

# Une tentative d'identification des facteurs critiques du succès de l'introduction d'un Atelier de Génie Logiciel dans une organisation

Christian VARINARD<sup>1</sup> et Robert REIX<sup>2</sup>

<sup>1</sup> Docteur en Sciences de Gestion  
Enseignant contractuel à l'Université Jean-Moulin Lyon III

<sup>2</sup> Professeur à l'Université Montpellier II  
Responsable de la formation doctorale du Centre de Recherche  
en Gestion des Organisations de l'IAE de Montpellier

---

## RÉSUMÉ

---

*Les Ateliers de Génie Logiciel sont des technologies de l'information dont l'utilisation peut avoir un impact positif sur la productivité des équipes de développement et sur la qualité des logiciels réalisés. A partir des résultats d'une étude empirique, cet article montre cependant que cet impact n'est pas systématique et propose quatre facteurs critiques du succès du recours à un AGL pour la réalisation d'applications informatiques : l'implication des acteurs dans le choix de l'outil, la formation sérieuse des utilisateurs à l'outil et à la méthodologie qu'il supporte, le recours à un champion de la technologie et l'apprentissage de l'outil sur des projets pilotes puis la généralisation progressive de son utilisation sur des projets réels.*

**Mots-clés** : AGL, Impact, Productivité, Qualité, Facteurs clés de succès.

## ABSTRACT

---

*CASE tools are information technologies which use can have a positive impact on development productivity and software quality. From the results of an empirical study, this paper shows however that this impact is not systematic and suggests four critical factors of success resorting to a CASE tool for the realisation of business applications : the actors implication in the choice process of the new tool, the rigorous and practical training of the user's concerning the new technology and its related development methodology, the existence of a champion of the new technology and the learning of the tool on pilot projects and its progressive use on real projects.*

**Key-words** : CASE tool, Impact, Quality, Productivity, Critical factors of success.

## INTRODUCTION

Dans un contexte général de concurrence accrue, les responsables des départements Système d'Informations (SI) sont soumis à des exigences croissantes de la part des directions générales. Dans le cadre de budgets limités, ils doivent, sous des contraintes de temps généralement sévères, réaliser des applications rendues plus complexes à la fois par des besoins d'intégration à l'existant et l'élargissement des possibilités des matériels. En outre, ils doivent consacrer une part de plus en plus importante de leurs moyens aux activités de maintenance des logiciels (supérieure à 50 % du temps de travail). Ils sont donc à la recherche en permanence, de méthodes et d'outils propres à améliorer la productivité des équipes de développement et à assurer le niveau de qualité satisfaisant des solutions construites.

Le recours à un Atelier de Génie Logiciel (AGL) est une des solutions les plus récentes pour tenter de faire face à cette "crise du logiciel". On entend par AGL, une collection d'outils et de composants de génie logiciel regroupés autour d'un référentiel central et utilisée par les informaticiens afin d'automatiser tout ou partie du cycle de vie des logiciels (c'est-à-dire les phases d'analyse, de conception, de réalisation, de mise en œuvre voire de maintenance). Si l'on se réfère aux arguments des fournisseurs de ce type d'outils, leur utilisation pour le développement d'applications informatiques débouche tout à la fois sur une amélioration de la qualité des logiciels produits, sur une hausse de la productivité des équipes de développement et sur un renforce-

ment de l'efficacité organisationnelle. En dépit de ces arguments alléchants, le taux de pénétration des AGL demeure assez faible puisqu'on l'estime à environ 20 % des entreprises dans les pays occidentaux tels que les Etats-Unis, le Royaume-Uni ou la France.

Ceci semble indiquer que l'AGL n'est pas perçu par les organisations comme une solution miracle aux nombreux problèmes qu'elles rencontrent dans leurs activités de développement de logiciels. Plusieurs études (Card, Mc Garry et Page, 1987 ; Currid, 1993 ; Selamat *et al.*, 1994 ; Stobart, Thompson et Smith, 1991 ; Yellen, 1990) confirment d'ailleurs cette appréciation en indiquant que les gains espérés en termes de productivité et/ou de qualité sont illusoire. D'autres démontrent que les bénéfices attendus sont fortement réduits par un défaut de formation et d'expérience vis-à-vis de l'outil, par la résistance au changement des développeurs ou par une augmentation du temps consacré aux phases de conception et de test (Norman *et al.*, 1989 ; Orlikowski, 1988 ; Orlikowski, 1989 ; Vessey, Jarvenpaa et Tractinsky, 1992). Cependant, il est possible de recenser des études qui montrent qu'au contraire, l'utilisation d'un Atelier de Génie Logiciel procure les différents bénéfices escomptés (Banker, Kaufman, 1991 ; Necco, Tsai, Hogelson, 1989 ; Norman, Nunamaker, 1988 ; Swanson *et al.*, 1991).

Les résultats contradictoires des études relatives à l'utilisation d'un AGL sont difficiles à interpréter. L'une des hypothèses les plus plausibles est que le recours à une technologie de ce type ne conduit pas nécessairement à une amélioration de la qualité des lo-

giciels produits et à une augmentation de la productivité des équipes de développement, et qu'il existe un certain nombre de facteurs clés du succès de l'introduction et de l'utilisation d'un AGL.

Dans cette perspective, l'objet de notre article consiste tout d'abord à analyser l'impact observable de l'utilisation d'un AGL à la fois sur la qualité des logiciels produits et sur la productivité des développeurs puis ensuite à mettre en évidence l'existence de facteurs clés du succès de l'introduction de cette technologie dans l'organisation. En raison de la complexité du phénomène étudié et des difficultés d'utiliser une démarche expérimentale méthodologiquement satisfaisante, nous avons choisi de recourir à une observation longitudinale dans trois organisations. Cette approche soulève plusieurs questions de caractère méthodologique qui seront abordées dans la première partie de cet article ; dans une seconde partie, après avoir montré le caractère non systématique des impacts de l'utilisation de l'AGL, nous tenterons d'identifier, sur la base des observations, les quatre facteurs critiques du succès vraisemblables de cette technologie.

## **I. CADRE METHODOLOGIQUE DE L'ETUDE LONGITUDINALE**

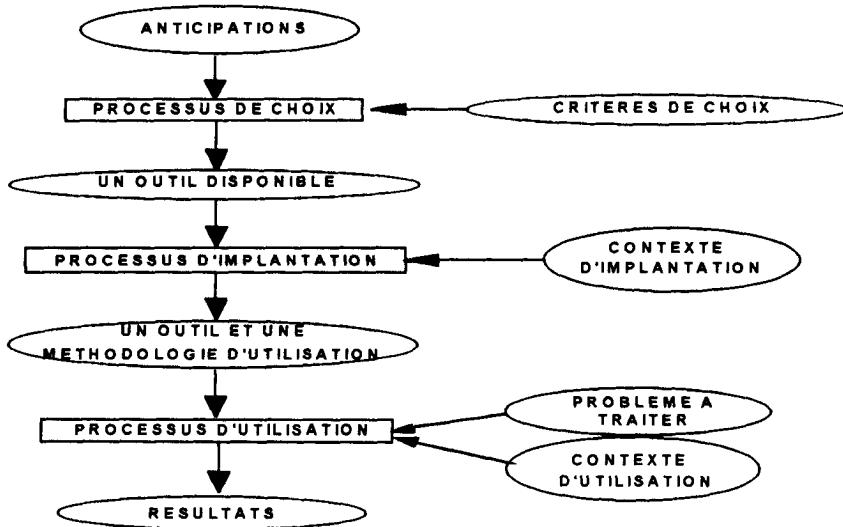
Les études longitudinales peuvent être définies comme des analyses spatiales et temporelles de phénomènes complexes en raison des conditions, des acteurs et des implications qui en découlent (Wacheux, 1996). On considère généralement que des études de ce type sont pleinement adaptées lorsque l'on tente d'analyser un processus, c'est-à-dire un enchaî-

nement d'événements dans le temps, susceptibles d'expliquer un phénomène (ici, l'utilisation réussie d'un AGL). Ces processus sont décrits par référence à un cadre d'analyse présenté dans le premier paragraphe. Ils ont été observés au sein de trois organisations différentes décrites dans le second paragraphe. Enfin, parce que les concepts de productivité et de qualité en matière de logiciels ne font pas l'objet d'une définition rigoureuse unanimement reconnue, nous préciserons, dans un troisième paragraphe les instruments de mesure utilisés.

### **1.1. Le cadre d'analyse**

Les objectifs de la recherche nous ont conduit à retenir le cadre d'analyse présenté figure 1. Ce modèle descriptif traduit l'hypothèse sous-jacente que les résultats obtenus par l'utilisation d'un AGL, en terme de qualité et de productivité découlent à la fois des processus de choix, des processus d'implantation et des processus d'utilisation. Il peut être décomposé en trois sous-modèles spécifiques correspondant respectivement aux processus de choix, d'implantation et d'utilisation de l'AGL (voir pour plus de détails, Varinard, 1997) et qui s'enchaînent chronologiquement. C'est par rapport à ce cadre d'analyse qu'ont été définies et recueillies les données dans les organisations étudiées.

