveur.

III.4.5. Etape 4 : Total après Pondérations

Il suffit par la suite de remplacer les valeurs trouvées grâce aux pondérations dans le tableau de répartition des charges.

Nom du Projet									
Ne rien saisir sur cette feuille									
Etapas:	Maîtrise d'Oeuvre (MOE)			Maîtrise d'Ouvrage (MOA)		MOE + MOA			
	Charge	% MOE	/tot	Charge	% MOA	Charge	% MOE+MOA	% de chacun	
	J/H	/MOE	/tot	J/H	/tot	J/H	/tot	MOE	MOA
Suivi de Projet	138	13%	8%	60	3%	198	11%	70%	30%
Encadrement Projet	138			60					
Initialisation	8	1%	0%	21	1%	29	2%	28%	72%
Cadrage	2			6					
Protocole de Projet	1			6					
Comité de Pilotage	3			6					
Réunion d'initialisation	2			3					
Etude Préalable	130	12%	8%	193	11%	323	19%	40%	60%
Analyse et Modélisation	48			90					
Etude de l'existant	18			12					
Prise en cpte des contraintes	18			30					
Etude de Faisabilité	36			48					
Estimation des charges	6			6					
Revue Qualité	1			0					
Comité de Pilotage	3			7					
* Etude de l' Option Progiciel	0			0					
* Comité de Pilotage	0			0					
Conception Détaillée	330	30%	19%	229	13%	559	32%	59%	41%
Analyse des Flux	60			60					
Analyse des Données	60			60					
Impacts Organisationnels	6			24					
Maquettage	67			18					
Architecture Applicative/Technique	27			0					
Spécifications Fonctionnelles Détaillées	30			18					
Retour revue d'ergonomie	6			3					
Spécifications Techniques	67			12					
* Reprise d'antériorité	0			0					
Organisation pour l'implantation	0			24					
Estimation des Charges	3			3					
Revue Qualité	1			0					
Comité de Pilotage	3			7					
Construction	398	36%	23%	24	1%	422	24%	94%	6%
Préparation des Tests (jeux d'essais)	24			24					
Codage des composants	228			0					
Tests unitaires	60			0					
Tests fonctionnels	48			0					
Qualification/Intégration	36			0					
Revue Qualité	1			0					
Recette	44	4%	3%	98	6%	142	8%	31%	69%
Préparation logistique	6			18					
Tests de Recette	12			60					
Préparation de l'exploitation	24			18					
Revue Qualité/ Comité de Pilotage	2			2					
Charges Architecture	50	5%	3%			50	3%	100%	
TOTAL PROJET	1098	100%	64%	625	36%	1723	100%	64%	36%

figure III.4.10 – Total après pondérations

III.4.6. Etape 5 : Reste à Faire

Le "Reste à Faire" calculé en fin de méthode est calculé ainsi :

Le suivi de projet MOE sera égal à la somme des charges MOE de conception, de construction, et de recette, pondérées par la variable SUIVI (cf plus haut).

Le suivi de projet MOA sera égal à la somme des charges MOA de conception, de construction, de recette, et de suivi de projet MOE, le tout pondéré par la variable SUIVI.

Nom du Projet		
Ne rien saisir sur cette feuille		
Reste à Faire		
Etapes	J/H MOE *	J/H MOA *
Conception Détaillée	330	229
Construction	398	24
Recette	44	98
Charges Architecture	50	
Suivi de Projet	107	47
TOTAL PROJET	929	398
TOTAL PROJET (MOE+MOA)		1327

figure III.4.11 – Reste à faire

Le "Total Projet MOE" revient à une estimation MOE en terme de charge de projet (ce qui nous intéresse ici).

III.4.7. Conclusions

Cette méthode, même si réservée au départ au Client-Serveur exclusivement, possède l'avantage d'être d'une transparence inégalée. La plupart des paramètres correspondants réellement au Client-Serveur peuvent être modifiés, voire complètement adaptés. Nous verrons par la suite s'il est possible d'ajouter et d'enlever plusieurs paramètres dans la méthode afin de coller au maximum à l'évaluation des xNet.

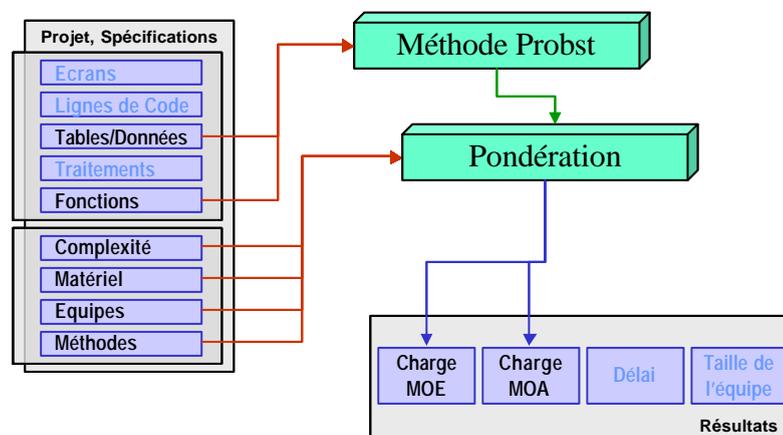


figure III.4.12 – Vue d'ensemble de la méthode

III.5. Méthode interne ICDC (ESTIMAT)

III.5.1. Philosophie d'ESTIMAT

ESTIMAT est un outil mis au point au début des années 1990 par la cellule MIP (Méthode Ingénierie Proiciels, à laquelle appartenait également A.Probst) d'Informatique CDC. Cette méthode se base principalement sur la méthode Algorithmique : le principe consiste à utiliser des formules mathématiques pour exprimer l'effort en fonction d'éléments mesurés, ou estimés.

Par la suite, on leur applique un coefficient d'ajustement se basant sur l'environnement. (Le principe de ces coefficients est relativement similaire au modèle COCOMO). Ce sont des données issues de l'expérience qui vont servir à déterminer ces coefficients.

III.5.2. Estimation de l'Effort Brut de Réalisation

L'effort est la mesure de la charge humaine nécessaire à l'étude et à la réalisation du projet. Cet "effort" va permettre d'estimer la charge et le délai.

L'Effort Brut de Réalisation est un effort calculé à partir d'éléments dénombrés ou estimés, et de données statistiques.

Le but de l'opération va être d'obtenir une Estimation Théorique à la suite d'une étude préalable. Le chiffre obtenu aura une valeur et une précision toute relative, mais aidera déjà à avoir un aperçu de la situation.

Il faudra, bien entendu, par la suite, après l'étude détaillée, faire un chiffrage plus précis et justifié du projet.

La première étape consiste donc à faire une estimation en fin d'étude préalable :

ESTIMAT - Effort de Phase			
	Facile	Moyen	Difficile
Valeur (JH)	20	40	60

figure III.5.1 – Effort de Phase

On commence par calculer le nombre de phase à l'intérieur du projet, et on en estime l'effort en fonction du niveau de difficulté et du tableau des efforts nécessaires à chacune des phases (figure III.5.1).

L'Effort Brut de Réalisation sera donc de :

$$\text{EBR} = n_{\text{facile}} * 20 + n_{\text{moyen}} * 40 + n_{\text{difficile}} * 60$$

- n_{facile} est ne nombre de phases dites "faciles".
- n_{moyen} est ne nombre de phases dites "moyennes".
- $n_{\text{difficile}}$ est ne nombre de phases dites "difficiles".

A cet EBR, il est nécessaire d'ajouter 14% d'"Effort d'Encadrement".

$$\text{" Effort Théorique = EBR * 1.14 "}$$

III.5.2. Estimation de l'Effort de Réalisation et Répartition

Une fois la phase d'étude détaillée amorcée et bien avancée, il sera possible d'effectuer un chiffrage plus pertinent.

Pour cela, on va calculer un effort tenant compte, à la fois de la complexité des règles de gestion et des règles de mise à jour de la base, mais aussi du volume des travaux à effectuer.

On distingue ainsi deux types de traitements :

- **Le traitement Conversationnel** : (ou Transactionnel) - Nombre d'écrans dédiés à l'utilisateur
- **Le traitement par lots** : Nombres d'Etats – Traitement automatisé

A l'intérieur de ces traitements, on va distinguer deux parties :

- **Partie Variable** : Ecrans, ou traitement.
- **Partie Fixe** : Traitement de la partie variable, Processus. (Coût de la séquence entre les parties variables)²²

ESTIMAT				
Estimation Détaillée				
(JH)		Simple	Moyen	Difficile
Traitements Conversationnels	Fixe	5	10	15
	Variable	3	4	5
Traitements par lots	Fixe	4	8	12
	Variable	4	6	8

figure III.5.2 – Charges par type de traitement

On peut donc calculer l'effort de réalisation grâce à la *figure III.5.2*.

