



ADELI

La LETTRE n° 48

Juillet 2002



GCL : pour quoi faire, pour éviter quoi ?

(Objectifs, enjeux, contraintes, ...)

Introduction

Développer quelque chose d'original sur la Gestion de Configuration Logicielle est probablement un pari impossible à relever, tant il a déjà été écrit et bien écrit. ADELI a notamment publié dans ses Lettres n^{os} 44 et 45 :

- d'une part, un article critique sur les difficultés quotidiennes à surmonter par les entreprises face à la GCL (ou l'absence de GCL) ;
- d'autre part, un guide méthodologique de GCL en deux volets, resituant notamment la GCL dans le contexte plus global de l'ingénierie des systèmes d'information.

C'est pourquoi on s'efforcera simplement de rappeler ici deux ou trois réalités simples, en abordant le sujet sous des angles différents : celui de l'importance des définitions du « pour quoi faire », et du « pour éviter quoi », celui des « contraintes » impliquées par les choix effectués ; c'est-à-dire toute la question des « enjeux ».

On notera que les illustrations de ces questions se feront plus particulièrement au travers d'exemples pris au sein de grandes organisations, celles-ci étant plus familières à l'auteur ; d'autre part, ces questions ont donné lieu, pour certaines, à divers échanges d'expériences au sein de la Commission « Gestion de Configuration Logicielle » d'ADELI.

Nature de la GCL : rappels

De quoi s'agit-il ?

Il s'agit de gérer – c'est-à-dire acquérir, stocker, historiser, construire, restituer, livrer - tous les composants logiciels utiles, sources et dérivés exécutables, autant que possible, définissant les traitements et les données du système informatique ou d'une partie de celui-ci. Cela, normalement, au moyen d'un « Référentiel » - le gestionnaire de configuration et ses contenus - autrement dit une « armoire électronique organisée, associée à des automates logiciels traitant les échanges utilisateurs ». Nous traitons bien entendu des logiciels utilisés pour la gestion au sens le plus large.

Composants logiciels : quels sont les types de composants logiciels gérés ? Tous : ceux définissant les traitements peuvent être des programmes batch et transactionnels, des JCL et procédures cataloguées, des descriptions d'états et d'écrans, etc., ceux définissant les données peuvent être des descriptions des structures des données au travers des copys, DDL de SGBD etc.

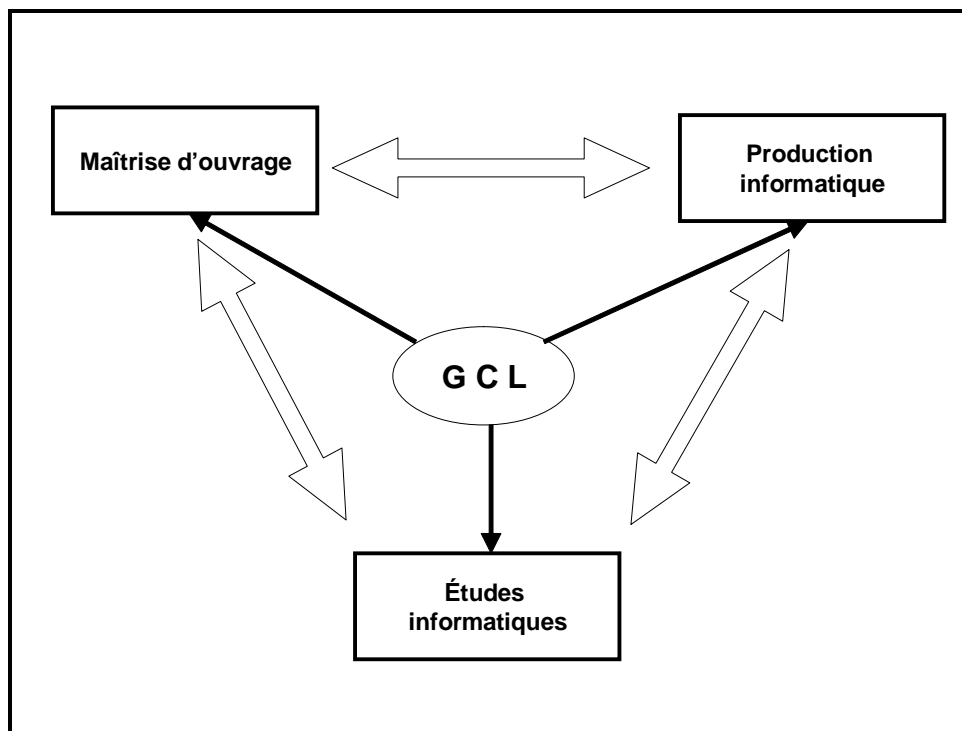
À minima, cette gestion des composants peut être rudimentaire, implicite, empirique : sauvegarde sommaire par les développeurs de leurs sources, à une date donnée, sur un support de stockage affecté à cet effet : disque, bande, voire disquette... Sauvegarde des exécutables à une date donnée par la production. Même une procédure aussi simplifiée, si elle est faite proprement, méthodiquement, et documentée (à minima des étiquettes claires) peut tenir lieu d'embryon de GCL tout à fait acceptable.