Le sous-modèle relatif au processus de choix pose l'hypothèse que les organisations prennent la décision de recourir à un AGL pour le développement de leurs logiciels afin d'atteindre divers objectifs qui concernent essentiellement les utilisateurs, la maintenance, la planification du travail

**Figure 1 : Cadre d'analyse du processus d'introduction**

des équipes, la culture et les habitudes de travail (Adeli, 1990). Le choix est ensuite réalisé à l'issue d'une démarche (plus ou moins rigoureuse) par application de critères d'évaluation qui peuvent être regroupés en cinq grandes catégories (Du Plessis, 1993) : les critères généraux qui peuvent aider la direction à réaliser une évaluation coûts / bénéfices, les critères de gestion qui ont un rapport avec la planification, le contrôle et les standards des projets, les critères de configuration qui sont liés à l'environnement physique de l'outil, les critères techniques (fonctionnels) qui concernent le support fourni par l'AGL et les critères d'utilisation qui évaluent les possibilités offertes par l'AGL pour le développement d'applications particulières.

Le sous-modèle relatif au processus d'implantation pose l'hypothèse que l'implantation se réalise

en trois étapes (Corbitt, Norman, 1991), à savoir l'installation (mise en place physique de l'outil), l'activation par l'intermédiaire d'un projet pilote réalisé par les développeurs les plus performants et l'institutionnalisation (l'utilisation est étendue à tous les développeurs). Au cours de ce processus, un certain nombre de variables (cf. tableau A de l'annexe 1) ont une influence plus ou moins forte : elles concernent soit les individus, soit l'organisation, soit la technologie implantée.

Enfin, le sous-modèle relatif au processus d'utilisation traduit l'hypothèse que les résultats obtenus au niveau de la qualité des applications et de la productivité des équipes de développement sont fonction d'un certain nombre de variables qui sont liées soit aux individus qui utilisent l'outil, soit au problème à résoudre, soit à l'organisation (et en particulier

aux outils de contrôle et de mesure qui y sont utilisés). Le tableau B de l'annexe 1 détaille les différentes variables indépendantes du cadre d'analyse proposé dont les variables dépendantes sont le niveau de qualité du logiciel produit et le niveau de productivité des développeurs.

A la différence de la plupart des études consacrées au sujet, nous avons tenté d'évaluer l'impact de l'utilisation d'un AGL sur la qualité des logiciels et sur la productivité des équipes de développement à partir de données mesurées et non en fonction de perceptions. Il nous a donc fallu définir une instrumentation de mesure pour la productivité du développement et pour la qualité des logiciels.

## 1.2. Les instruments de mesure

### a. La mesure de la productivité

D'une manière générale, la productivité peut être définie comme le rapport entre produit et facteurs de production.

Pour ces travaux concernant la réalisation d'applications informatiques, nous nous sommes attachés à mesurer la seule productivité du développement (elle concerne les phases d'analyse, de conception, de réalisation et de mise en œuvre des logiciels), laissant volontairement de côté la productivité "à terme" qui résulte principalement des activités de maintenance. Elle correspond à ce qu'un programmeur ou une équipe de développement peut produire dans une période donnée (Jones, 1993). Si on considère un logiciel, on peut donc considérer la productivité comme le rapport du "volume" du produit, c'est-à-dire sa taille, et l'effort consenti

pour le réaliser, traditionnellement appelé "effort de développement".

Pour évaluer la taille des logiciels, une méthode hybride appelée Backfiring a été utilisée. Dans un premier temps, on évalue la taille d'un produit par le nombre de lignes de code source non commentées car il s'agit de la méthode la plus facile à mettre en œuvre. Puis dans un second temps, afin de pouvoir réaliser les indispensables comparaisons entre logiciels ayant été développés avec des langages différents, on calcule le nombre de Points de Fonction correspondant au nombre de lignes de code obtenu, grâce à un tableau de correspondance élaboré par Behrens (voir Dreger, 1989).

L'effort de développement a été mesuré en homme-mois. Il s'agit d'une mesure qui est a priori assez facile à mettre en œuvre puisqu'il suffit d'évaluer le temps qui est consacré pour le développement d'un projet sachant que l'on considère qu'un homme-mois équivaut en moyenne à 152 heures de temps de travail.

En définitive, la productivité des équipes d'étude concernant les divers projets étudiés a donc été mesurée par le ratio :

$$\frac{\text{Nombre total de points de fonction}}{\text{Effort de développement}}$$

### b. La mesure de la qualité

Donner une définition de la qualité en matière de logiciels de gestion est une tâche délicate. Il n'existe aucune définition standard du terme "qualité". Les seules définitions officielles qui peuvent être recensées, nous sont fournies par l'ISO (Organisation Internationale de Standardisation) et l'AFNOR (Association française de Normalisation). Selon la

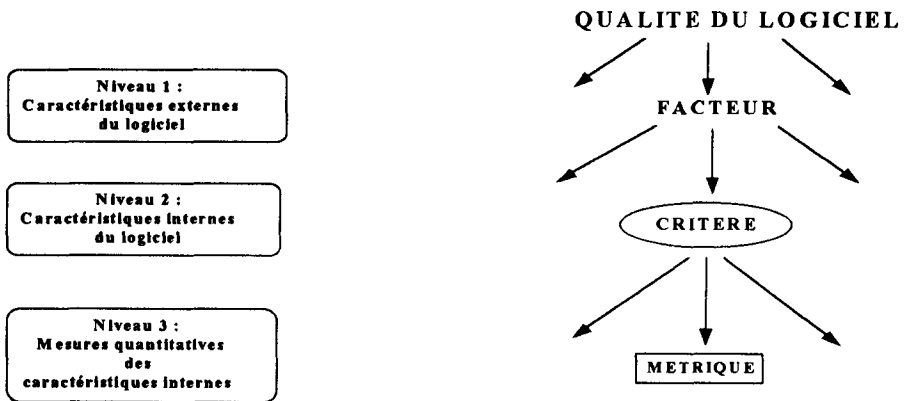
norme ISO 8402 (AFNOR Z 67 133), la qualité correspond à "l'ensemble des propriétés et caractéristiques d'un produit ou service qui lui confèrent l'aptitude à satisfaire les besoins exprimés ou implicites". Selon cette définition, en notre domaine, par produit ou service, il faut entendre application informatique ou logiciel. Autrement dit, la qualité correspond à ce que le client souhaite explicitement ou implicitement.

Un certain nombre de méthodes ont déjà été proposées pour évaluer la qualité d'un logiciel (Factor Criteria Metric, Quality Function Deployment, COQUAMO, Goal Question Metric par exemple). Toutes présentent un certain nombre d'avantages mais aucune n'est vraiment facile à mettre en œuvre. Nous avons donc décidé de définir notre propre méthode (Varinard, 1997) qui est fondée sur le modèle de qualimétrie standardisé par l'Institute of Elec-

trical and Electronics Engineers (IEEE) et qui s'inspire des travaux de Boehm (Boehm, 1978) et McCall (MacCall, 1977). Nous l'avons retenu essentiellement parce que ce modèle constitue la seule approche, véritablement complète et suffisamment commode à mettre en œuvre, du concept subjectif qu'est la qualité.

Dans ce modèle, la qualité globale d'un logiciel dépend tout à la fois de sa qualité intrinsèque et de la qualité du processus de développement. Ces deux pôles sont indépendants car il arrive quelquefois que l'on produise un logiciel de bonne qualité avec de mauvais procédés alors que l'on élabore parfois de mauvais logiciels avec d'excellents procédés. L'approche utilisée est une approche de type hiérarchique<sup>(1)</sup> dans laquelle il y a trois niveaux d'abstraction de la qualité (figure 2) :

**Figure 2 : Illustration du principe de décomposition de la qualité**



(1) Le schéma complet de la décomposition de la qualité est présenté en annexe 2.

• Le premier niveau est celui des Facteurs qui correspondent aux composantes de la qualité du logiciel, identifiées d'un point de vue externe (autrement dit, le point de vue de l'utilisateur). Le modèle est construit autour de deux axes : la qualité du produit et la qualité du processus de développement qui constituent donc les deux facteurs principaux. Ils sont chacun décomposés en sous-facteurs qui eux-mêmes, le sont éventuellement en "sous-sous-facteurs".

• Le second niveau admis correspond aux critères qualité. Ces derniers correspondent à des subdivisions des composantes de haut niveau que sont les facteurs de qualité et sont identifiés à partir d'un point de vue interne au développement, à savoir le point de vue du développeur. Les critères qualité sont donc des attributs du logiciel par l'intermédiaire desquels, les facteurs peuvent être évalués et définis.

• Le dernier niveau retenu est celui des métriques qualité. Il s'agit de mesures quantitatives des critères qui sont toutes en relation avec un critère unique. Elles constituent donc de véritables sondes qui prélèvent l'information qualité sur le logiciel.

La détermination des différents critères qualité est nécessairement précédée par celle des métriques qui permettent de les calculer. Aussi souvent que possible, plusieurs métriques sont associées à chacun des critères qui doivent être évalués. L'évaluation d'un critère  $i$  donné est faite en fonction d'une formule de type :

$$C_i = \frac{1}{N} \sum_N^{i-1} M_{ij}$$

Pour certains critères, la décomposition comporte un niveau supplémentaire et on a donc :

$$M_{ij} = \frac{1}{Z} \sum_Z^{k-1} M_{ijk}$$

L'évaluation d'un critère grâce à la combinaison de plusieurs métriques est préférée à une évaluation par l'intermédiaire d'une métrique unique pour trois raisons principales : tout d'abord, parce qu'il est admis que lorsque l'on combine plusieurs métriques, on élimine nécessairement les difficultés liées à la spécificité de chacune des métriques utilisées. Il est en effet assez rare qu'une métrique unique ait une corrélation forte avec le critère qu'elle doit évaluer. Il en résulte nécessairement qu'en combinant les métriques, on aboutit à une moyenne des spécificités. Ensuite, parce que la combinaison de plusieurs métriques permet de réaliser des distinctions plus fines entre les projets. En effet, lorsque l'on utilise une seule métrique pour mesurer un critère, on ne peut classer les projets, par rapport à celui-ci, qu'en un nombre réduit de groupes, à savoir autant qu'il y a de niveaux sur l'échelle utilisée. Enfin, parce que la combinaison de métriques permet d'augmenter la fiabilité de la mesure du critère et donc de diminuer les erreurs.