²² Les appellations "Variable" et "Fixe" sont trompeuses. On pourrait les remplacer pour plus de compréhension par "Etape" et "Enchaînement d'étapes".

$$E = E_{\text{fixe}} + n_{\text{simple}} * E_{\text{simple}} + n_{\text{moyen}} * E_{\text{moyen}} + n_{\text{complexe}} * E_{\text{difficile}}$$

E_{fixe} : Effort fixe, en fonction de la difficulté de la phase (Partie fixe)

n_{simple} : Nombre d'écrans "simples"

E_{simple} : Effort pour un écran simple (Partie Variable)

n_{moyen} : Nombre d'écrans "moyens"

E_{moyen} : Effort pour un écran moyen (Partie Variable)

n_{complexe} : Nombre d'écrans "difficiles"

$E_{\text{difficile}}$: Effort pour un écran complexe (Partie Variable)

III.5.3. Répartition en Charges (Méthode ESTIMAT)

Une fois l'Effort de réalisation obtenu, on va pouvoir assigner une charge à chacune des phases. Soit un effort nominal E, il faudra affecter comme suit les charges :

Etude Préalable: 10% * E

Etude Détaillée : 60% * E

Etude Technique : 20% * E

Production du Logiciel : K * E

Mise en Œuvre : 40% * E

Soit un total de $(1.3 + K) * E$ d'Effort Total.

$$E_{\text{TOTAL}} = (1.3+K) * E$$

Le coefficient de pondération K sera calculé grâce à 12 facteurs (*figure III.5.3*).

ESTIMAT Facteurs d'Ajustement	Bas	Moyen	Elevé	Très Elevé	Similarité COCOMO
Attributs du produit					
Contraintes de Sécurité et de Qualité	0,85	1,00	1,15	1,40	RELY
Taille de l'Application	0,95	1,00	1,10	1,20	DATA
Complexité des Spécifications	0,85	1,00	1,15	1,30	CPLX
Attributs de l'Ordinateur					
Contraintes d'implantation/d'exécution	0,95	1,00	1,10	1,25	TIME
Instabilité du support Matériel/Logiciel	0,85	1,00	1,15	1,30	VIRT
Attributs de l'équipe					
Compétence de l'équipe	1,20	1,00	0,85	0,70	A(P)CAP
Expérience du domaine	1,15	1,00	0,90	0,80	AEXP
Expérience du support Matériel/Logiciel	1,10	1,00	0,90	0,90	VEXP
Maîtrise du langage de Programmation	1,05	1,00	0,85	0,85	LEXP
Attributs du Projet					
Utilisation de Méthodes & Outils	1,10	1,00	0,90	0,80	MODP
Niveau de Langage utilisé	1,10	1,00	0,90	0,80	N/A
Contraintes de délai de Livraison	0,80	1,00	1,05	1,10	SCED

figure III.5.3 – Facteurs d'ajustement

Ces douze facteurs doivent être classés suivant une échelle à six degrés, allant de "Bas" à "Très Elevé"

Le coefficient de pondération est le produit des 12 facteurs calculés à partir de la *figure III.5.3*.

Notons que ce coefficient peut varier de **0.12** à **7.5** (un rapport de plus de 60... ce qui reste très raisonnable par rapport à COCOMO, mais reste extrêmement élevé dans l'absolu), sa valeur étant égale à 1 pour un projet normal.

III.5.4. Répartition en Charges (Méthode alternative)

Au-delà du modèle ESTIMAT, ICDC utilise un autre type de modèle pour le calcul de la charge et des délais, en fonction de l'effort.

Soit **E**, l'effort à fournir en terme de réalisation et de tests unitaires.

ICDC a déterminé les coefficients suivants :

Spécifications Fonctionnelles Globales (SFG) : 20% * E
 Spécifications Fonctionnelles Détaillées (SFD) : 40% * E
 Etude Technique : 20% * E
 Réalisation : 100% * E
 Tests Unitaires : 25% * E
 Mise en Œuvre : 25% * E
Total : 230% * E

Il convient donc d'appliquer un coefficient de **C = 2.3** à E pour avoir le temps total nécessaire à la réalisation de l'application.

On admet que ce coefficient soit réduit à **C = 2.1** si $E < 80$.

Soit :

$$\boxed{\text{Charge}_{\text{TOTALE}} = C * E}$$

Le Délai sera égal à la racine de la Charge Totale (En Mois*Homme), à laquelle on aura enlevé les SFG et les SFD :

Soit :

$$\boxed{\text{Délai} = 21 * \left(((C-0.6) * E) / 21 \right)^{1/2} \text{ "}}$$

III.5.5 Conclusions

La méthode ESTIMAT, visiblement inspirée d'une métrique de type "Points de Fonction" pour l'estimation nominale de la charge, et de COCOMO pour la pondération, a l'avantage indéniable d'affecter à la charge nominale des coefficients permettant d'estimer la charge totale, des spécification à la mise en production.

La méthode est, à priori, bien adaptée à tous les projets en général, puisque ne dispose pas de paramètres spécifiques au type d'application développé. Il est donc aisé, une fois de plus, d'adapter les coefficients de la méthode pour obtenir une estimation propre aux xNet.

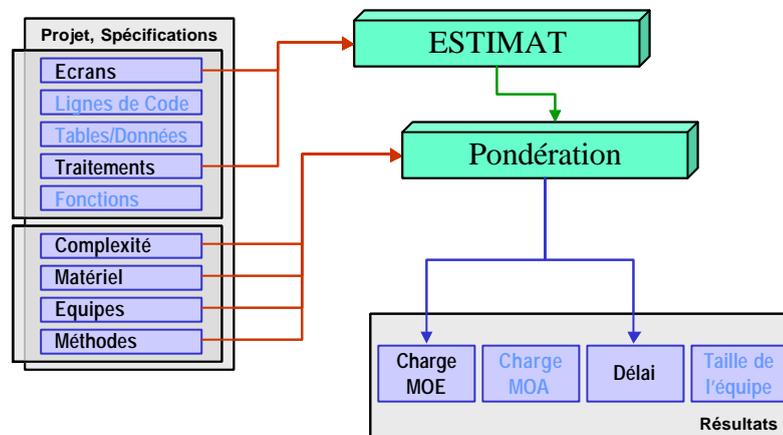


figure III.5.4 – Vue d'ensemble de la méthode

IV. Evaluation d'un xNet

IV.1. Introduction

Les méthodes que nous avons analysées jusqu'à présent (hormis celle de J-P.Vikoff) ne sont pas, à proprement parler, dédiées à l'évaluation des xNet.

C'est d'autant plus cohérent, que ces méthodes – qui nous sont arrivées, et par conséquent ont eu le temps de faire leurs preuves – datent d'avant le milieu des années 1990²³.

Il est relativement légitime de considérer le développement d'un xNet comme celui d'une application en client-serveur (et ce, pour des raisons techniques). Cependant, on notera tout de même quelques différences flagrantes entre ces deux mondes, qui ne sont pas sans avoir un impact important sur un paramètre de développement qui nous intéresse particulièrement ici, à savoir la charge nécessaire au développement.

Tout d'abord, il faut savoir que le client sur le xNet est universel (c'est le navigateur). Le code de l'application "client" s'en trouve de fait considérablement simplifié : grâce à un "métalangage" (comprenez : le HTML, bien évidemment) à la fois simple, et ayant un formidable potentiel en terme d'interface, il n'est plus nécessaire d'avoir un code lourd du côté client... sans compter que la simplicité du HTML permet une capacité de modification du code client sans comparaison.

De même, grâce à la standardisation du CGI²⁴ et du HTTP²⁵, on s'affranchit des protocoles complexes de communication (Sockets, RMI, RPC, CORBA, SOAP, COM, etc.) inhérents au client-serveur... ce qui est un avantage indéniable.

Ensuite, nous avons pu voir dans certains méthodes que les applications de type "RAD" pouvaient être développées beaucoup plus rapidement... pour dire la même chose d'une autre façon, si l'utilisateur peut donner le plus souvent possible un retour sur l'application en cours de développement, les chances de voir le projet dévier seront amoindries (dans une certaine mesure, bien entendu), et le développement devrait en être d'autant plus rapide...

Dans la réalité, ce genre de fonctionnement n'est pas forcément du goût de la maîtrise d'œuvre... en effet, si ce levier est mal utilisé par la maîtrise d'ouvrage, la pression subie par les développeurs, et les changements d'opinions (inévitables) de la part des MOA pourrait également faire perdre du temps de développement.

²³ C'est en effet au milieu des années 1990 que l'Internet a réellement commencé à se démocratiser, (il n'était auparavant destiné qu'aux instituts de recherche et aux universités) et à toucher de plus en plus les entreprises et le grand public, et ce, jusqu'à son paroxysme en 2000, où l'intérêt économique pour les nouvelles technologies a commencé à décroître.

²⁴ CGI: Common Gateway Interface. Il s'agit d'un protocole de "client-serveur" normalisé, utilisé entre une page HTML et un processus de l'application se situant sur le serveur.

²⁵ HTTP: HyperText Transfer Protocol. C'est le protocole sur lequel se base le CGI pour fonctionner. Il sert de support de communication entre un client xNet et un serveur xNet.

Il faut aussi bien voir que si une partie de la charge de la MOE est réduite, du fait d'une vérification assez assidue de la MOA, ce gain de charge est contrebalancé de l'autre côté par une trop grande implication et une perte de temps pour la MOA... ce qui est également coûteux.

Enfin, il existe divers problèmes de "sécurité" dus au fait que le client soit standardisé... ceci pouvant nécessiter une charge supplémentaire, assignée au contrôle du déroulement correct du processus, aux vérifications des authentifications, au comblement des trous de sécurité, etc.

En bref, le type d'application xNet peut aussi bien se trouver être largement plus simplifiée en terme de développement que le client-serveur, ou bien nécessiter à peu près la même charge, voire complètement dépasser les délais prévus, et ce, en fonction principalement, de la complexité de l'application... et des contraintes annexes.

IV.2. Création d'un xNet & ajout de modules

Il faut savoir, par la suite, que le développement d'un xNet connaît plusieurs étapes potentielles. La première, est à proprement parler, celle de la "création", et la seconde, celle de l'"ajout de modules".