Propriété et maîtrise du logiciel : il est à noter que, de plus en plus, les composants informatiques ne sont plus considérés comme la propriété et la maîtrise du seul service informatique mais d'abord comme la propriété et la maîtrise de l'entreprise et ses métiers, représentées par la maîtrise d'ouvrage d'utilisateurs et, dans une moindre mesure, sa production et ses études : c'est une évolution capitale.

Acteurs et aspect contractuel de la GCL : On trouve alors trois types d'acteurs majeurs autour de la GCL :

- la maîtrise d'ouvrage (client, donneur d'ordre) ;
- les études informatiques (architecte, maître d'œuvre, fournisseur) ;
- la production informatique (également pour partie fournisseur, architecte technique et maître d'œuvre).

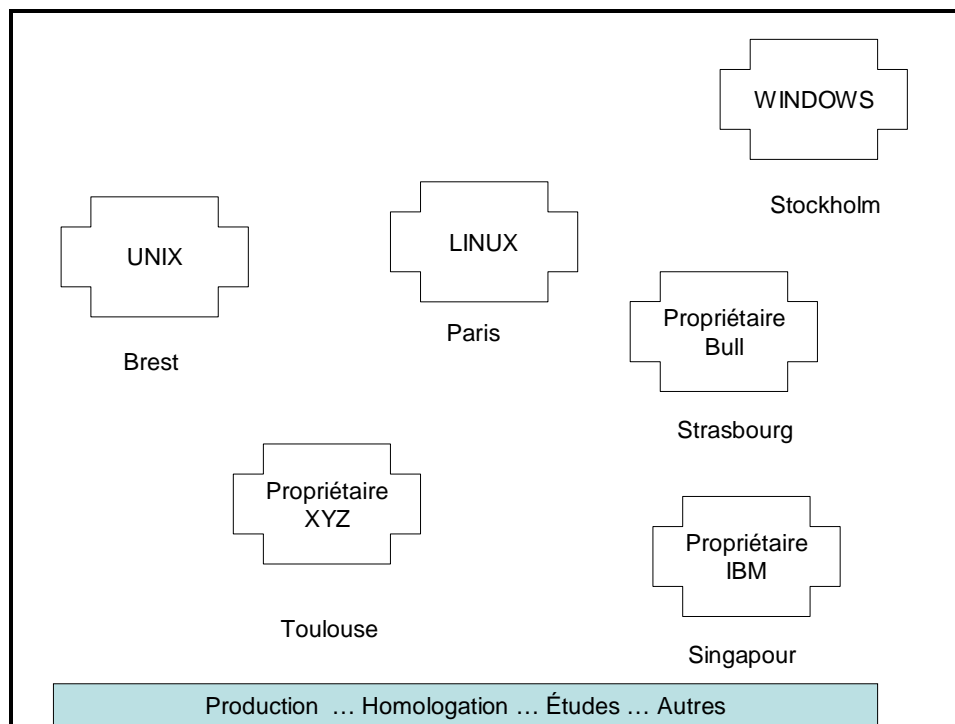
Chacun peut représenter, de plus, des centres de profits, des entités économiques, voire des entités juridiques indépendantes, voire concurrentes, dotées chacune, dans le cadre général, de finalités particulières, la GCL prend donc dès lors un rôle très contractuel.