Dans un souci d'harmonisation et, pour que les mesures soient réalisables quel que soit l'environnement, chaque métrique utilisée dans notre instrument de mesure correspond à l'évaluation par un certain nombre d'acteurs du projet d'un critère ou d'un sous-critère sur une échelle ordinaire allant de 1 à 7. La valeur 1 représente la réponse "très insatisfaisant" alors que la valeur 7 cor-

respond à la réponse "très satisfaisant". Les valeurs intermédiaires (2 à 6) permettent dès lors d'exprimer un avis plus nuancé sur le critère concerné. Certaines de nos métriques sont calculées en fonction du projet pris dans son ensemble, d'autres le sont pour une phase particulière du cycle de développement. Nous nous référerons pour cela à un cycle classique découpé en quatre

phases : Analyse, Conception, Réalisation, Mise en œuvre.

Comme il est extrêmement difficile de juger de l'importance d'un critère relativement aux autres, aucune pondération des critères n'a été retenue pour les différents calculs. Ainsi, on déterminera par exemple la qualité du processus de production de la manière suivante :

$$F1 = 1/3(F11 + F12 + F13) \text{ avec :}$$

$$F11 \text{ (Capacité à estimer)} = 1/2(C1 + C2)$$

$$F12 \text{ (Capacité à produire)} = 1/2(C3 + C4)$$

$$F13 \text{ (Efficacité d'utilisation des techniques)} = 1/2(C5 + C6)$$

Chacun des critères (C1 à C6 dans notre exemple) est évalué en fonction de plusieurs scores. Il peut s'agir de scores correspondant à différentes métriques mesurant divers aspects du concept étudié et/ou de

scores correspondant à la même métrique évaluée à différentes étapes du cycle de développement. Le critère C1 (Capacité à estimer les délais) est par exemple calculée à partir de huit métriques :

$$C1 = [(M011 + M012) / 2 + (M013 + M014) / 2 + (M015 + M016) / 2 + (M017 + M018) / 2] / 4$$

Le niveau de qualité global d'un logiciel obtenu par cette méthode est donc une valeur comprise entre 1 et 7 qui n'a pas de signification en tant que tel mais constitue un indicateur intéressant en comparaison avec les niveaux de qualité des applications réalisées dans la même organisation. En définitive, c'est donc un score compris entre un et sept et correspondant à un niveau de qualité globale pour un projet logiciel qui va découler des divers calculs réalisés. Ce résultat est obtenu grâce à l'évaluation préalable de 18 critères (grâce aux métriques associées) et de 13 facteurs ou sous-facteurs (grâce aux critères associés). L'annexe 3 de

cette article présente les différentes modalités de calcul des scores de tous les critères étudiés et par suite des différents facteurs qualité retenus. Contrairement à ce que nous avons souhaité initialement, l'instrument défini est donc par certains aspects, subjectif puisqu'il fait appel aux perceptions des hommes mais nous avons tenté d'éliminer au maximum l'arbitraire en multipliant les éléments évalués.

Le cadre d'analyse fixé et les instruments de mesure définis ont déterminé les protocoles de recueil des données. La méthode retenue a tenté d'éliminer au mieux la subjectivité individuelle et la ratio-

nalisation a posteriori (plusieurs acteurs interrogés, recours aux documents). Deux questionnaires ont été élaborés. Le premier pour l'évaluation du contexte de développement et du processus d'introduction de l'AGL dans chaque organisation. Le second pour l'évaluation de la qualité et de la productivité de chaque projet (20 observations).

Pour recueillir les informations, la méthode de l'entretien directif a toujours été utilisée. Ainsi, nous nous sommes efforcés de poser scrupuleusement et exclusivement les questions prévues par les protocoles et toujours dans l'ordre logique préalablement défini. Dans la grande majorité des cas, les réponses apportées par nos différents interlocuteurs ont été confirmées par des archives (pour les projets anciens) ou des documentations du développement (pour les projets les plus récents). Tous les projets, sans exception, ont été évalués par leurs principaux acteurs : directeur informatique, chefs de projets, analystes suivant les cas.

### 1.3. Les sites d'observation

Les trois organisations retenues pour les études longitudinales l'ont été en raison de leurs caractéristiques différentes tant en ce qui concerne le contexte de développement que les méthodes d'introduction de l'AGL. Il s'agit de deux entreprises privées, la première, spécialisée dans la vente par correspondance (VPC), la seconde dans la prestation de services informatiques, et d'une orga-

nisation publique (une administration rectorale) :

**Les 3 Suisses** : cette première organisation est une entreprise très importante de vente par correspondance fondée en 1932, implantée à Roubaix et qui se situe au deuxième rang français dans sa catégorie en termes de chiffre d'affaires (5 500 millions de francs en 1995). Elle emploie 3 200 personnes environ.

Le service informatique de la société a une importance fondamentale puisqu'il sert de support à un nombre très important d'opérations. En effet, près de 100 000 commandes sont traitées chaque jour, et globalement, ce sont près de 25 millions de colis qui sont acheminés chaque année vers les clients de la société. De ce fait, les équipes d'études informatiques sont, en moyenne, confrontées à près de 200 projets de dimensions variées.

Actuellement, environ 15 000 programmes fonctionnent au sein de cette organisation, ce qui induit une importante charge de maintenance puisque selon le directeur informatique, près de la moitié du temps des développeurs est consacrée aux activités de maintenance.

La société a investi en 1988 dans deux AGL utilisés conjointement : AMC\*Designor qui est un AGL de conception et CA-TELON<sup>(2)</sup> qui est à la base, un produit qui couvre l'ensemble du cycle de vie des logiciels mais qui est utilisé pour la partie "réalisation" des projets.

**Le rectorat de Lille** : il s'agit de l'entité de gestion d'une acadé-

(2) Les principales fonctions de ces AGL ainsi que de ceux utilisés au Rectorat de Lille et chez ARES Nord sont présentées en annexe 4.

mie qui comprend les départements du Nord et du Pas-de-Calais. Comme les 29 autres rectorats de France, il constitue une déconcentration du ministère de l'Education nationale. Sa mission essentielle (par l'intermédiaire du Recteur) est de permettre au plan régional l'application de la politique du gouvernement dans le domaine de l'éducation.

En tout, ce sont plus de 87 000 personnes qui relèvent pour leur gestion des différents services de cette administration. Ayant un budget annuel de fonctionnement de l'ordre de 18 milliards de francs, son organisation couvre 4 100 écoles, 451 collèges et 294 lycées. En termes d'usagers de ce service public, l'académie accueille près de 1 000 000 d'élèves et d'écoliers, 150 000 étudiants et 50 000 stagiaires de formation permanente.

Le service informatique de cette administration est constitué de 60 personnes, toutes salariées du rectorat. 14 personnes sont affectées aux activités de développement proprement dites. Ce chiffre peut paraître peu important mais il s'explique par le fait que chaque rectorat ne réalise pas toutes les applications qu'il utilise. En fait, pour la réalisation des applications communes à tous les rectorats et donc d'une certaine importance, le ministère désigne à tour de rôle chacun des différents services informatiques académiques. Une fois terminées, ces applications sont fournies à toutes les autres académies. En conséquence, les équipes d'études de chaque rectorat ne développent que les logiciels pour lesquels elles ont été désignées ou d'éventuelles applications spécifiques à l'académie. Dès lors, leur nombre n'est donc pas très important

(une quinzaine par an) mais il s'agit en majorité de projets de taille importante.

Le nombre des programmes fonctionnant dans un rectorat est extrêmement important (de l'ordre de plusieurs milliers), ce qui explique que le temps consacré aux activités de maintenance par le service, soit considérable.

Les équipes d'études sont dotées de l'AGL Système DELF. Il s'agit d'un AGL de conception conçu par la société DELF en 1987. Il est extrêmement répandu dans les administrations (60 % du CA de la société), ce qui peut éventuellement expliquer que le ministère de l'Education nationale ait opté pour ce produit. L'outil n'est pas encore employé systématiquement pour tous les projets mais les équipes de développement l'utilisent de plus en plus fréquemment.

**La société ARES** : avec 3 000 clients, 1 700 progiciels installés et plus de 500 réseaux d'entreprises mis en place, cette société est actuellement une des toutes premières SSII Françaises. Son siège est à Paris mais elle est implantée sur tout le territoire français puisqu'elle dispose d'une présence régionale à Lille, Rouen, Caen, Nantes, Lyon, Bordeaux et Strasbourg.

Son activité se situe à trois niveaux différents : l'édition et la distribution de progiciels, l'ingénierie des systèmes d'information, la distribution d'outils et de matériel.