IV.2.1. Création d'un xNet

La création d'un xNet est l'étape qui a lieu avant toutes les autres. (les autres étant des "ajouts de modules", si le cas se présente)

Il s'agit en fait, de la toute première réalisation, de la "création" à proprement parler. Lors de cette étape, sont établies toutes les bases de l'application : framework applicatif, module d'authentification, module de navigation, charte graphique, etc. et, bien évidemment, les modules métiers, pour lesquels le xNet a été mis en place.

IV.2.2. Ajout de modules

La seconde étape pourrait être comparée à une simple "maintenance évolutive" : il suffit de greffer à l'existant une application possédant ses propres règles fonctionnelles. Le terme "greffer" est tout à fait adapté : il s'agit en effet de se baser sur les modules d'identification et de navigation existants (et au besoin, les modifier) pour ajouter une application à l'existant. Ce type de "projet" est beaucoup moins spécifique que lors d'une création, et s'apparente, dans ce cas, plus à de l'informatique classique.

IV.2.3 Importance de l'environnement technologique

Il faut savoir que les solutions techniques de créations de xNet sont multiples. Celles-ci peuvent aller de la simple page en HTML sans base de données, à des technologies objets métiers (Beans, EJB, Java...), des procédures stockées en base (PL/SQL...), en passant par

des langages pré-compilés (CGI en général...), ou interprétés côté serveur (Perl, ASP, PHP...).

Bref, la MOE dispose d'un panel de choix technologique on ne peut plus important. Le choix en est d'autant plus difficile qu'un même résultat peut être obtenu en utilisant des technologies tout à fait différentes, parfois de la plus simple à la plus complexe.

Par exemple : une application affichant le calendrier du mois d'Octobre 2002 en HTML peut être développée en 5 minutes pour les technologies les plus simples, ou en ½ journée avec les langages les plus complexes...

En fait, tout dépend :

- du niveau de réutilisabilité de l'application,
- du besoin en maintenance,
- de la performance désirée.

Il va falloir, quel que soit les délais imposés, trouver le meilleur compromis entre le temps de développement, les contraintes de plateforme technologique, le besoin en réutilisabilité et en maintenance, etc.

D'autant qu'il est de plus en plus nécessaire de revenir par la suite sur les applications, pour des besoins en terme d'ergonomie, ou d'évolutivité du produit.

Le temps passé en développement doit être considéré comme un "investissement" sur les maintenances à venir. (Comme dans tous les projets informatiques, d'ailleurs)

IV.3. Alors comment évaluer le xNet ?

La méthode idéale pour évaluer un xNet serait, de façon intuitive, d'adapter un modèle existant (La méthode Probst, par exemple, étant la plus transparente, est très adaptée) en modifiant les coefficients attribués.

Les coefficients suivants :

- Maquettage
- Ergonomie

doivent faire l'objet d'une révision indéniable, puisque l'ergonomie en général est devenu un point-clef de ce type d'application.

Il est également indispensable de faire apparaître les coefficients suivants :

- Module d'Identification
- Module de Navigation
- Production de "contenu"
- Complexité de la plateforme

En revanche, certains coefficients (les transverses, par exemple), seront beaucoup moins importants.

V. Un exemple concret : RPI

V.1. Présentation

V.1.1. Contexte de RPI

La mission d'extension du projet RPI à l'intranet de la Caisse des Dépôts et Consignations a été initialisée dans le cadre du partenariat entre le Mastère Spécialisé HEC/ENSMP MSIT (Management des Systèmes d'Informations et des Technologies) et de la CDC.

La mission est sponsorisée par Monsieur Viteau (Direction de la coordination de la stratégie informatique du groupe CDC), demandée par Monsieur Guy Audoli (Directeur du SI de la Caisse des Dépôts), et supervisée par Monsieur André Probst.

La mission consiste à étendre à l'intranet de la CDC le projet RPI.

L'application RPI met en place des tableaux de bord des projets informatiques de la Caisse des Dépôts. Ces tableaux font apparaître aux décideurs des informations pertinentes sur les projets informatiques qui porteront sur les coûts, les délais et les risques.

L'application RPI est actuellement constituée d'une base de données Access mono-utilisateur, et d'une application développée sous Visual Basic permettant son interrogation, et la création des tableaux de bord.

L'objectif cible consiste à définir et mettre en place un ensemble de modules se greffant sur l'intranet de la Caisse des Dépôts. Ces modules constitueront des interfaces d'alimentation et de consultation de la base de données de l'application RPI.

Il s'agira d'une chaîne de reporting permettant à la MOE, la MOA, et la Production de faire part à la DSI de leurs budgets et de leurs consommations, en introduisant leurs données respectives dans RPI.

La création des modules suivants est envisagée :

Lot 1.1 :

- Module de saisie d'informations de budgets via des formulaires HTML, par l'intranet. Ce module est destiné à remplacer à terme l'alimentation actuelle de la base de données de RPI par la MOA, qui s'effectue à l'heure actuelle par l'intermédiaire de fichiers Excel envoyés sous Outlook.
- Module de mise à disposition d'informations budgétaires contenues dans la base RPI, suivant un modèle de tableau de bord défini par la DSI.

Lot 1.2 :

- Module de saisie d'informations de suivi mensuel de consommations, via des formulaires HTML, par l'intranet.
- Module de publication des tableaux prédéfinis, similaire à l'application existante.

Lot 2 :

- Module de publication des tableaux alternatifs, (par création de nouveaux tableaux), (éventuellement sous BO).

Processus transversal :

- Gestion du changement, fondé sur le nouveau processus de reporting et de tableaux de bord SI.

NB: Seul le lot 1.1 ayant été réalisé et finalisé à la date de rédaction de cette thèse, l'évaluation de RPI grâce aux méthodes proposées portera uniquement sur le lot 1.1 .

V.1.2. Vue d'ensemble du Lot 1.1

Voici un ensemble de captures d'écrans de l'application RPI, donnant un aperçu assez global du contenu de l'application.

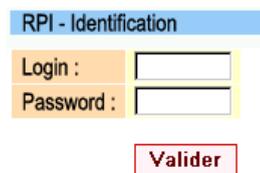


figure V.1.1 : Écran RPI [A]

Le premier écran [A] est un écran d'identification standard. Cet écran passé, l'utilisateur arrive sur la liste des projets [B] qu'il va pouvoir consulter/modifier. Il va pouvoir également ajouter des projets, et obtenir des tableaux de bord.

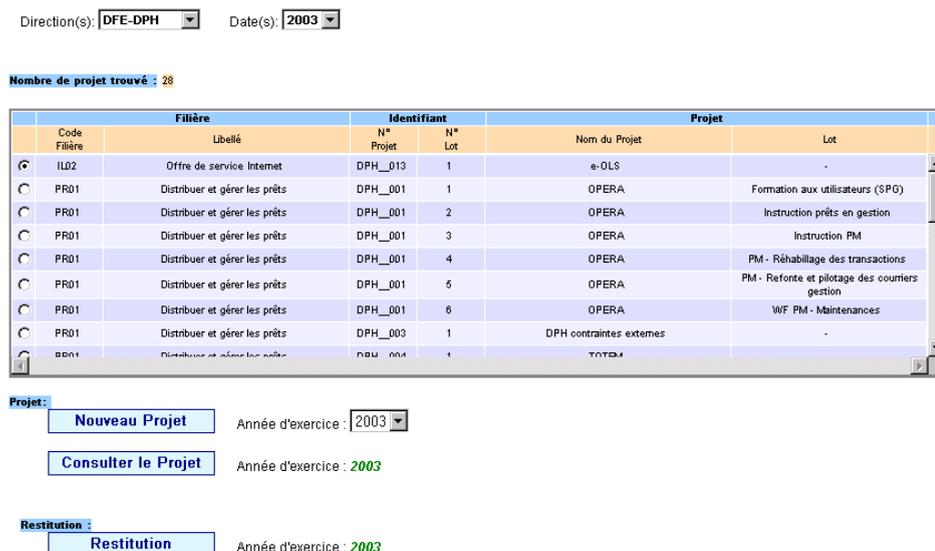


figure V.1.2 : Écran RPI [B]

Fiche d'initialisation de la Procédure Budgétaire 2003

Date de saisie	Emetteur de la fiche	N°e Fiche	N°Projet	N°Lot	Priorité interne
16/09/2002	toto			0	1
Direction/Pôle	Sous-Direction	Filère			
DFE-DPH	DPH	BA01 - Tenue de compte bancaire			

Caractéristiques		Type de projet	
Nom du projet ou de l'application	Responsable Métier	Progiciel	
Nom du lot	Directeur Projet	[Réservé DSI]	Pluriannuel
		0 - normal	Oui
			Date CEPI (j/mm/aaaa)

Dates du Projet (j/mm/aaaa)	
Date de début	
Date de Fin	

Economies envisagées (K€)	
2003	0.0
2004	0.0
2005	0.0

Interdépendances	
	Nom du Lot
Dépendance 1	
Dépendance 2	
Dépendance 3	
Dépendance 4	

ETUDES (Projet)													
Exercice	Budget Exercice (K€)	Charge MOA Interne (J*H)	Charge MOA Externe (J*H)	Coût MOA Externe (K€)	Charge Utilisateur (J*H)	Coût Utilisateur (K€)	Charge MOA ICDC (J*H)	Charge MOA ICDC (K€)	Charge MOE ICDC (J*H)	MOE Bude Externe (J*H)	MOE Bude Externe (K€)	Progiciel (K€)	Autres dépenses (K€)
2002	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Récurrent/ En Cours 2003	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Nouveaux Besoins 2003	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2003	0	0	0	0	0	0	0	0	0	0	0	0	0
2004	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2005	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
TOTAL 2003/2005	0	0	0	0	0	0	0	0	0	0	0	0	0
dont Étude préalable en 2003	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