Organisation

Hétérogénéité et dispersion des plates-formes : On notera que la variété, l'hétérogénéité et la dispersion des plates-formes matérielles et logicielles, utilisées dans l'entreprise, pourront accroître fortement les enjeux de la mise en place d'un ou de plusieurs gestionnaires de configurations logicielles, mais aussi leurs complexités et contraintes. Cela peut faciliter dans certains cas les déploiements (mises en production simultanées sur plusieurs sites et plates-formes). Mais cela pose en particulier la question du choix des plates-formes sur lesquelles seront mis en place les outils de GCL et, en conséquence, celle de l'interfaçage, soit des outils entre eux, soit de ceux-ci avec l'ensemble des plates-formes de développement, de recette, de production.

Gestion des composants matériels : les composants matériels, leurs assemblages et leurs évolutions peuvent-ils être gérés au sein de la GCL, comme les composants logiciels ? Cette question importante, qui donne lieu à discussions, semble trouver une double réponse. D'une part, la GCL gérant, comme c'est sa vocation, les différentes versions d'un logiciel, lorsque ces versions sont associées à différentes plates-formes de matériels, systèmes et langages, il est indispensable que ces versions y soient naturellement clairement répertoriées. D'autre part, chaque type de plate-forme, concerné doit-il être considéré comme composant à gérer à part entière dans la GCL. Rien ne l'empêche effectivement et cela est bien souvent souhaitable, même si cette pratique n'est pas encore courante.



Plates-formes informatiques

De quoi ne s'agit-il pas ?

Rappeler, paradoxalement « à quoi la GCL ne sert pas » est nécessaire pour dissiper les confusions.

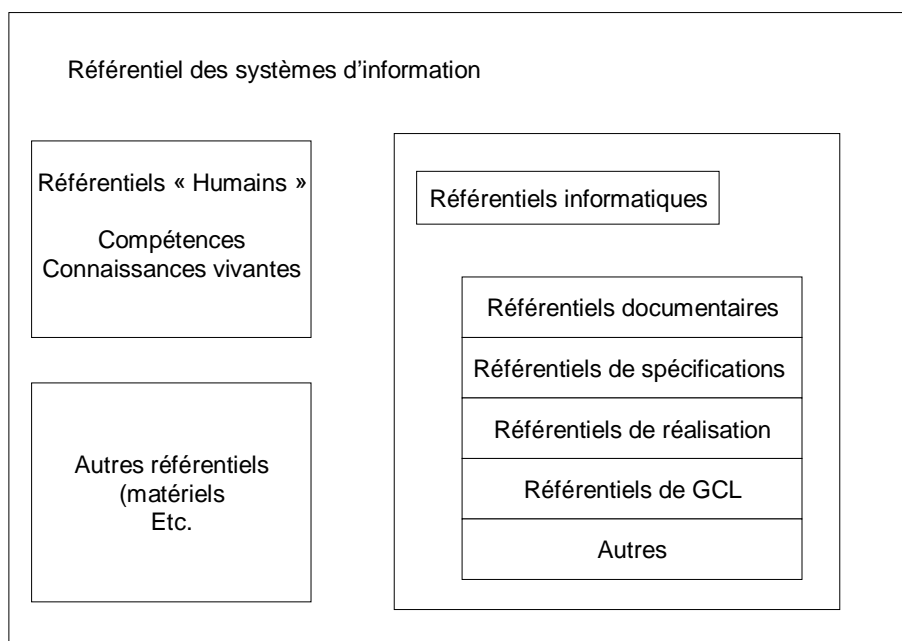
Notons d'abord que nous ne traitons pas ici des logiciels strictement industriels, qui sont soumis à des enjeux, des exigences, contraintes et compétences débordant de notre cadre d'informatique de gestion.