Le chiffre d'affaires total de la société pour 1995 était de 650 millions de francs pour un résultat bénéficiaire de 14 millions de francs. ARES employait cette même année, 420 personnes.

Nos études ont été réalisées en collaboration avec l'antenne régionale de Lille (Ares Nord) qui comme toutes les directions régionales, tente de répondre à l'attente des entreprises en matière d'architecture de systèmes d'information, que ce soit pour le pilotage de projets, l'intégration de solutions, l'assistance technique, le développement sur Systèmes de Gestion de Bases de Données, la gestion de réseaux et de systèmes ainsi que la tierce maintenance applicative. Pour cette activité, elle dispose d'un effectif d'environ trente personnes.

Les développeurs d'ARES disposent depuis juin 1995 du produit Oracle\*Case. Il s'agit d'un outil intégré qui comprend une partie conception (DESIGNER/2000) et une partie réalisation (DEVELOPER/2000). Son support est utilisé par les équipes de développement d'Ares à toutes les étapes du cycle de vie des logiciels.

Au total, 20 projets se répartissant comme l'indique le tableau 1 ont été évalués dans ces trois organisations au niveau de la qualité des logiciels produits et de la productivité des équipes de développement.

**Tableau 1 : Répartition des 20 projets évalués**

	Projets conduits sans AGL	Projets conduits avec AGL
Les 3 Suisses	3	4
Rectorat de Lille	3	4
Ares	3	3

L'application de ce cadre méthodologique a permis de mettre en évidence différents résultats qui sont présentés dans la seconde partie de cet article.

## **II. L'IMPACT DE L'UTILISATION D'UN AGL : L'HYPOTHESE DE FACTEURS CRITIQUES DU SUCCES**

L'analyse des observations recueillies au sein des trois organisations étudiées conduit à formuler la proposition suivante : l'introduction d'un AGL n'entraîne pas automatiquement une amélioration de la qualité des logiciels et de la productivité des équipes de développement ; dans la mesure où l'effet de l'utilisation n'est pas systématique, on peut formuler

l'hypothèse de l'existence de facteurs critiques du succès.

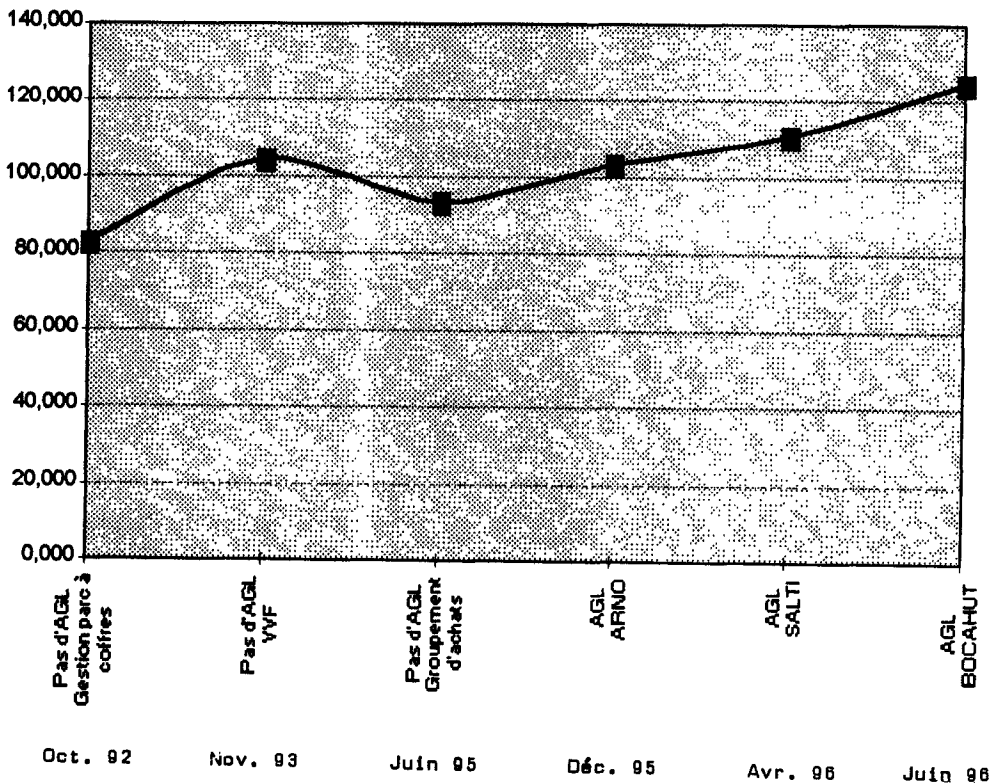
### **2.1. Le caractère non systématique de l'impact positif (de l'utilisation de l'AGL sur la productivité des équipes de développement et sur la qualité des logiciels)**

En raison de difficultés de collecte de données fiables, la productivité des équipes de développement n'a pu être mesurée que dans deux des trois organisations (Ares, rectorat de Lille). L'analyse des mesures permet de mettre en évidence le résultat suivant : il y a amélioration de la productivité du développement ; cette amélioration n'est pas immédiate ni automatique. Comme le montre la

figure 3 (cas Ares), l'évolution de la productivité mesurée au niveau des projets, semble traduire un phénomène d'apprentissage. Cependant, les mesures relatives au rectorat de Lille révèlent également que cet apprentissage ne se déclenche pas systématiquement avec l'utilisation de l'outil (figure 4). On remarque qu'au contraire, l'utilisation de DELF correspondit à un score très satisfaisant en

termes de productivité pour le premier projet développé (Médecine scolaire) puis les résultats dans ce domaine diminuèrent régulièrement jusqu'à atteindre un niveau très faible pour le projet MAMISE, sans doute à cause du manque de motivation des développeurs vis-à-vis d'une technologie dont l'emploi n'est toujours pas imposé par la direction informatique.

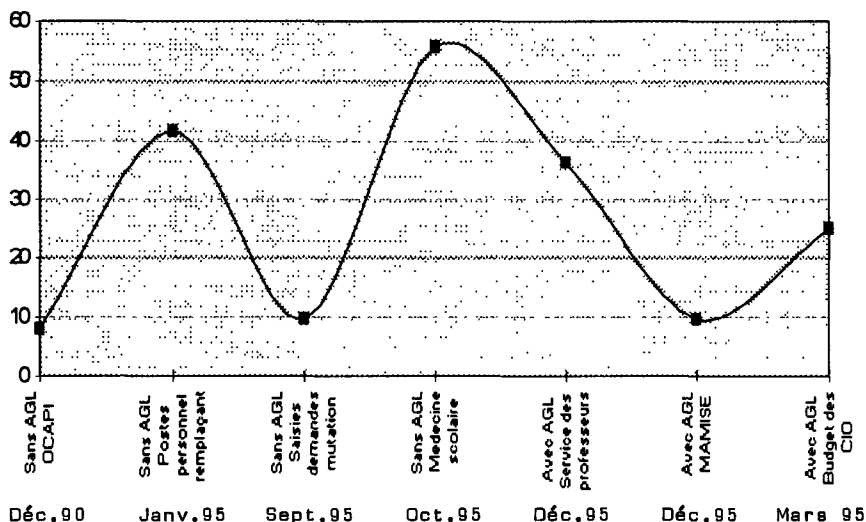
**Figure 3 : Evolution de la production chez Ares**



Un constat très voisin peut être fait pour ce qui concerne l'évaluation de la qualité : le recours à un Atelier de Génie Logiciel semble également avoir un impact positif sur le niveau de qualité globale des applications développées.

Les mesures réalisées aux 3 Suisses et chez Ares Nord le démontrent assez nettement (voir tableau 2). Néanmoins, cet impact n'est pas automatique comme le confirme l'exemple du rectorat de Lille.

**Figure 4 : Evolution de la productivité au rectorat de Lille**



**Tableau 2 : Ecart constaté sur la qualité globale perçue depuis le recours à l'AGL**

	Les 3 Suisses	Rectorat de Lille	Ares Nord
Niveau de qualité moyen des projets réalisés sans AGL	4,603	4,766	3,912
Niveau de qualité moyen des projets réalisés avec AGL	5,454	5,057	4,587
Ecart en %	+18,47 %	+6,32 %	+17,25 %

On peut remarquer que l'impact semble exister d'une manière assez comparable au niveau des deux composantes principales de la qualité globale, à savoir la qualité du processus de développement et la qualité intrinsèque du logiciel. On notera d'ailleurs qu'aux 3 Suisses, l'impact a été plus fort au niveau de la qualité intrinsèque des applications alors que chez Ares, c'est au niveau du processus de production qu'il a été le plus

nettement ressenti (voir tableau 3).

En outre, les observations montrent que l'amélioration de la productivité ne se réalise pas au détriment de celle de la qualité ; il y a absence de phénomène de substitution. Le cas Ares semble suggérer l'existence d'un apprentissage efficace pour les deux dimensions étudiées (la confirmation de ce phénomène exigerait cependant une observation sur une période plus longue que celle retenue).