ETUDES (Maintenance Evolutive)													
Exercice	Budget Exercice (K€)	Charge MOA Interne (J*H)	Charge MOA Externe (J*H)	Coût MOA Externe (K€)	Charge Utilisateur (J*H)	Coût Utilisateur (K€)	Charge MOA ICDC (J*H)	Charge MOA ICDC (K€)	Charge MOE ICDC (J*H)	MOE Bude Externe (J*H)	MOE Bude Externe (K€)	Progiciel (K€)	Autres dépenses (K€)
2002	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Récurrent/ En Cours 2003	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Nouveaux Besoins 2003	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2003	0	0	0	0	0	0	0	0	0	0	0	0	0
2004	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2005	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
TOTAL 2003/2005	0	0	0	0	0	0	0	0	0	0	0	0	0

ETUDES (Maintenance Corrective)													
Exercice	Budget Exercice (K€)	Charge MOA Interne (J*H)	Charge MOA Externe (J*H)	Coût MOA Externe (K€)	Charge Utilisateur (J*H)	Coût Utilisateur (K€)	Charge MOA ICDC (J*H)	Charge MOA ICDC (K€)	Charge MOE ICDC (J*H)	MOE Bude Externe (J*H)	MOE Bude Externe (K€)	Progiciel (K€)	Autres dépenses (K€)
2002	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Récurrent/ En Cours 2003	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Nouveaux Besoins 2003	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2003	0	0	0	0	0	0	0	0	0	0	0	0	0
2004	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2005	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
TOTAL 2003/2005	0	0	0	0	0	0	0	0	0	0	0	0	0

PRODUCTION (K€)					
Exercice	Budget Exercice (K€)	Contrat de service	Equipements Dédies	Exploitation Serveur	Site Central
2002	0.0	0.0	0.0	0.0	0.0
Récurrent/ En Cours 2003	0.0	0.0	0.0	0.0	0.0
Nouveaux Besoins 2003	0.0	0.0	0.0	0.0	0.0
2003	0	0	0	0	0
2004	0.0	0.0	0.0	0.0	0.0
2005	0.0	0.0	0.0	0.0	0.0
TOTAL 2003/2005	0	0	0	0	0

INVESTISSEMENTS 2003 (Projets)	
Besoins Nouveaux (K€)	Observations
0.0	

Annuler Page Suivante >>

figure V.1.3 : Écran RPI [C]

L'écran [C] représente un écran de création ou de modification de projet. (ou en tout cas la première partie de la création/modification, la seconde étant dans un second écran, appelé [D]) On peut considérer l'écran [C] comme complexe, puisqu'il permet de modifier nombre important de données, et qu'il contient des vérifications d'erreurs de saisies relativement pointues, à la fois du côté client et du côté serveur.

Rappel

Date de saisie	Emetteur de la fiche	N°de Fiche	N°Projet	N°Lot	Priorité interne
16/09/2002	toto		123	0	1
Direction/Rôle	Sous-Direction	Filière			
DFE-DPH	DPH	BA01 - Tenue de compte bancaire			

Caractéristiques

Nom du Projet ou de l'application	Responsable Métier	Type de projet			
		Projetiel			
Nom du lot	Directeur Projet	[Réservé DSI]	Pluriannuel	Date CEPI (jj/mm/aaaa)	
		0	OUI		

Fiche descriptive

Rappel Lettre d'objectifs (nb de caractères restant :2048)

Description du projet (nb de caractères restant :2048)

Objectifs et Contraintes

Aspects vitaux		Aspects stratégiques			Niveau de complexité					
Exigences réglementaires, contraintes de place ...	Remplacement d'une application complètement obsolète ou inutilisable	Valeur ajoutée pour le client (économique, image, nouveaux produits)	Valeur ajoutée pour la CDC (enrichissement du savoir-faire et des connaissances)	Cohérence avec l'architecture cible	Contraintes organisationnelles	Maturités des solutions/Maîtrise de la qualité	Evolution de l'existant	Logique d'enchâinement entre applications et solutions intermédiaires	Facilité de la mise en oeuvre	
nul	nul	nul	nul	nul	nul	nul	nul	nul	nul	

Caractéristiques et Risques

Taille du projet	Coût	Détails	Technologie	Organisation	Sécurité	Interdépendances	Conduite du projet	Disponibilité des ressources	Résistance au changement
nul	nul	nul	nul	nul	nul	nul	nul	nul	nul

figure V.1.4 : Écran RPI [D]

La suite de l'écran [C], l'écran [D], est lui relativement moins complexe que l'écran précédent, mais possède lui aussi des vérifications d'erreurs de saisie.

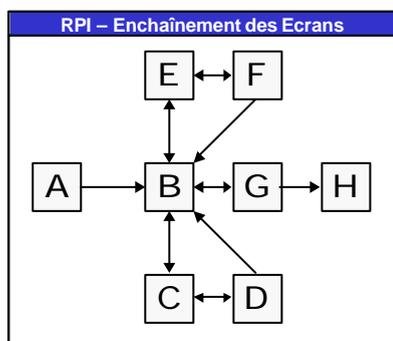


figure V.1.5 : Enchaînement des écrans RPI

Fiche d'initialisation de la Procédure Budgétaire 2003

Date de saisie	16/09/2002	Emetteur de la fiche	M Tran	N°Me Fiche	1204	N°Projet	DPH_001	N°Lot	5	Prorité	0
Direction/Rôle	DFE/DPH	Sous-Direction	DPHO	PR01 - Distribuer et gérer les prêts							

Caractéristiques											
Nom du Projet ou de l'application				Responsable Mètier				Type de projet			
OPERA				M. Damière				Développement			
Nom du lot				Directeur Projet				[Réserve4 DSI]	Pluriannuel	Date CEPI (jj/mm/aaaa)	
PM - Refonte et pilotage des courtiers gestion				T Flao				0	NON		

Dates du Projet (jj/mm/aaaa)		Economies envisagées (M€)		
Date de début		2003		
Date de fin		2004		
		2005		

Interdépendances	
Nom du Projet	Nom du Lot
Dépendance 1	
Dépendance 2	
Dépendance 3	
Dépendance 4	

ETUDES (Projet)														
Exercice	Budget Exercice (M€)	Charge MDA Interne (F/H)	Charge MDA Externe (F/H)	Coût MDA Externe (M€)	Charge Utilisateur (F/H)	Coût Utilisateur (M€)	Charge MDA ICDC (F/H)	Charge MDA ICDC (M€)	Charge MDE ICDC (F/H)	Charge MDE ICDC (M€)	MDE Bude Externe (M€)	MDE Bude Externe (M€)	Projetiel (M€)	Autres dépenses (M€)
2002	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Récurrent/ En Cours 2003	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Nouveaux Besoins 2003	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2003	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2004	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2005	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
TOTAL 2003/2005	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

ETUDES (Maintenance Evolutive)														
Exercice	Budget Exercice (M€)	Charge MDA Interne (F/H)	Charge MDA Externe (F/H)	Coût MDA Externe (M€)	Charge Utilisateur (F/H)	Coût Utilisateur (M€)	Charge MDA ICDC (F/H)	Charge MDA ICDC (M€)	Charge MDE ICDC (F/H)	Charge MDE ICDC (M€)	MDE Bude Externe (M€)	MDE Bude Externe (M€)	Projetiel (M€)	Autres dépenses (M€)
2002	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Récurrent/ En Cours 2003	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Nouveaux Besoins 2003	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2003	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2004	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2005	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
TOTAL 2003/2005	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

ETUDES (Maintenance Corrective)														
Exercice	Budget Exercice (M€)	Charge MDA Interne (F/H)	Charge MDA Externe (F/H)	Coût MDA Externe (M€)	Charge Utilisateur (F/H)	Coût Utilisateur (M€)	Charge MDA ICDC (F/H)	Charge MDA ICDC (M€)	Charge MDE ICDC (F/H)	Charge MDE ICDC (M€)	MDE Bude Externe (M€)	MDE Bude Externe (M€)	Projetiel (M€)	Autres dépenses (M€)
2002	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Récurrent/ En Cours 2003	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Nouveaux Besoins 2003	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2003	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2004	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2005	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
TOTAL 2003/2005	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

PRODUCTION (K€)					
Exercice	Budget Exercice (M€)	Contrats de service	Equipements Dédiés	Exploitation Serveur	Site Central
2002	0.0	0.0	0.0	0.0	0.0
Récurrent/ En Cours 2003	0.0	0.0	0.0	0.0	0.0
Nouveaux Besoins 2003	0.0	0.0	0.0	0.0	0.0
2003	0.0	0.0	0.0	0.0	0.0
2004	0.0	0.0	0.0	0.0	0.0
2005	0.0	0.0	0.0	0.0	0.0
TOTAL 2003/2005	0.0	0.0	0.0	0.0	0.0

INVESTISSEMENTS 2003 (Projets)	
Besoins Nouveaux (M€)	0.0
Observations	

figure V.1.6 : Écran RPI [E]