Tendre vers la Maîtrise du SI : bien que ce ne soit pas notre sujet central, rappelons que l'entreprise, son système d'information et son système informatique doivent, tout d'abord, être en priorité connus et aptes à être maîtrisés. Pour cela, il existe, entre autres, la **connaissance humaine** vivante, la **connaissance écrite** dont la **connaissance informatisée** contenue sur des supports référentiels. Ce souci de maîtrise du SI n'est couvert que très partiellement par la GCL : en particulier pour les aspects informatiques physiques (non fonctionnels). Pour être connus et maîtrisés, les différents aspects de l'entreprise ou organisation doivent être, hormis la place centrale de la problématique humaine, prioritairement bien documentés.

Outils AGL et documentaires : Cette documentation peut être supportée utilement (mais supportée seulement) par des outils électroniques de type « gestion documentaire » (adaptée à une documentation faiblement structurée et la gestion historique des versions), ou bien de type dits « AGL » (mieux adaptée à une documentation fortement structurée du SI). Ces deux types de contenants électroniques nécessiteront, de toute façon, un suivi systématique, depuis la direction générale jusqu'à chaque équipe responsable.

Notion de Référentiel : Nous avons utilisé, ci-dessus, le terme AGL pour désigner les étages permettant la conception et la réalisation du logiciel. Pour être plus rigoureux, il faudrait, comme on l'a vu dans les guides méthodologiques publiés par ADELI, considérer l'atelier de génie logiciel de façon plus globale, c'est-à-dire comme le « Référentiel » du SI (et automates d'échanges associés), et donc considérer les outils de GCL comme l'une des composantes de l'AGL ou du Référentiel global, comme le sont les autres outils de conception, de réalisation, de documentation etc.

Dans l'idéal, notons qu'un référentiel informationnel global de l'entreprise devrait contenir l'exhaustivité des documents décrivant l'entreprise et ses SI et, au minimum, tous les documents, contractuels ou non, tant amont qu'aval, décrivant les projets d'évolution et de maintenance du (ou des) système(s) d'information.



Organisation

Place des Tests : Il est clair que la Gestion de Configuration Logicielle elle-même « couvre » bien toutes les étapes du cycle de vie du logiciel, même si certaines de ces étapes s'effectuent principalement hors des frontières de l'outil de GCL. Il semblerait, par contre, intéressant de noter le rôle, pour le moment modeste, de la GCL par rapport aux tests : elle permet avant tout d'identifier avec précision (en développement informatique, en recette utilisateur comme en production...) quels sont les composants testés et leur version, ce qui n'est déjà pas négligeable puisqu'il y a la diminution d'un important facteur d'erreur. Mais la GCL ne sera pas d'un grand secours pour la constitution des jeux d'essais, pour la rédaction des scénarios, la consignation de leurs résultats, ni pour le stockage et l'archivage des données de tests proprement dites, pour lesquelles il convient de trouver des moyens plus adaptés et moins coûteux.

La documentation électronique du test reste, par contre, cruciale, via l'alimentation, contrôlée en quantité et en qualité, d'un dossier de modifications et d'un dossier de tests structurés et précis.

La structure actuelle des outils de GCL ne permet guère d'intégrer avantageusement ce type de document en son sein, qui trouveront mieux leur place, par exemple, sur le gestionnaire documentaire, mieux adaptés au format concerné (Word, Excel etc.) l'outil de gestion de configuration pouvant par contre porter sur ses versions de composants ou lots de livraison les références exactes permettant de localiser et d'extraire instantanément les dossiers concernés.