**Tableau 3 : Ecart constatés sur les deux composantes principales de la qualité globale**

	Les 3 Suisses	Ares Nord
Niveau de qualité moyen des processus de production - Projets sans AGL	4,659	3,640
Niveau de qualité moyen des processus de production - Projets avec AGL	5,093	4,985
Ecart en %	<b>+9,32 %</b>	<b>+36,95 %</b>
Niveau de qualité moyen des logiciels - Projets sans AGL	4,548	4,185
Niveau de qualité moyen des logiciels - Projets avec AGL	5,812	4,189
Ecart en %	<b>+21,75 %</b>	<b>+0,10 %</b>

En résumé, il est possible de faire le constat suivant :

- l'utilisation d'AMC Designor et TELON aux Trois Suisses a un impact positif sur la qualité des logiciels ;
- l'utilisation de Système DELF au rectorat de Lille n'a pas eu, pour l'instant, d'impact positif ni au niveau de la productivité, ni au niveau de la qualité ;
- le recours à ORACLE pour les développeurs à un impact positif simultanément sur la productivité et sur la qualité.

Il est clair que des différences apparaissent entre les diverses organisations quant aux effets positifs de l'utilisation de l'AGL. Il convient dès lors de rechercher s'il existe des éléments explicatifs de ces différences observées ; il ne semble pas que les caractéristiques spécifiques des outils utilisés jouent un rôle important dans l'explication de ces différences (les fonctionnalités des AGL sont très voisines) ; en revanche, l'analyse des processus d'introduction de la nouvelle technologie dans les trois organisations semble suggérer le rôle important de certains choix

de gestion et donc l'existence de facteurs clés de succès.

## **2.2. L'existence de facteurs critiques du succès de l'introduction d'un AGL dans une organisation**

L'étude rétrospective des processus d'introduction et les entretiens conduits avec les différents acteurs dans chaque organisation nous amènent à attribuer un rôle essentiel à quatre facteurs :

- l'implication des acteurs dans le choix de l'AGL,
- le recours à des "champions de la technologie",
- la formation des utilisateurs,
- le passage par des projets pilotes.

### **2.2.1. L'implication des acteurs dans le choix de l'AGL**

L'échec du recours à un AGL dans le service informatique du rectorat de Lille et les succès plus ou moins importants constatés chez Ares et aux 3 Suisses nous permettent de mettre en évidence l'importance d'une implication des

futurs utilisateurs dans le processus de choix de l'outil. Compte tenu de la taille et de la configuration de l'organisation, les membres des équipes d'études du rectorat n'ont pas eu la possibilité d'explicitier leurs objectifs et donc de choisir eux-mêmes un outil leur permettant de les atteindre et de prendre en compte les contraintes de l'entreprise. L'effet négatif de cet élément fut du reste souligné par le directeur informatique du rectorat qui précisa en outre "qu'il lui fut très difficile de motiver ses développeurs à utiliser un outil pour le choix duquel ils n'avaient pas été consultés". A l'inverse, aux 3 Suisses, le choix d'un environnement de génie logiciel composé d'AMC Designor et de Telon a été réalisé à l'issue d'une démarche raisonnée prenant en compte les objectifs à atteindre et les caractéristiques organisationnelles, démarche à laquelle ont été associés les développeurs. Il n'y eut ainsi pas de rejet de la part de certains développeurs comme ce fut visiblement le cas au rectorat. Chez Ares, le choix du Case d'Oracle fut immédiatement évident pour tous les développeurs, compte tenu de l'existence de relations de partenariat entre les deux sociétés et aucun phénomène de résistance au changement ne fut donc observé. Ces différentes observations confirment l'hypothèse déjà formulée (Norman *et al.*, 1989) selon laquelle il semble important que les opérateurs futurs (les développeurs) soient impliqués dans le choix de l'outil qu'ils vont devoir utiliser. Compte tenu des coûts correspondant à l'acquisition d'un outil de ce type, il paraît souhaitable, pour chaque organisation, de se donner les moyens de choisir celui qui paraît le plus en

adéquation avec les besoins exprimés par les développeurs eux-mêmes. Il est intéressant de noter que pour certains auteurs, l'implication des futurs utilisateurs de l'outil ne doit pas concerner uniquement le processus de sélection mais la totalité du processus d'implantation (Sumner, 1993 ; Azani, Khorramshahgol, 1993).

### **2.2.2. Le recours à des "champions" de la technologie**

De l'avis de nos différents interlocuteurs, il importe de confier l'implantation du nouvel outil aux développeurs les plus compétents et les plus motivés par l'utilisation de la nouvelle technologie. Cette observation confirme tout à fait ce que nous avons pu constater lors de nos observations sur le terrain (aux 3 Suisses et chez Ares Nord). Une fois encore, l'échec (provisoire ?) de l'utilisation de Système DELF au rectorat de Lille corrobore cette impression puisqu'aucun "champion" de la technologie proposée ne s'est dégagé au moment de l'implantation de l'AGL. En fait, seul le directeur informatique tenta un moment de remplir cette mission mais ses autres fonctions ne lui permirent pas d'y consacrer le temps nécessaire. On peut également ajouter que le fait que l'utilisation systématique de l'AGL n'ait pas été imposée, a sans aucun doute conduit les développeurs, par confort, à continuer à utiliser les méthodes traditionnelles. A l'inverse, les bons résultats obtenus chez Ares, ont sans doute été favorisés par le fait que la plupart des développeurs étaient sensibilisés aux bénéfices à retirer de l'utilisation d'Oracle\*Case avant même son implantation. Ceci confirme que la

motivation des acteurs est un élément fondamental pour que l'introduction d'une nouvelle technologie soit couronnée de succès. En tout état de cause, cette dernière peut sans aucun doute être accrue grâce à l'intervention de "champions" qui seront chargés non seulement, d'implanter la technologie dans les meilleures conditions mais aussi, de convaincre leurs collègues du bien-fondé de son utilisation. L'idée qu'un ou des "champions" de la nouvelle technologie soient indispensables à la réussite de l'introduction d'un AGL dans une organisation est reprise par plusieurs auteurs (Corbitt, Norman, 1991 ; Everest, Alanis, 1993 par exemple). Certains considèrent que les chances d'obtenir de bons résultats lors des premières utilisations de l'AGL sont en effet plus importantes si ce sont les meilleurs développeurs qui sont concernés (Burkhard, 1989 ; Hayley, Lyman, 1990). Ceux-ci doivent donc être formés avant les autres à l'utilisation de la méthodologie et de l'outil pour qu'ils jouent ensuite le rôle d'experts "maison" et constituent dès lors des équipes de soutien aux équipes de développement (Zagorsky, 1990).

### **2.2.3. La formation des utilisateurs**

Les diverses personnes interrogées ont également suggéré que l'utilisation de l'Atelier de Génie Logiciel ne peut être couronnée de succès que dans la mesure où les trois conditions suivantes sont relevées :

- tout d'abord, les membres des équipes de développement doivent être compétents tant dans l'utilisation de l'AGL que dans l'utilisation de la méthodologie qu'il sup-

porte. Ils doivent en particulier avoir une certaine expérience concernant l'utilisation tant de l'outil que de la méthodologie structurée de développement à laquelle il est lié ;

- ensuite, l'ensemble des membres des équipes doivent être motivés par le recours à une nouvelle technologie en ayant l'optique d'améliorer les performances en termes de qualité et de productivité.

Toutes ces suggestions, comme les différentes observations que nous avons pu faire, mettent en évidence l'importance de la phase de formation des développeurs. En effet, chaque organisation qui implante un AGL doit se donner les moyens (en temps et en argent) de former l'ensemble des futurs acteurs, tout à la fois à l'outil et à la méthode (Burkhard, 1989). Il doit y avoir une formation initiale, précédant l'introduction de l'outil (Wilson, 1989) afin d'éviter que certains développeurs, non formés à l'origine, refusent ensuite le changement engendré par l'introduction du nouvel outil (voir par exemple le cas du rectorat de Lille). Il doit cependant y avoir également une formation continue et un perfectionnement "sur le tas" une fois que l'outil a été mis en œuvre afin d'assurer une efficacité maximum dans l'utilisation. Pour confirmer l'importance de la formation et de l'expérience des développeurs vis-à-vis de l'outil, nous pouvons relever que l'impact de l'AGL semble plus important chez Ares qu'aux 3 Suisses où la très grande majorité des développeurs n'avaient aucune expérience de l'utilisation d'un Atelier de Génie Logiciel avant qu'AMC Designer et Telon ne soient implantés. Chez Ares, au contraire, la plupart des développeurs connais-

saient déjà les grands principes de fonctionnement du Case d'Oracle au moment de l'institutionnalisation de son utilisation. Notons pour terminer qu'il paraît préférable que la formation concerne dans un premier temps les développeurs les plus brillants (Hayley, Lyman, 1990) et qu'elle soit ensuite étendue aux autres lorsque les premiers peuvent jouer le rôle d'experts internes (Zagorsky, 1990).