Rappel

Date de saisie	16/09/2002	Emetteur de la fiche	M Tran	N°Me Fiche	1204	N°Projet	DPH_001	N°Lot	5	Prorité	0
Direction/Rôle	DFE/DPH	Sous-Direction	DPHO	PR01 - Distribuer et gérer les prêts							

Caractéristiques											
Nom du Projet ou de l'application				Responsable Mètier				Type de projet			
OPERA				M. Damière				Développement			
Nom du lot				Directeur Projet				[Réserve4 DSI]	Pluriannuel	Date CEPI (jj/mm/aaaa)	
PM - Refonte et pilotage des courtiers gestion				T Flao				0	NON		

Fiche descriptive

Rappel Lettre d'objectifs

Description de projet

CI. Note

Objectifs et Contraintes		Niveau de complexité			
Aspects vitaux	Aspects stratégiques	Maturité des solutions/Abilité de la qualité		Logique d'articulation entre applications et solutions intermédiaires	
Exigences réglementaires, contraintes de place ...	Remplacement d'une application complètement obsolète ou inutilisable. Valeur ajoutée pour le client (économique, image, nouveaux produits).	Maturité des solutions/Abilité de la qualité		Logique d'articulation entre applications et solutions intermédiaires	
	Maturité des solutions/Abilité de la qualité	Maturité des solutions/Abilité de la qualité		Logique d'articulation entre applications et solutions intermédiaires	
	Maturité des solutions/Abilité de la qualité	Maturité des solutions/Abilité de la qualité		Logique d'articulation entre applications et solutions intermédiaires	

Caractéristiques et Risques							
Taille du projet	Coût	Détails	Technologie	Organisation	Sécurité	Interdépendances	Conduite du projet

Imprimer Modifier Retour Liste Projet << Page Précédente

figure V.1.7 : Écran RPI [F]

Les écrans [E] et [F] sont les pendants des écrans [C] et [D], mais en consultation seulement. Du fait qu'il ne s'agit là que de la restitution d'information, la difficulté de ces deux écrans peut être considéré comme d'un niveau moindre.

On pourra également demander une restitution des informations de RPI contenues en base. L'écran [G] permet de choisir la sous-direction pour laquelle on veut obtenir la restitution.

Choisissez la sous-direction désirée

Direction : **DFE-DPH**

Date : **2003**

Sous-directions consultables : **DPH**

<< Retour menu Consulter

figure V.1.8 : Écran RPI [G]

Cette restitution (Ecran [H]), sera considérée comme complexe, du fait d'un contenu résultant de calculs ardu, à la fois en fonction du profil de l'utilisateur et de la sous-direction pour laquelle il voudra obtenir sa restitution.

Tableau récapitulatif par sous-directions

SSDirection : DPHD		MOE 2003						Forecast MOE Fin Juin 2002		MOA 2003						Forecast MOA Fin Juin 2002		Somme 2003 MOE+MOA		Somme 2002 MOE+MOA			
ETUDES		Récurent			Nouveau			Total		Récurent			Nouveau			Total		Récurent		Nouveau		Total	
Projet	Lot	Type	JH	HE	JH	HE	JH	HE	JH	HE	JH	HE	JH	HE	JH	HE	JH	HE	JH	HE	JH	HE	
ISHTAR	1	CORR	140	104.3	0	0	140	104.3	140	112	126	81.9	0	0	126	81.9	0	0	266	186.2	140	112	
		EVOL	210	156.45	0	0	210	156.45	227	181.8	147	95.55	0	0	147	95.55	0	0	357	252	227	181.8	
		PROJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		Total	350	260.75	0	0	350	260.75	367	293.8	273	177.45	0	0	273	177.45	0	0	623	439.2	367	293.8	
e-OLS	1	CORR	0	0	170	126.65	170	126.65	0	0	0	0	108	70.2	108	70.2	0	0	278	196.85	0	0	
		EVOL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
		PROJ	1000	745	0	0	1000	745	900	720	700	455	30	70	730	525	0	0	1730	1270	900	720	
		Total	1000	745	170	126.65	1170	871.65	900	720	700	455	138	140.2	838	595.2	0	0	2008	1466.95	900	720	
		Total	1350	1005.75	170	126.65	1520	1132.4	1267	1013.6	973	632.45	138	140.2	1111	772.65	0	0	2831	1905.05	1267	1013.6	

Totaux des récapitulatifs par direction

Direction : DFE-DPH		MOE 2003						Forecast MOE Fin Juin 2002		MOA 2003						Forecast MOA Fin Juin 2002		Somme 2003 MOE+MOA		Somme 2002 MOE+MOA			
ETUDES		Récurent			Nouveau			Total		Récurent			Nouveau			Total		Récurent		Nouveau		Total	
Projet	Lot	Type	JH	HE	JH	HE	JH	HE	JH	HE	JH	HE	JH	HE	JH	HE	JH	HE	JH	HE	JH	HE	
		Total	1720	1281.4	370	275.65	2090	1557.05	1068	1574.4	1523	859.95	285	187.2	1611	1047.15	0	0	3701	2804.2	1868	1574.4	
		EVOL	1610	1199.45	450	335.25	2060	1534.7	1593	1274.4	1097	713.05	315	204.75	1412	917.8	0	0	2472	2462.5	1593	1274.4	
		PROJ	2875	2373.25	3900	2004.05	5555	4377.3	4077	3981.6	2810	1912	1753	1085.95	4593	2997.95	105	63	10138	7375.25	5382	4044.8	
		Total	6205	4854.1	3510	2614.95	9715	7469.05	8538	8830.4	4230	3485	2356	1477.9	7896	4962.9	105	63	17301	12431.95	8843	6893.4	

Totaux généraux

Toutes les Directions		MOE 2003						Forecast MOE Fin Juin 2002		MOA 2003						Forecast MOA Fin Juin 2002		Somme 2003 MOE+MOA		Somme 2002 MOE+MOA			
ETUDES		Récurent			Nouveau			Total		Récurent			Nouveau			Total		Récurent		Nouveau		Total	
Projet	Lot	Type	JH	HE	JH	HE	JH	HE	JH	HE	JH	HE	JH	HE	JH	HE	JH	HE	JH	HE	JH	HE	
		Total	2112	1573.44	1112	863.44	3224	2436.88	1680	1584	1553	1009.7	773	685.95	2331	1695.95	0	0	5555	4132.53	1980	1504	
		CORR	2074	1595.88	1520	1229.44	3594	2825.32	1706	1395.36	3150	1081.125	2914	1057.225	6064	21668.35	1112	10013.85	9658	24493.67	2818	11409.22	
		EVOL	4851	3857.0575	6636	8113.0925	11487	13970.15	5706	5873.03	4577.8	3237.885	4065.9	3621.885	8843.7	6859.77	1173.6	786.05	20130.7	20829.92	6879.5	6659.08	
		PROJ	8037	6026.3775	9288	10205.9725	18305	18232.35	9392	8862.39	9280.8	15059.71	7757.9	15185.06	17038.7	30223.77	2285.6	10799.81	35343.7	49466.12	11677.5	19652.3	

figure V.1.9 : Écran RPI [H]

V.2. Estimations théoriques de la charge de RPI

V.2.1. Avec les points de Fonctions

L'analyse des Spécifications Fonctionnelles Détaillées et de la Base de Données de l'application RPI donne les résultats suivants en terme d'estimation de GDI, GDE, ENT, SOR et INT :

RPI - Points de Fonctions Grille de Complexité			
Nombre	Faible	Moyen	Élevé
GDI	5	2	0
GDE	0	0	0
ENT	0	0	2
SOR	0	0	0
INT	0	0	1

figure V.2.1 : Grille de complexité de RPI

Soient :

$$\mathbf{PFB = 5 * 7 + 2 * 10 + 2 * 6 + 1 * 6 = 73 \text{ PdF}}$$

Si l'on considère les ajustements suivants :

RPI - Pts de Fonctions, Pondération	
Degré d'influence :	/5
Communication des données	1
Distribution des données ou des traitements	3
Performance	3
Intensité d'utilisation de la configuration matérielle	1
Taux de transition	2
Taux de transaction	2
Saisie interactive	5
Convivialité	4
Mise à jour en temps réel des GDI	3
Complexité des traitements	2
Réutilisation	3
Facilité d'installation	1
Facilité d'exploitation	4
Portabilité	1
Facilité d'adaptation	3
Total :	38

figure V.2.2 : Grille de Pondération pour RPI

On obtient :

$$\mathbf{FA = 0.65 + 38 / 100 = 1.03}$$

Soit :

$$\mathbf{PFA = 73 * 1.03 = 75}$$

Dans le cas d'un projet "simple", la méthode conseille d'utiliser une estimation 2 Jours*Homme par Points de fonctions.

On obtient donc le résultat suivant :

$$\mathbf{Charge = 2 * 75 = 150 J*H}$$

V.2.2. Avec COCOMO

La taille de l'application RPI (Taille effective, donnée par la MOE après la réalisation) est de **3.798 KDSI**²⁶. (Notons que l'estimation du nombre de ligne de code ne comprend que la partie développée du côté serveur, en JAVA, contenant quelques requêtes SQL.)

Soit en appliquant les équations :

$$\mathbf{T = 0.1187 * 75 - 6.49 = 2.412}$$

On se rapproche de façon étonnante des 2.500 KDSI

Le langage considéré est le Java (qui n'était pas mentionné dans la méthode de Dreger, de 1989).

Si l'on considère les 3.798 KLOC, on obtient 0.050 PdF par KLOC.

Si l'on considère les 2.500 KDSI, on obtient 0.033 PdF par KDSI.