Sorties à obtenir de la GCL

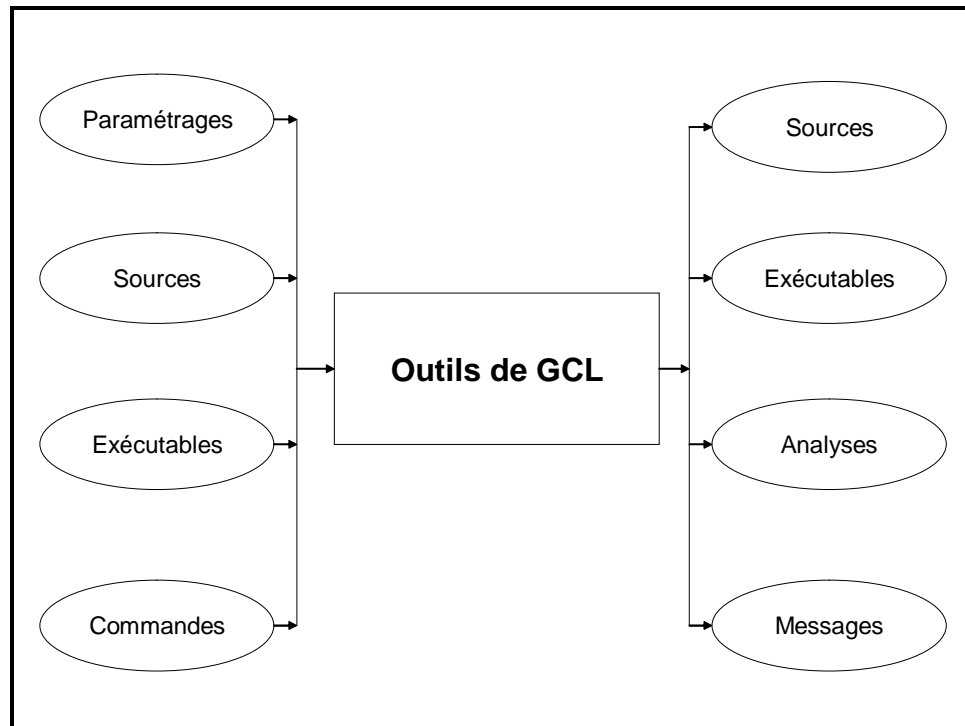
Quelles seront les sorties attendues de la GCL, pour reprendre la démarche claire, chère à Warnier ?

Sources : on souhaite, premièrement, retrouver de façon ordonnée, ce qui est la moindre des choses, les programmes sources que l'on a stockés, mais cela dans leurs différentes variantes de versions référencées. On doit aussi pouvoir faire évoluer le cycle de vie de ces versions de composants sources, c'est-à-dire modifier leur statut en les faisant passer, par exemple, de l'étape cycle de vie de tests informatiques à celle de recettes utilisateurs, ensuite de qualification informatique ou de production, pouvoir effectuer des reports de modifications ou fusions de versions.

Exécutables : on souhaite, deuxièmement, pouvoir produire, générer, construire et livrer (installer) les exécutables, à partir des sources dans leur version la plus récente ou une version plus ancienne. (Notons que l'historisation des composants exécutables n'est pas une nécessité, ceux-ci étant dérivés, donc reconstituables à partir des versions de sources historisées.)

Rapports d'Analyses : on souhaite, troisièmement, pouvoir effectuer et produire toutes sortes d'analyses des composants, des plus simples aux plus complexes. Comme, par exemple, des historiques de modifications, des comparaisons de versions ou de composants, des analyses d'impacts entre composants en cas de modifications, des références croisées entre les différents composants et différents types de composants, des statistiques de mises à jour, ou tout simplement retrouver quand a été effectuée une modification, comment elle a été faite (son contenu) et par qui, avec possibilité de retour arrière, si nécessaire, c'est-à-dire d'annulation de la modification et de retour à une version précédente des logiciels.

Messages : enfin, le système de GCL doit nous fournir naturellement, à chaque instant, des messages nous informant précisément du résultat des commandes accomplies : résultats de bonne exécution, d'avertissements (warnings) ou encore d'erreurs bloquantes.



Outils de GCL

Entrées à fournir à la GCL

Quelles seront les entrées à fournir à la GCL ?