#### **2.2.4. Le passage par des projets pilotes**

Ce facteur semble également un élément déterminant de la réussite future de l'utilisation de l'AGL (Menendez, 1991). Le choix des premiers projets est aussi important que celui des individus qui vont les développer. Il faut qu'ils ne soient ni trop petits (pour que l'utilisation de l'AGL ait un intérêt), ni trop gros (pour que les premiers utilisateurs ne soient pas noyés sous les données et qu'ils puissent se focaliser principalement sur la technologie) et ils ne doivent pas être trop complexes afin de ne pas ternir l'image de la nouvelle technologie en cas d'échec (Burkhard, 1989). Comme rien n'avait été effectué dans ce sens au rectorat de Lille, la plupart des développeurs se sont plaints d'avoir à utiliser l'AGL directement sur des projets réels et souvent complexes. Cette erreur a eu pour conséquence directe que les développeurs ont, pour la plupart, renoncé (temporairement ?) à utiliser l'AGL et ont de nouveau recouru aux méthodes traditionnelles de développement. Ce recours aux projets pilotes doit être considéré comme la suite logique du programme de formation (Hayley, Lyman, 1990). Il précède la mise en œuvre de l'outil (Ben-

dure, 1991) et il permet notamment aux individus chargés de cette tâche d'acquérir un minimum d'expérience (notion de savoir-faire tacite) vis-à-vis de l'outil et de la méthodologie. De plus, lorsque l'impact de l'utilisation de l'AGL peut être démontré à travers la conduite de ces projets, cela renforce la motivation et le moral des futurs utilisateurs de l'AGL. Pour cette raison, il est essentiel que la communication soit la plus ouverte possible s'agissant de la réalisation des projets pilotes afin que les éventuels bons résultats obtenus grâce à la nouvelle technologie, soient connus du plus grand nombre. Précisons, pour en terminer sur ce point, qu'en général cette période d'apprentissage sur des projets pilotes ne débouche pas directement sur une utilisation systématique de l'AGL. Elle est traditionnellement suivie d'une deuxième période au cours de laquelle l'utilisation de l'AGL est progressivement étendue à tous les projets. Cette méthode a particulièrement bien fonctionné pour Les trois Suisses et Ares puisqu'une fois l'AGL correctement implanté, les résultats en termes de qualité et de productivité y sont très vite devenus intéressants.

L'étude empirique a mis en évidence, dans les trois organisations étudiées, l'existence de quatre facteurs clés du succès. Il est intéressant de noter qu'ils sont en partie identiques à ceux concernant n'importe quel projet d'informatisation qui ont été mis en évidence, entre autres, par Delone et Mac Lean (1992). Le faible nombre d'observations comme la durée relativement brève de la période étudiée, interdisent de considérer ce résultat comme définitif et généralisable ; il s'agit davan-

tage d'une hypothèse à confirmer que l'énoncé d'une règle universelle à prétention normative. Il faut d'ailleurs souligner qu'un certain nombre d'autres éléments ont été reconnus, dans des études précédentes, comme susceptibles de jouer un rôle important dans la réussite de l'utilisation de l'AGL (l'utilisation avant l'implantation d'une méthodologie de développement structuré par exemple). Le fait que, dans cette étude empirique, ces facteurs n'aient pas été jugés comme déterminants par nos interlocuteurs ne saurait en aucun cas être considéré comme l'invalidation des résultats des études antérieures.

## CONCLUSION

La dimension limitée de cette étude interdit toute conclusion définitive ; elle montre cependant que la technologie AGL peut permettre d'améliorer simultanément la productivité du développement et la qualité des logiciels produits. Elle suggère l'existence de phénomènes d'apprentissage organisationnel plus ou moins rapides selon les organisations. Elle confirme le caractère non automatique de l'impact positif de la technologie (absence d'un déterminisme technologique systématique) et le rôle déterminant des processus d'introduction. Comme pour tout système de caractère socio-technique, la gestion du changement organisationnel conditionne le succès de l'utilisation. En cela, l'introduction d'un AGL ne constitue qu'un exemple supplémentaire venant compléter une longue série d'observations familières aux spécialistes des sciences de l'organisation. En revanche, la mise au point définitive d'un modèle nor-

matif d'introduction de cette technologie particulière précisant la hiérarchie des facteurs clés de succès appelle, à l'évidence, des études complémentaires.

## BIBLIOGRAPHIE

Aaen, I. (1994), « Problems in CASE Introduction : Experiences from User Organizations », *Information and Software Technology*, Vol. 36, n° 11, p. 643-654.

Adeli, (direction Paul Theron) (1990), *Génie logiciel et productivité des études*, Afnor technique, Paris la Défense, Octobre.

Azani, C.H., Khorramshahgol, R. (1993), « The Adoption and Implementation of CASE Technology » in *Computer-Aided Software Engineering. Issues and Trends for the 1990s and Beyond*, Thomas J. Bergin (ed.), Idea Group Publishing, Harrisburg (USA) et Londres (UK), p. 211-226.

Banker, R.D., Kaufman, R.J. (1991), « Reuse and Productivity in ICASE : an Empirical Study », *MIS Quarterly*, Vol 15, n° 1, September, p. 375-401.

Barki, H., Hartwick, J. (1989), « Re-thinking the Concept of User Involvement », *MIS Quarterly*, March, p. 53-63.

Bendure, C.O. (1991), « A Case Study on CASE. Its Evolution and Use at HHMI », *Information Systems Management*, Automne 1991, p. 50-56.

Boehm, B.W. et al. (1978), *Characteristics of Software Quality*, North Holland Publishing Company.

Brisebois, R., Dion, P. (1990), « Implementing CASE. Six critical factors », *Computing Canada*, Vol. 16, n° 11, May 24, p. 21-22.

Burkhard, D.L. (1989), « Implementing CASE Tools », *Journal of Systems Management*, Vol. 40, n° 5, May 20, p. 20-25.

Card, D.N., McGarry, F.E., Page, G.T. (1987), « Evaluating Software Engineering Technologies », *IEEE Transactions on Software Engineering*, Vol. 13, n° 7, July, p. 845-851.

- Corbitt, G.F., Norman, R.J. (1991), « CASE Implementation : Accounting for the Human Element », *Information and Software Technology*, Vol. 33, n° 9, November, p. 637-640.
- Corbitt, G.F., Norman, R.J., Butler, M.C. (1991), « Assessing Proximity to Fruition : a Case Study of Phases in CASE Technology Transfer », *International Journal of Software Engineering*, Vol. 1, n° 2, June, p. 189-201.
- Currid, C. (1993), « A Case against CASE : it's Expensive and often Unsuccessful », *Infoworld*, August 2, p. 68.
- Delone, W.H., MacLean, E.R. (1992), « Information System Success : the Quest for the Independent Variable », *Information Systems Research*, Vol. 3, n° 1, p. 60-95.
- Dreger, B.J. (1989), *Function Point Analysis*, Prentice-Hall.
- Du Plessis, A.L. (1993), « A Method for CASE Tool Evaluation », *Information and Management*, n° 25, p. 93-102.
- Jones, C. (1993), *Software Productivity and Quality Today : the Worldwide Perspective*, IS Management Group, Carlsbad.
- Everest, G.C., Alanis, M. (1993), « Survey of CASE User Experiences », *Computer-Aided Software Engineering. Issues and Trends for the 1990s and Beyond*, Thomas J. Bergin (ed.), Idea Group Publishing, Harrisburg (USA) et Londres (UK), p. 227-246.
- Finlay, P.N., Mitchell, A.C. (1994), « Perceptions of the Benefits from the Introduction of CASE : an Empirical Study », *MIS Quarterly*, December, p. 353-370.
- Franz, C.R., Robey, D. (1986), « Organizational Context, User Involvement, and the Usefulness of Information Systems », *Decision Sciences*, Vol. 17, July, p. 329-356.
- Hayley, K.J., Lyman, T.H. (1990), « The Realities of CASE », *Journal of Information Systems Management*, Summer, p. 18-23.
- Jarvenpaa, S.L., Ives, B. (1991), « Executive Involvement and Participation in the Management of Information Technology », *MIS Quarterly*, Vol. 2, n° 15, June, p. 205-227.
- Lee, T.H., Chen, M., Norman, R.J. (1989), « Computer-Aided Software Engineering (CASE) Adoption and Implementation : a Theoretical Framework Analysis from an Organisational Innovation Perspective », *Proceedings of the 24th Annual Hawaii International Conference of Systems Sciences*, Vol. 3, Koloa (Kauai), HA, USA, January, p. 3-17, 8-11.
- Lin, C.Y., Chung, C.H. (1991), « End-User Computing in a CASE Environment », *Journal of Information Systems Management*, Spring, p. 17-21.
- MacCall, A. et al. (1977), *Factors in Software Quality*, Report N° NTIS AD/A-049 014 015, US Rome Air Development Center, National Technical Information services.
- Menendez, D.A. (1991), « The Impact of CASE », *Internal Auditor*, Vol. 48, n° 6, December, p. 48-53.
- Necco, C., Tsai, N.W., Hogelson, K.W. (1989), « Current Usage of CASE Software », *Journal of Systems Management*, May, Vol. 40, n° 5.
- Norman, R.J., Numamaker, J.F., (1988), « An Empirical Study of Information Systems Professionals' Productivity Perceptions of CASE Technology », *Proceedings of the Ninth International Conference on Information Systems*, J.I. DeGross, M.H. Olsson (eds), New York, p. 111-118.
- Norman, R.J. et al. (1989), « CASE Technology Transfer : a Case Study of Unsuccessful Change », *Journal of Systems Management*, Vol. 40, n° 5, May, p. 33-37.
- Norman, R.J., Corbitt, G.F. (1991), « A Cure for Technology Implementation Headaches », *Chief International Officer Journal*, Vol. 3, n° 4, Spring, p. 35-38.
- Orlikowski, W.J. (1988), « CASE Tools and the Information System Workplace : Some Findings from Empirical Research », *Proceedings of the ACM SIGCPR conference*, E.M. Awad (ed.), ACM press, Baltimore, April, p. 88-97.
- Orlikowski, W.J. (1989), « Division Among the Ranks : The Social Implications of CASE Tools for System Developers » *Proceedings of the Tenth International Conference on Information*

*Systems*, J.I. DeGross, J.C. Henderson, B.R. Konsynski (eds), New York, p. 199-210.