Le code contenant à la fois du SQL et un L3G, le chiffre 0.050 se trouve être tout à fait acceptable.

COCOMO basique, semi-détaché : a=3.0 , b=1.12 :

$$\mathbf{E = 21 * 3.0 * 3.798^{1.12} = 281 J*H}$$

COCOMO basique, semi-détaché : a=3.0 , b=1.12, c=2.5, d=0.35 :

²⁶ Le nombre donné correspond au nombre effectif de lignes du programme. Si on compte le nombre de ";" (points-virgules), on obtient environ 2.5 KDSI, ce qui correspond à un effort nominal de 176 Jours*Homme. En réalité, il serait plus judicieux de parler d'un projet de 2.5 KDSI sur 3.798 KLOC.

Projet RPI - COCOMO		Très Bas	Bas	Normal	Haut	Très Haut	Extr. Haut
Facteurs d'Ajustement							
Attributs du produit							
Fiabilité Requise	RELY	0,75	0,88	1,00	1,15	1,40	
Taille de la Base de Données	DATA		0,94	1,00	1,08	1,16	
Complexité du produit	CPLX	0,70	0,85	1,00	1,15	1,30	1,65
Attributs du Matériel							
Contraintes de temps d'exécution	TIME			1,00	1,11	1,30	1,66
Contraintes de taille mémoire principale	STOR			1,00	1,06	1,21	1,56
Instabilité de la Machine Virtuelle	VIRT		0,87	1,00	1,15	1,30	
Temps de Retournement	TURN		0,87	1,00	1,07	1,15	
Attributs de l'équipe							
Compétence des Analystes	ACAP	1,46	1,19	1,00	0,86	0,71	
Expérience du domaine d'application	AEXP	1,29	1,13	1,00	0,91	0,82	
Compétence des Programmeurs	PCAP	1,42	1,17	1,00	0,86	0,70	
Expérience de la Machine Virtuelle	VEXP	1,21	1,10	1,00	0,90		
Expérience du langage	LEXP	1,14	1,07	1,00	0,95		
Méthodes et Outils							
Pratique des Méthodes Modernes	MODP	1,24	1,10	1,00	0,91	0,82	
Utilisation d'outils logiciels	TOOL	1,24	1,10	1,00	0,91	0,83	
Contraintes de planning	SCED	1,23	1,08	1,00	1,04	1,10	

figure V.2.3 – Facteur d'ajustement pour RPI

M = 1.18 - La valeur de M a été calculée grâce au tableau des facteurs d'ajustement COCOMO (figure V.2.3).

$$E = 21 * 3.0 * 3.798^{1.12} * 1.18 = 331.5 \text{ J*H}$$

$$DUREE = 21 * 2.5 * (331.5/21)^{0.35} = 138 \text{ J}$$

$$S = 331 / 138 = 2.4 \Rightarrow 3 \text{ personnes}$$

On obtient ainsi le résultat :

$$\text{Charge} = 21 * 3.0 * 3.798^{1.12} * 1.18 = \mathbf{331.5 \text{ J*H}}$$

V.2.3. Avec Evaluator-RAD

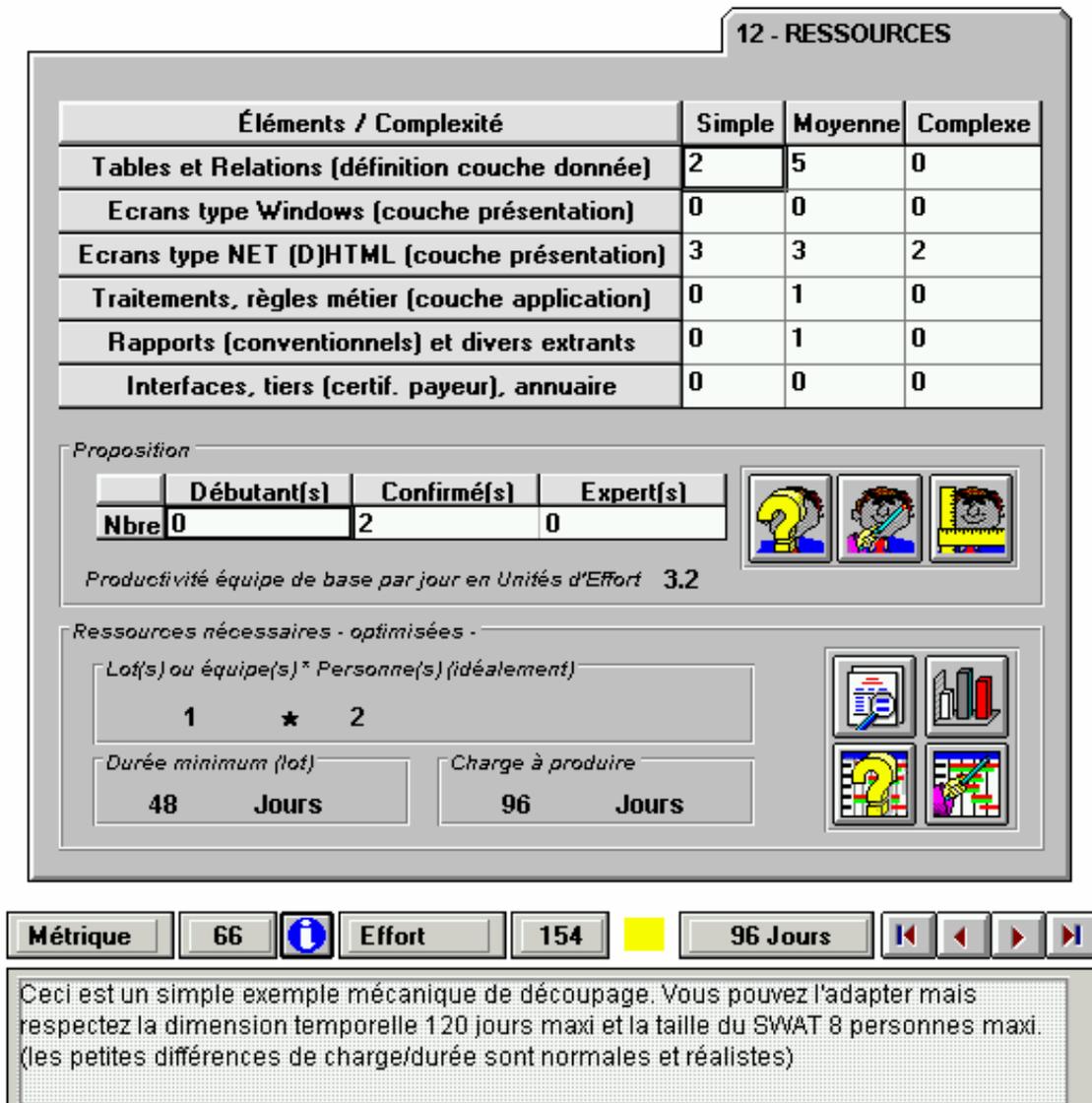


figure V.2.4 – Estimation grâce à Evaluator-RAD

La valeur nominale du projet est de 66 Jour*Homme. Les différents facteurs annexes intervenant amène une charge de 96 J*H. (145% de la taille nominale)

Charge = 96 J*H

V.2.4. Avec Méthode Interne CDC (A.Probst)

RPI - Lot 1.1				
<i>Fonctions recensées en étude préalable</i>	<i>Estimation nbre de fonctions élémentaires</i>			
	<i>simple</i>	<i>standard</i>	<i>complexe</i>	<i>Total</i>
Identification de l'Utilisateur	0	1	0	1
Affichage des Projets	0	1	0	1
Nouveau Projet	0	1	1	2
Consulter le Projet	1	1	0	2
Modifier le Projet	1	1	0	2
Imprimer Détails Projet	1	0	0	1
Restitution Budgétaire	0	1	1	2
TOTAL Nbre Fonctions	3	6	2	11

MCD Nbre d'Objets/Relations	12	Nombre d'objets/relations à saisir
Effort Brut	24	

Effort Brut	24
--------------------	-----------

Effort Corrigé	37
-----------------------	-----------

Reste à Faire		
Etapes	J/H MOE *	J/H MOA *
Conception Détaillée	20	13
Construction	22	1
Recette	3	7
Charges Architecture	12	
Suivi de Projet	3	2
TOTAL PROJET	60	23
TOTAL PROJET (MOE+MOA)		82

figure V.2.5 – Application de la méthode Probst à l'évaluation de RPI

Soit la charge de réalisation MOE :

$$\text{Charge} = \mathbf{60 \text{ J} * \text{H}}$$

V.2.3. Avec ICDC-ESTIMAT

ESTIMAT <i>Estimation Détaillée de RPI</i>					RPI <i>Complexité</i>	
		Simple	Moyen	Difficile	Ecran	Complexité
Traitements	Fixe	0	1	0	A	Simple
Conversations	Variable	3	3	2	B	Moyen
Traitements	Fixe	0	0	0	C	Difficile
par lots	Variable	0	0	0	D	Moyen
					E	Moyen
					F	Simple
					G	Simple
					H	Difficile