Paramétrages : les premières entrées à fournir au système sont constituées par un paramétrage important du système de GCL : cycles de vie du logiciel choisis et contrôles de validation, demandés dans ce cycle, organisation en arborescence des domaines applicatifs et applications et dépendances entre ceux-ci, types de composants logiciels gérés et procédures d'analyses, de dérivations et de compilations associées à ces types, procédures d'installations et de livraisons, normes à mettre en places, autorisations et interdictions, etc. Il faut en même temps pouvoir définir et organiser sur le pourtour de la GCL, celle-ci n'en constituant que le noyau, les environnements d'exécution (batch et TPs) à mettre en place correspondant à chacune des étapes du cycle de vie et pour les besoins de chaque équipe de projets.

Sources : les deuxièmes types d'entrées à fournir sont les « sources » lors de l'alimentation initiale comme lors de chaque évolution. Ces entrées doivent être ordonnées par type de source (types de programmes et de langages, procédures, JCL, descriptions de données, etc.), par applications destinataires, par label de version (par exemple, demande de modification no 1345 du 10 août, Projet Euro) et par configurations (production, recette, tests informatiques, etc.). L'alimentation initiale est bien sûr à effectuer avec un soin particulier, mais sera simplifiée par le fait que l'on ne devrait pas reprendre d'historique, mais uniquement une version initiale ; cela peut être l'occasion d'une légère réorganisation, les plus grandes réurbanisations étant à effectuer de préférence ultérieurement.

Exécutables : troisièmement, les exécutables systématiquement, autant que possible, dérivés (compilés) à partir des sources de GCL, seront donc des résultats réinjectés en entrée de cette GCL au fur et à mesure de leur production, pour y être stockés.

Commandes : enfin, il convient, naturellement, d'entrer pour chaque sollicitation, les commandes permettant d'effectuer les actions souhaitées, celles-ci pouvant être partiellement cataloguées par le paramétrage ou accessibles par le biais de menus conviviaux. La convivialité n'exclura pas la nécessité d'une formation, d'une information et d'une bonne documentation du système de GCL et son paramétrage.

Facteurs exclus des entrées : notons qu'il est deux « pseudo entrées » que l'on ne considère pas, mais que l'on peut, bien sûr, prendre en compte comme facteurs intervenant dans les résultats :

- le **logiciel de GCL** qui conditionne naturellement, à la base, le résultat, et peut évoluer, du fait du fournisseur, dans le cas de passage d'une version à une autre, parfois à la surprise de l'utilisateur.
- L'**environnement** : on peut nommer ainsi l'ensemble des facteurs considérés « extérieurs » à la problématique, mais pouvant influencer indirectement sur les résultats, deux exemples extrêmes pouvant en être les compilateurs installés par le constructeur et l'alimentation en courant électrique (dont il est « probable » que le dysfonctionnement puisse perturber l'obtention du résultat souhaité !)

Transformations de la GCL : obtention des sorties par les entrées

Quelques mots, enfin éventuellement, dans ce chapitre, si cela représente toutefois un intérêt pour le lecteur sans le noyer dans des détails inutiles – sinon qu'il passe au chapitre suivant - sur l'obtention des sorties à partir de la transformation des entrées.

Les deux types d'entrée que constitueront le **paramétrage**, d'une part, et la **commande** lancée, d'autre part, étant dans tous les cas les deux facteurs primordiaux qui conditionneront la restitution des différents types de sorties suivants, nous omettrons donc de les citer une nouvelle fois ci-dessous.

Sources : quand l'on extrait un composant source en sortie de l'outil de GCL, le source restitué est le plus souvent... celui qui a été précédemment fourni en entrée, à une date donnée, à ceci près que l'outil de GCL est capable de reconstituer l'image du source à la date donnée à partir de son image initiale et des différents deltas de modifications qui seuls ont été stockés.

De plus, cela concernant aussi le paragraphe sur les analyses, des rapports d'analyses permettent de mettre en évidence un historique des différentes mises à jour, une comparaison entre plusieurs sources, des références croisées, etc., etc.