Rowe, J.M. (1993), « Can Effort Standardization Affect CASE Usage ? », *Journal of Systems Management*, March, p. 29-33.

Selamat, et al. (1994), « Non-Use Phenomenon of CASE Tools : Malaysian Experience », *Information and Software Technology*, Vol. 36, n° 9, p. 531-537.

Stobart, S.C., Thompson, J.B., Smith, P. (1991), « Use, Problems, Benefits and Future Direction of CASE in United Kingdom », *Information and Software Technology*, Vol. 33, n° 9, November 9, p. 629-636.

Sullivan, Kristine B. (1988), « Using CASE Tools Effectively Requires Training », *PC Week*, July 11.

Sumner, M. (1993), « Factors Influencing the Adoption of CASE » in *Computer-Aided Software Engineering. Issues and Trends for the 1990s and Beyond*, Thomas J. Bergin (ed.), Idea Group Publishing, Harrisburg (USA) et Londres (UK), p. 130-155.

Swanson, K. et al. (1991), « The Application Software Factory : Applying Total Quality Techniques to Systems Development », *MIS Quarterly*, Vol. 15, n° 4, December, p. 567-579.

Urwiler, R. et al. (1995), « Computer-Aided Software Engineering : the Determinants of an Effective Implementation Strategy », *Information and Management*, Vol. 29, n° 4, October, p. 215-225.

Varinard, C. (1997), *Impact de l'utilisation des Ateliers de Génie Logiciel en informatique de gestion (étude exploratoire)*, Thèse, Université Montpellier II, Mai.

Vessey, I., Jarvenpaa, S.L., Tractinsky, N. (1992), « Evaluation of Vendor Products : CASE Tools as Methodology Companions », *Communications of the ACM*, Vol. 35, n° 4, April, p. 90-105.

Wacheux, F. (1996), *Méthodes qualitatives et recherche en gestion*, Economica, Paris.

Wilson, D.N. (1989), « CASE : Guidelines for Success », *Information and Software Technology*, Vol. 31, n° 7, September, p. 346-350.

Yellen, R. (1990), « Systems Analysts Performance Using CASE Versus Manual Methods », *Proceedings of the twenty-third annual Hawaii international conference on system sciences*, IEEE computer society press, Los Alamitos, CA.

Zagorsky, C. (1990), « Case Study : Managing the Change to CASE », *Journal of systems management*, Summer, p. 24-32.

**ANNEXE 1****Tableau A : Variables associées au modèle de processus d'implantation des AGL**

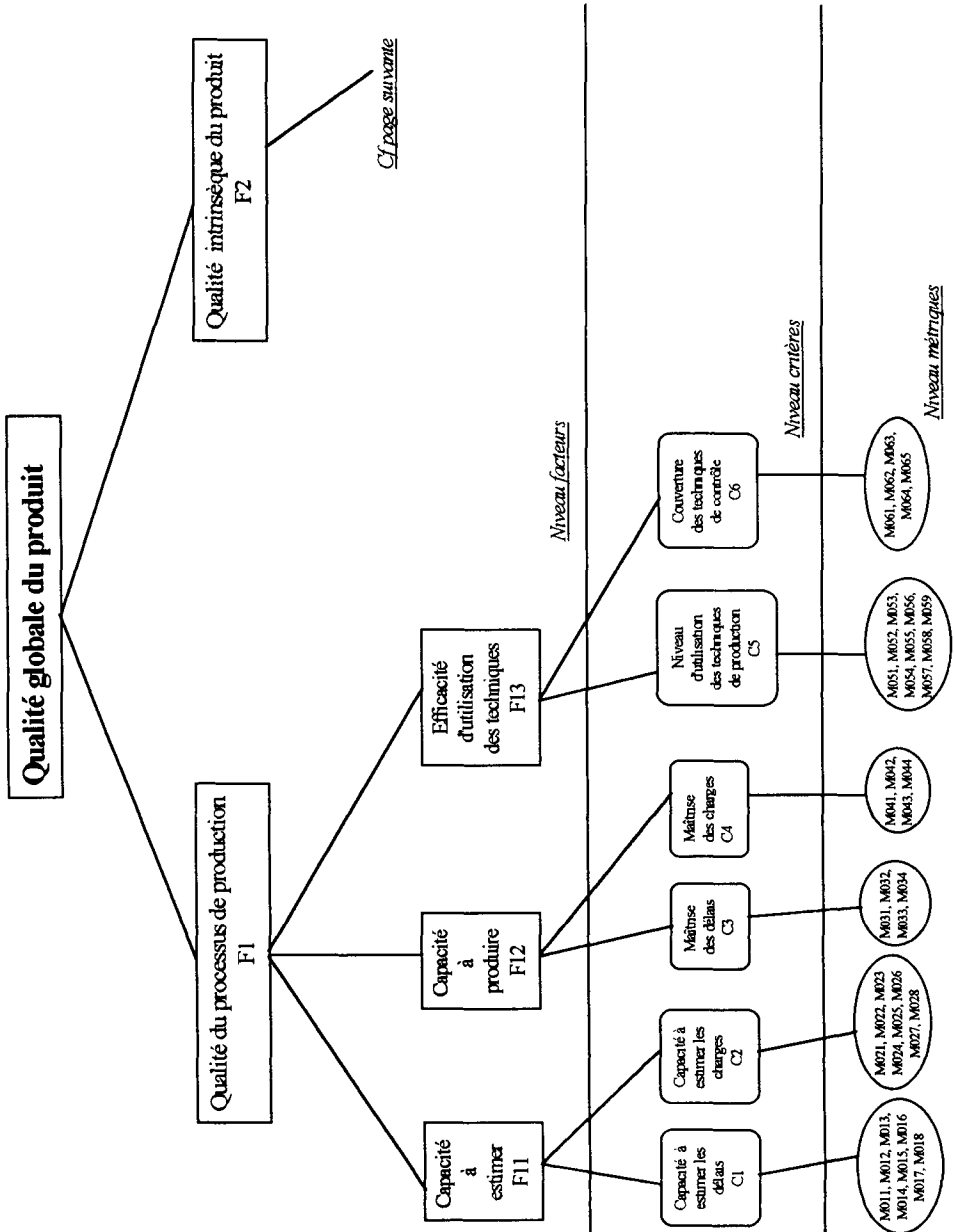
<b>Variables relatives aux individus</b>
<ul style="list-style-type: none"> <li>- Compétence des individus chargés de l'implantation concernant la méthodologie</li> <li>- Compétence des individus chargés de l'implantation concernant l'AGL</li> <li>- Expérience des individus chargés de l'implantation vis-à-vis de la méthodologie</li> <li>- Expérience des individus chargés de l'implantation vis-à-vis de l'outil</li> <li>- Motivation des individus chargés de l'implantation</li> <li>- Soutien apporté par les responsables du département Systèmes d'Information</li> <li>- Soutien apporté par les responsables de l'organisation</li> <li>- Niveau des consultants extérieurs</li> <li>- Expérience des consultants extérieurs</li> <li>- Consentement des "clients" vis-à-vis du changement engendré par la technologie</li> </ul>
<b>Variables relatives à l'organisation</b>
<ul style="list-style-type: none"> <li>- Taille de l'organisation</li> <li>- Complexité de l'organisation</li> <li>- Expérience de l'organisation vis-à-vis de l'introduction d'innovations</li> <li>- Expérience de l'organisation vis-à-vis de l'utilisation d'une méthodologie structurée</li> <li>- Expérience de l'organisation vis-à-vis de la méthodologie supportée par l'AGL</li> <li>- Complexité de l'environnement de développement avant l'introduction</li> <li>- Degré d'ouverture des communications au sujet de la technologie</li> <li>- Modes de travail des équipes avant l'implantation</li> <li>- Taille des équipes de développement</li> </ul>
<b>Variables relatives à la nouvelle technologie implantée</b>
<ul style="list-style-type: none"> <li>- Facilité d'installation de l'innovation</li> <li>- Facilité de personnalisation de l'innovation</li> <li>- Compatibilité de l'innovation avec l'environnement</li> <li>- Complexité de l'innovation (des concepts et des techniques utilisées)</li> <li>- Capacité de l'innovation à intégrer des applications anciennes</li> </ul>