figure V.2.6 – Complexité de RPI

RPI - ESTIMAT <i>Facteurs d'Ajustement</i>	Bas	Moyen	Elevé	Très Elevé	Similarité COCOMO
Attributs du produit					
Contraintes de Sécurité et de Qualité	0,85	1,00	1,15	1,40	RELY
Taille de l'Application	0,95	1,00	1,10	1,20	DATA
Complexité des Spécifications	0,85	1,00	1,15	1,30	CPLX
Attributs de l'Ordinateur					
Contraintes d'implantation/d'exécution	0,95	1,00	1,10	1,25	TIME
Instabilité du support Matériel/Logiciel	0,85	1,00	1,15	1,30	VIRT
Attributs de l'équipe					
Compétence de l'équipe	1,20	1,00	0,85	0,70	A(P)CAP
Expérience du domaine	1,15	1,00	0,90	0,80	AEXP
Expérience du support Matériel/Logiciel	1,10	1,00	0,90	0,90	VEXP
Maîtrise du langage de Programmation	1,05	1,00	0,85	0,85	LEXP
Attributs du Projet					
Utilisation de Méthodes & Outils	1,10	1,00	0,90	0,80	MODP
Niveau de Langage utilisé	1,10	1,00	0,90	0,80	N/A
Contraintes de délai de Livraison	0,80	1,00	1,05	1,10	SCED

figure V.2.7 – Facteur d'ajustement pour RPI

$K = 1.03$ - La valeur de K a été calculée grâce au tableau des facteurs d'ajustement ESTIMAT. (figure V.2.3)

$$E = 10 + 3 * 3 + 3 * 4 + 2 * 5 = 41 \text{ J*H}$$

$$E_{\text{TOTAL}} = (1.3 + 1.03) * 41 = 95.5 \text{ J*H}$$

$$\text{Charge}_{\text{TOTALE}} = 2.1 * 41 = 86.1 \text{ J*H}$$

$$\text{Délai} = 21 * \left(\frac{((2.1 - 0.6) * 41)}{21} \right)^{1/2} = 36 \text{ J}$$

Soit la charge de réalisation (Uniquement pour la réalisation) :

$$\text{Charge} = 1.7 * 41 * 1.03 = \mathbf{72 \text{ J*H}}$$

V.3. Déroulement Concret

V.3.1. Estimation de la MOE

Le lot 1.1 de l'application RPI a été estimée à 48 Jours*Homme.

Il faut garder en tête, lorsque l'on donne ce chiffre, que la première partie du projet (en l'occurrence, le lot 1.1) devait absolument être livré pour début septembre 2002, en vue de son utilisation par la DSI pour la procédure budgétaire 2003.

Notons que cette estimation couvre uniquement le développement, les tests, et la recette.

V.3.2. Charge Réelle

Le projet devait donc être opérationnel très vite... Si l'on considère les évaluations faites à travers les différentes méthodes proposées, on se rend compte que toutes les évaluations (sans exceptions) annoncent, sans ambiguïté, un chiffre plus important.

Ce projet a pris très peu de retard... Pourtant, un certain nombre de fonctionnalités non prévues à l'origine ont dû être rajoutées par la suite dans ce lot, sans compter la modélisation de la base, qui a pris un certain temps à se stabiliser. (ce qui n'est pas anormal, dans un contexte de ce type)... On pourra considérer que même si 48 jours ont été prévus, environ 55 ont été utilisés pour la réalisation.

On peut donc aisément conclure que la théorie de Parkinson sur les gaz parfaits s'est sans nul doute appliquée ici.

V.4. Conclusion

Après l'évaluation du projet RPI avec les cinq méthodes proposées, on obtient les valeurs suivantes :

Méthode	Valeur Nominale	Valeur Pondérée
Réel	48	55
Probst	60	60
ESTIMAT	70	72
RAD-Evaluateur	66	96
PDF	146	150
COCOMO	281	331,5

figure V.4.1 – Facteur d'ajustement pour RPI

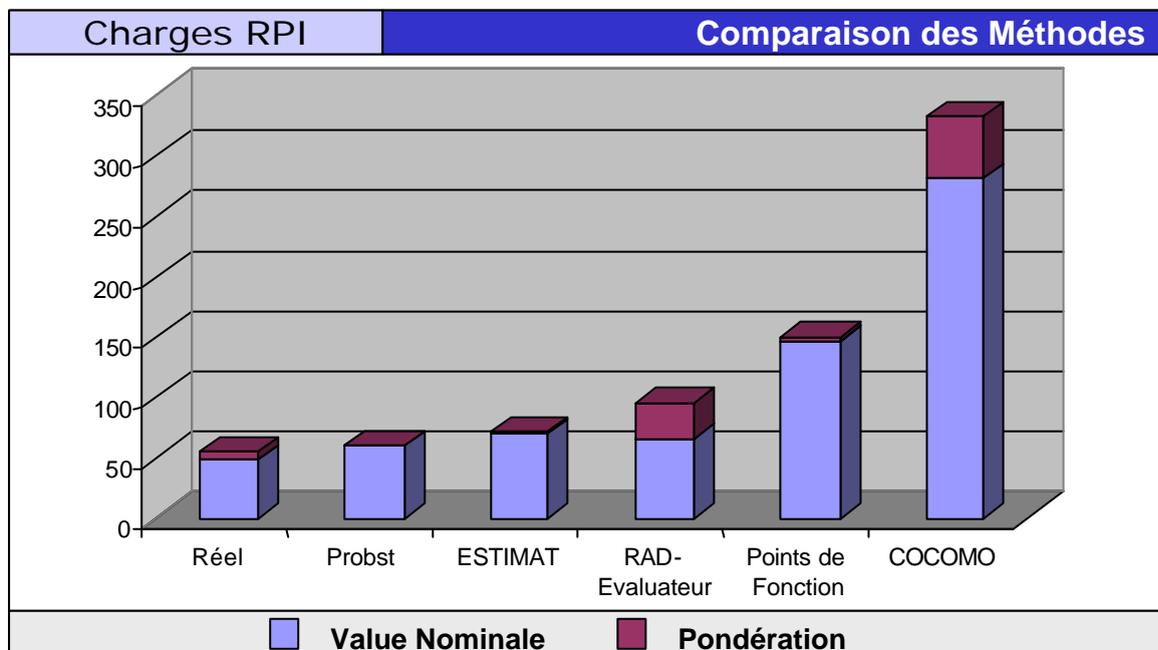


figure V.4.2 – Facteur d'ajustement pour RPI

On peut tirer de ces résultats les conclusions suivantes :

- La méthode COCOMO ne semble pas être adaptée : si l'on considère réellement les KDSI ou les KLOC, en fonction du langage utilisé, pour la même application, on obtiendra des résultats réellement disproportionnés. D'autant que la méthode COCOMO ne tient pas du tout compte de l'application du côté client et de la complexité de l'ergonomie.
- La méthode des points de fonction semble être excellente pour mesurer la "taille" d'une application, en terme de KDSI ou KLOC, mais le facteur de 2 JH par Point de Fonction est beaucoup trop élevé. Il aurait fallu, pour avoir un résultat exact, faire le calcul avec 0.75 JH par Point de Fonction.

- La méthode RAD-Evaluateur donne un résultat nominal (métrique d'origine) très acceptable, mais les pondérations opaques faites par le logiciel font perdre de la crédibilité à la méthode, ce qui est d'autant plus dommage que le logiciel a été développé spécifiquement pour estimer la charge des xNet.
- La méthode ESTIMAT est proche du résultat final. Il faudrait probablement pour être proche de la réalité, y ajouter certaines données relatives aux modules d'identification et de navigation, et modifier les coefficients de complexité des composants.
- C'est la méthode Probst qui se rapproche le plus de la réalité (moins de 10% d'erreur sur ce projet). Comme suggéré lors de l'analyse préalable, il faudrait là aussi, ajouter des données relatives aux modules d'identification et de navigation, et les charges d'opérations transverses, pour obtenir un résultat réellement fiable.

Conclusion

La bonne méthode ?

En résumé, si l'on voulait construire une méthode idéale d'évaluation des charges de projets xNet, on devrait effectuer les étapes suivantes :

- 1) Calcul de la métrique du projet, en terme d'écrans (ESTIMAT, RAD-Evaluateur), ou de Fonctions (Probst, Points de Fonctions).
- 2) Pondération en fonction de paramètres internes (COCOMO, Probst)
- 3) Pondération en fonction de paramètres externe (Toutes les Méthodes)
- 4) Ajout de contraintes transverses (Probst)
- 5) Calcul de charges restant à effectuer en fonction d'un coefficient correspondant à l'état actuel du projet (Probst)

Quelques bonnes pratiques

Au bilan, cinq méthodes plus ou moins différentes les unes des autres. Chacune a ses propres paramètres, chacune donne ses propres résultats.

Et la fourchette des paramètres entrés étant réellement large, on ne peut pas réellement faire ressortir les paramètres de métriques principaux. De plus, les variables de sortie étant elles mêmes peu cloisonnées, on ne sait jamais exactement ce que l'on est en train d'estimer (les tests unitaires et la recette sont-ils compris dans les charges de la réalisation, etc.)

En bref, vouloir faire ressortir un résultat précis et bien délimité, à partir d'une science non exacte prenant en compte toute sorte de paramètres, dont certains sont plus ou moins farfelus, se révèle être, au final plus un exercice de style, qu'une méthode fiable : on jongle avec beaucoup de chose, et il est on ne peut plus aisé de faire rendre à la méthode les chiffres que l'on veut.

Cependant tout n'est pas perdu : le fait de « croiser » toutes ces méthodes peut permettre de donner une estimation plus ou moins réaliste (à un facteur près, puisque toutes les évaluations se révèlent être trop élevées.)