Exécutables : il y a deux façons - deux vues - pour présenter et expliquer l'obtention des composants exécutables en sorties de la GCL ; la plus claire mais la moins rigoureuse sera de dire qu'un programme exécutable obtenu en sortie est obtenu à partir de la compilation, avec des compilateurs adaptés aux langages concernés, de la combinaison des sources exigées en entrée (combinaison de sources pouvant se justifier par de nombreux appels de type copy, de type sous-programme, etc.) ; la vue la plus rigoureuse et fidèle à la réalité expliquera que l'exécutable obtenu en sortie est le même exécutable qui a été acquis en entrée.

En effet, en réalité, l'outil lui-même ne contient ni compilateur, ni pré-compilateur ni d'éditeur de liens, bien qu'il puisse en donner l'illusion à l'utilisateur ; mais il agira, de façon cachée, le plus souvent, en trois temps :

- exportation sur la machine, dans un fichier temporaire, des sources nécessaires à la compilation ;
- déclenchement de la mise en œuvre des compilateurs adéquats ;
- importation, à partir d'un nouveau fichier, temporaire ou non, des exécutables obtenus.

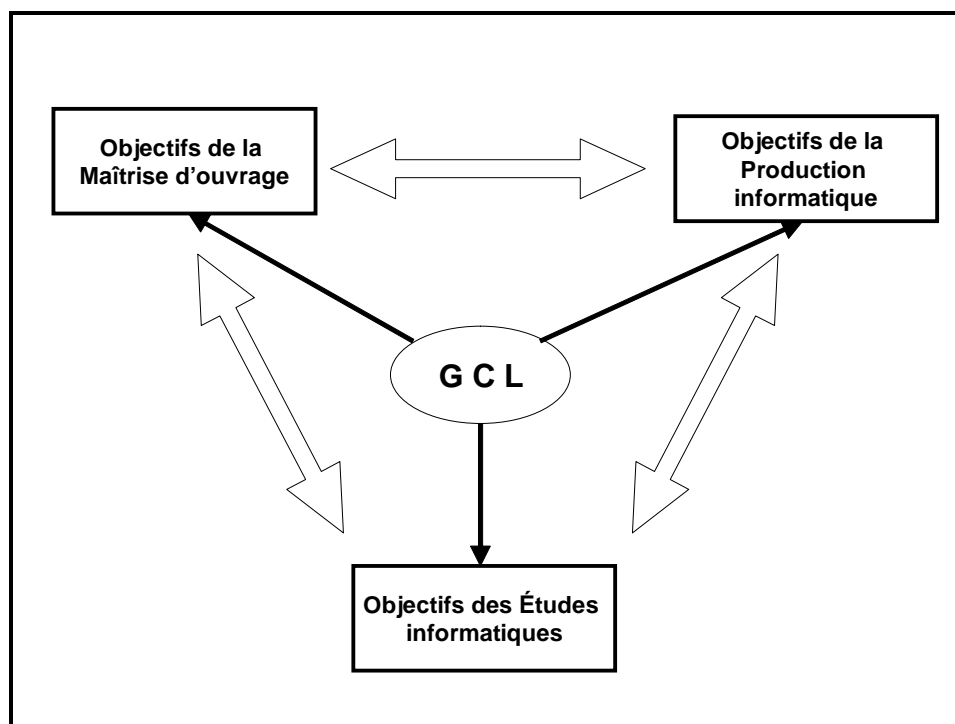
La première vue, plus simplificatrice, peut être néanmoins grossièrement acceptable, auquel cas il faudrait alors considérer tous les compilateurs (au sens le plus large) et procédures de compilations nécessaires comme constituant des entrées à fournir à l'outil de GCL faisant pleinement partie du paramétrage. Toutefois, les fournisseurs étant dissociés, la responsabilité économique et juridique est également dissociée en cas de dysfonctionnement.