**Tableau B : Variables concernées par le modèle d'utilisation**

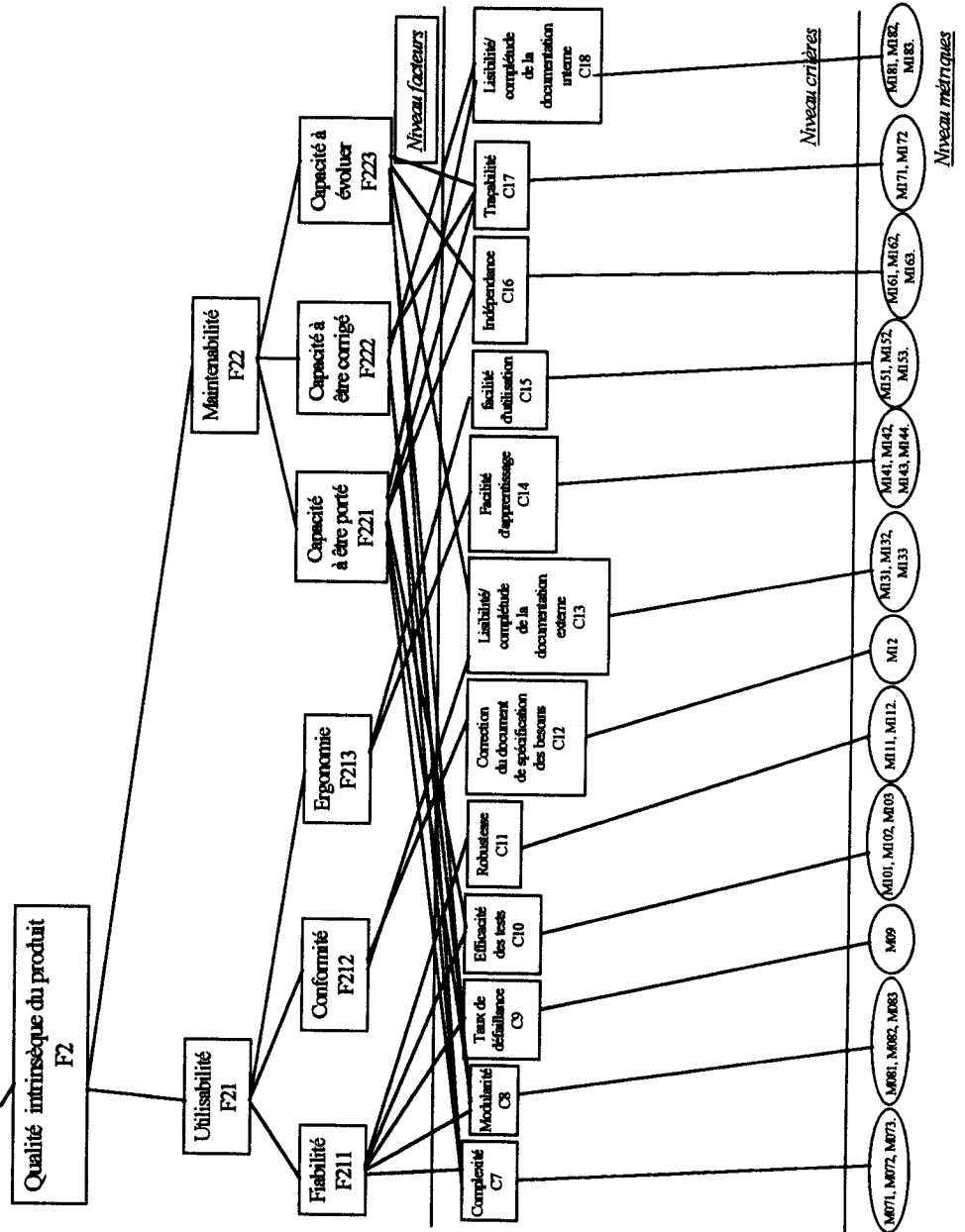
<p><b>Variables liées aux individus</b></p> <ul style="list-style-type: none"> <li>- Niveau de compréhension individuel de la manière dont l'outil s'adapte à l'approche utilisée pour développer les systèmes</li> <li>- Compétence des développeurs vis-à-vis de l'outil</li> <li>- Compétence des développeurs vis-à-vis de la méthodologie</li> <li>- Expérience des développeurs vis-à-vis de la méthodologie</li> <li>- Expérience des développeurs vis-à-vis de l'outil</li> <li>- Expérience des développeurs vis-à-vis du type d'application développée</li> <li>- Education des développeurs vis-à-vis du changement culturel nécessaire</li> <li>- Education des "clients" vis-à-vis du changement culturel espéré</li> <li>- Education des "clients" vis-à-vis des aspects techniques de l'AGL</li> <li>- Education des développeurs concernant le domaine d'activité de l'organisation</li> <li>- Moral des développeurs</li> <li>- Niveau de soutien de la direction générale</li> <li>- Niveau de soutien des responsables du département Systèmes d'Information</li> <li>- Disponibilité et qualité des consultants extérieurs</li> </ul>
<p><b>Variables liées au problème posé (c'est-à-dire liées à l'application à développer)</b></p> <ul style="list-style-type: none"> <li>- Complexité du problème posé</li> <li>- Volume de l'application</li> <li>- Possibilités de réutilisation</li> <li>- Sécurité requise</li> <li>- Volume de données à traiter</li> </ul>
<p><b>Variables liées à l'organisation</b></p> <ul style="list-style-type: none"> <li>- Qualité des instruments de contrôle du processus de développement</li> <li>- Qualité des outils de mesure de la qualité des produits</li> <li>- Méthodes et techniques de développement utilisées</li> <li>- Méthodes de gestion de projets pratiquées</li> <li>- Ressources matérielles disponibles</li> </ul>

**ANNEXE 2**

**Schéma de décomposition de la qualité (1<sup>re</sup> partie)**



**Schéma de décomposition de la qualité (2<sup>e</sup> partie)**



## ANNEXE 3

**Modalités de calcul du niveau de qualité global d'un logiciel**

<i>ELEMENT A CALCULER</i>	<i>DESIGNATION</i>	<i>TYPE</i>	<i>MODE DE CALCUL</i>
Qualité globale du logiciel	F0	<i>Facteur</i>	$(F1+F2)/2$
Qualité du processus de production	F1	<i>Facteur</i>	$(F11+F12+F13)/3$
Capacité à estimer	F11	<i>Facteur</i>	$(C1+C2)/2$
Capacité à estimer les délais	C1	<i>Critère</i>	$[(M011+M012)/2+(M013+M014)/2+(M015+M016)/2+(M017+M018)/2]/4$
Capacité à estimer les charges	C2	<i>Critère</i>	$[(M021+M022)/2+(M023+M024)/2+(M025+M026)/2+(M027+M028)/2]/4$
Capacité à produire	F12	<i>Facteur</i>	$(C3+C4)/2$
Maîtrise des délais	C3	<i>Critère</i>	$(M031+M032+M033+M034)/4$
Maîtrise des charges	C4	<i>Critère</i>	$(M041+M042+M043+M044)/4$
Efficacité d'utilisation des techniques	F13	<i>Facteur</i>	$(C5+C6)/2$
Niveau d'utilisation des techniques de production	C5	<i>Critère</i>	$[(M051+(M052+M053+M054+M055))/4+(M056+M057+M058+M059)/4]/3$
Couverture des techniques de contrôle	C6	<i>Critère</i>	$[(M061+(M062+M063+M064+M065))/4]/2$
Qualité intrinsèque du produit	F2	<i>Facteur</i>	$(F21+F22)/2$
Utilisabilité	F21	<i>Facteur</i>	$(F211+F212+F213)/3$
Fiabilité	F211	<i>Facteur</i>	$(C7+C8+C9+C10+C11)/5$
Complexité	C7	<i>Critère</i>	$(M071+M072+M073)/3$
Modularité	C8	<i>Facteur</i>	$(M081+M082+M083)/3$
Taux de défaillance	C9	<i>Critère</i>	M09
Efficacité des tests	C10	<i>Critère</i>	$(M101+M102+M103)/3$
Robustesse	C11	<i>Critère</i>	$(M111+M112)/2$
Conformité	F212	<i>Facteur</i>	$(C12+C13)/2$
Correction du document de spécification des besoins	C12	<i>Critère</i>	M012
Lisibilité et complétude de la documentation externe	C13	<i>Critère</i>	$(M131+M132+M133)/3$
Ergonomie	F213	<i>Facteur</i>	$(C14+C15)/2$
Facilité d'apprentissage	C14	<i>Critère</i>	$(M141+M142+M143+M144)/4$
Facilité d'utilisation	C15	<i>Critère</i>	$(M151+M152+M153)/3$
Maintenabilité	F22	<i>Facteur</i>	$(F221+F222+F223)/3$
Capacité à être porté	F221	<i>Facteur</i>	$(C7+C8+C10+C16+C17+C18)/6$
Indépendance	C16	<i>Critère</i>	$(M161+M162+M163)/3$
Traçabilité	C17	<i>Critère</i>	$(M171+M172)/2$
Lisibilité et complétude de la documentation du développement	C18	<i>Critère</i>	$(M181+M182+M183)/3$
Capacité à être corrigé	F222	<i>Facteur</i>	$(C7+C8+C17+C18)/4$
Capacité à évoluer	F223	<i>Facteur</i>	$(C7+C8+C13+C16+C17)/5$

## ANNEXE 4

**Principales fonctions des AGL utilisés  
dans les organisations étudiées**

	<i>AMC-DESIGNOR</i>	<i>CA-TELON</i>	<i>SYSTEME DELF</i>	<i>ORACLE</i>
Type	Conception	Intégré	Conception	Intégré
Fournisseur	Powersoft	Computer Associates	Delf	Oracle
Version utilisée	4.2	2.0 C	4.1	1.2
Environnement de travail	Windows	OS 2	Windows OS 2	Windows OS 2
Nombre de licences en France	6 000	>100	~ 70	Non communiqué
Méthodes de conception structurée supportées	Merise IE	IE	Merise	IE
Méthodes de conception objet supportées	OMT	-	-	-
Fonction de prototypage	Oui	Oui	Oui	Oui
Interface avec traitement de texte	Oui	Oui	Oui	Oui
Alimentation d'un référentiel central	Oui	Oui	Oui	Oui
Gestion de projets	Non	Oui	Non	Non