Ainsi, une bonne solution pourrait être, si l'on désire effectuer rapidement une évaluation d'un projet xNet, de créer une sorte de "Tableau de Bord", destiné à la MOA, qui serait également fourni à la MOE, et contenant un ensemble de points sur lesquels ils pourraient se mettre d'accord, en l'occurrence sur les paramètres d'entrée... ceci nécessitant une demi-journée, tout au plus.

La MOA pourrait rapidement avoir un aperçu global des estimations ainsi faites.

Ce tableau de bord pourrait contenir les informations suivantes ::

- Estimation de la MOE.
- Métriques des cinq méthodes (Nombre et complexité d'écrans, de tables, de fonctions, etc.)
- Estimations faites avec les Points de Fonctions, COCOMO, RAD-Evaluateur, Probst, et ESTIMAT (tableaux similaires aux *figures V.4.1* et *V.4.2*)

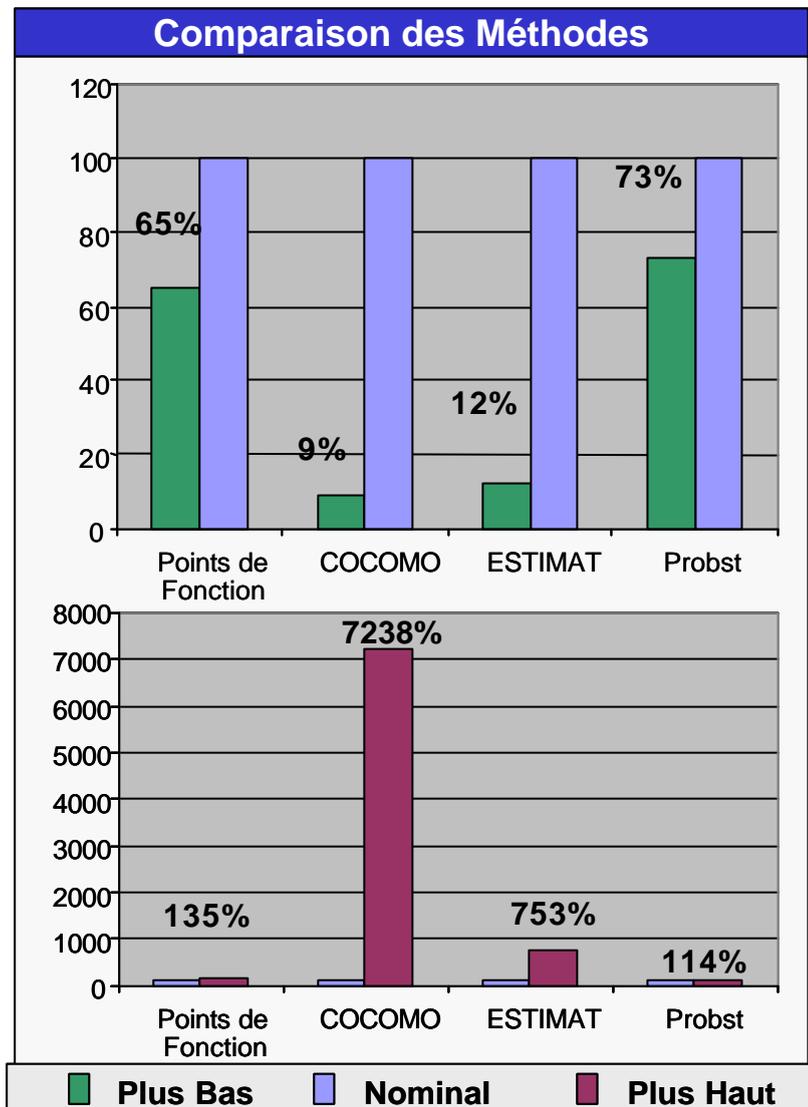


figure VI.1.1 – Extrêmes des pondérations

- Présentation des marges d'inexactitude des pondérations en fonction des méthodes (figure VI.1.1). En effet, ces pondérations font varier la taille nominale du projet de façon inquiétante. La MOA, en voyant ces chiffres, aura probablement le réflexe d'utiliser la méthode où l'erreur due à la pondération est la plus faible.
- Moyenne des estimations de $\frac{1}{6} * (\text{Charge la plus optimiste} + \text{Charge la moins Optimiste} + 4 * \text{Charge la plus probable})$.

... cette solution amènerait la MOA à juger elle-même du chiffre d'estimation des charges qu'elle peut obtenir, tout en lui mettant en avant qu'il est évident que le résultat obtenu sera d'une fiabilité relative, en raison des marges d'erreurs, et de la difficulté à comparer ces méthodes hétéroclites... Le plus important de ce « tableau de bord » étant probablement d'apporter des arguments nécessaires à une discussion saine et constructive, en matière d'élaboration de planning, entre la MOE et la MOA.

Annexes

Bibliographie

Ouvrages

- **La mesure du logiciel** (*Henri Habrias*)
2ème édition revue, corrigée et augmentée
Teknea (ISBN : 2877170454)
- **Piloter les projets informatiques de la nouvelle économie** (*Jean-Pierre Vickoff*)
Editions d'Organisation (ISBN : 2708124870)
- **Project Management: a Managerial Approach**
(*Jack R. Meredith, Samuel J. Mantel Jr.*)
John Wiley and Sons (ISBN: 0471434620)
- **Web Site Project Management** (*Ashley Friedlein*)
Morgan Kaufmann (ISBN: 1558606785)
- **Gestion d'un projet système d'information - Principes, techniques et mise en œuvre**
(*Chantal Morley*)
Dunod (ISBN: 2100059734)
- **Précis de Conduite de Projet Informatique** (*Cyrille Chartier-Kastler*)
Editions d'Organisation (ISBN : 2708118145)
- **Conduite de projets informatiques : principes et techniques**
(*José Moréjon, Jean-René James*)
Interéditions (ISBN : 272960457X)
- **Le management d'un projet** (*H.-P. Maders*)
Editions d'Organisation (ISBN : 2708117947)
- **Que Sais-je : Le management de projet** (*Olivier Badot, Jean-Marie Hazebroucq*)
Presses Universitaires de France (ISBN : 2130474179)
- **Le maître d'ouvrage du système d'information** (*Alain Berdugo*)
Hermès Sciences (ISBN : 2866016211)
- **Guide du Management des Systèmes d'Informations - Thèmes & Termes Essentiels**
(*Alain Berdugo, Robert Mahl, Gérard Jean, HEC/ENSMP MSIT 2002*)
Hermès Sciences Publications (ISBN : 2746205246)
- **De la Stratégie aux Systèmes d'Information** (*ALTIME*)
Formation ALTIME/HEC/ENSMP, février 2002

Documentations Electroniques

- **ADELI (d' Association pour le DÉveloppement de la Logique Informatique)**
<http://www.adeli.com/>
 - **Estimations des charges d'un projet Informatique** (*Alain Coulon*)
<http://www.adeli.com/actcomest.htm>
 - **Estimations de charges - Les orientations de la commission** (*Alain Coulon*)
Lettre ADELI n°39 - Square des Utilisateurs - avril 2000
<http://www.adeli.com/doc/l39p8.pdf>
 - **Estimation de projets informatiques** (*Kathleen Peters, traduit par Alain Coulon*)
Lettre ADELI n°41 - Square des Utilisateurs - octobre 2000
<http://www.adeli.com/doc/l41p23.pdf>
 - **Calibrage et étalonnage** (*Jean Joskowicz*)
Lettre ADELI n°42 - Square des Utilisateurs - janvier 2001
<http://www.adeli.com/doc/l42p20.pdf>
 - **Science ou magie ? - Le point sur les estimations des projets logiciels**
(*Nicolas Trèves et Alain Coulon*)
Lettre ADELI n°45 - Square des Utilisateurs - octobre 2001
<http://www.adeli.com/doc/l45p19.pdf>
- **La méthode des Points de Fonction** (*Olivier Denel*)
Thèse MSIT (HEC/ENSMP) - <http://www.denel.com/pdf/index.html>
- **Rapport CIGREF : Réseaux Internet/Intranet**
Rapport CIGREF, septembre 1997
- **COCOMO: Resource Estimation**
(*David Stotts, Associate Professor, Dept. of Computer Science*)
University of North Carolina - <http://www.cs.unc.edu/~stotts/COMP145/cocomo.html>
- **IFT3902 : Développement, maintenance de logiciels**
(Université de Montréal : Faculté des Arts et des Sciences)
Informatique et recherche opérationnelle
<http://www.iro.umontreal.ca/~pift3902/Cours2001/Planification.pdf>
- **Estimation des coûts et délais par la méthode COCOMO**
(*Tina Wilhelm, Évelyne Parthenay, Hugues Am*)
Ecole Supérieure en Sciences Informatiques
<http://www.essi.fr/~hugues/GL/COCOMO/cocomo.html>
- **Algorithmics Costs Model** (*Dan Snell*)
Bournemouth University - <http://www.ecfc.u-net.com/cost/models.htm>

- **Conduite de projets informatiques - Principes Généraux et Techniques**
(*Violaine Prince*)
Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier
<http://www.lirmm.fr/~joab/tni/cp2.pdf>
- **Evaluation, la théorie** (*Jean-Pierre Vickoff*)
<http://mapage.noos.fr/rad/boudeval.htm>
- **Principes d'Evaluation** (*Jean-Pierre Vickoff*)
<http://mapage.noos.fr/rad/evalint.htm>

Autres documents & Méthodes

- **Méthode Interne CDC – Feuille de Calcul d'Estimation de Projets**
(*André Probst*)
- **Méthode ESTIMAT**
Informatique CDC, Méthodes Ingénierie Progiciels