Analyses : les rapports d'analyses sont bien sûrs, souvent, des combinaisons complexes, à défaut d'être compliquées, de nombreux éléments d'entrée, sources en particulier, exécutables quelquefois, et enfin de résultats intermédiaires éventuellement stockés résultants de requêtes précédentes. Ce sujet est, bien sûr, si vaste qu'il pourrait justifier, à lui seul un article, sinon un livre.

Messages : enfin, les messages nous informant précisément du résultat des commandes accomplies - bonne exécution, avertissement (warning) ou erreurs bloquante - seront fonction : du logiciel, de l'environnement (à ne pas négliger), et de généralement tous les types d'entrées : paramétrages, commandes et sources ou exécutables.

Un source non trouvé à l'endroit indiqué par l'utilisateur déclenchera, par exemple, un message d'erreur bloquante, alors que toutes les conditions normalement réunies déclencheront au contraire le message de fin de bonne exécution.

Quels types d'objectifs et d'enjeux ?



Objectifs globaux de l'organisation

Objectifs généraux de la GCL

Les principaux objectifs d'une gestion de configuration logicielle, pour une direction d'entreprise, concernent, tout d'abord, la traçabilité intégrale des évolutions du système informatique, comme on l'a déjà vu dans les articles précédents, traçabilité comparable, par exemple, à celle d'un laboratoire pharmaceutique ou d'une industrie alimentaire. Cela pour l'obtention d'une meilleure fiabilité des modifications, une plus grande maîtrise et plus de sécurité.

Le système informatique passe ainsi du stade artisanal au stade industriel, d'« atelier à l'usine » dont tout dysfonctionnement est surveillé, tracé, puis systématiquement et méthodiquement corrigé.

Un autre enjeu central peut-être aussi la réutilisabilité des composants. En premier lieu essentiellement pour les études informatiques, et plus rarement mais parfois pour les autres types d'acteurs tels la maîtrise d'ouvrage utilisateur. Cette réutilisabilité est toutefois limitée aux composants physiques, puisque pour ce qui concerne les composants fonctionnels, cela nécessite une autre démarche complémentaire, beaucoup plus ambitieuse, évoquée plus haut, qui n'est pas pour le moment dépendante de la GCL telle qu'elle est considérée le plus couramment.

Examinons à présent des enjeux plus spécifiques à chacun des trois corps de métiers cités.

Objectifs de la GCL pour la production informatique et l'industrialisation

Les responsables de la production informatique de l'entreprise (externalisée ou non) souhaiteront avoir un regard, sinon la maîtrise globale, de la qualité en terme de performances (rapidité notamment) de fiabilité et de robustesse, pour tout ce qui est exploité tous les jours en production, ainsi que les évolutions et corrections effectuées ; ils souhaiteront avoir la maîtrise maximum de la gestion des incidents et pouvoir se montrer compétents et réactifs face à toute sollicitation, tout projet, et face à tout problème.

Utilisateurs et Informaticiens sont bien évidemment intéressés eux aussi, par l'atteinte de ces objectifs, mais plus indirectement.

Objectifs de la GCL pour la maîtrise d'ouvrage et l'homologation / recette

Les acteurs de la maîtrise d'ouvrage et l'homologation / recette veulent être certains de la nature des objets qu'ils sont en train de réceptionner, tester, homologuer, comme de la nature et l'échéancier précis des livraisons passées et futures, tant en recette qu'en production, afin de pouvoir maîtriser la satisfaction des besoins des acteurs de l'entreprise et les sollicitations des opérationnels, sans éprouver le sentiment d'être « menée en bateau » par leurs fournisseurs techniques (comme, par ailleurs, par leurs clients utilisateurs).

Objectifs de la GCL pour les études informatiques et la maintenance

Les chefs de projets des études informatiques et de la maintenance, souhaiteront pouvoir suivre très précisément, a posteriori au moins, l'évolution des composants, et pouvoir effectuer les études d'impacts physiques nécessaires des évolutions possibles.

Les développeurs souhaiteront avoir à leur disposition une vision autonome, complète et cohérente de l'ensemble des composants et, si nécessaire, de leur historique, sans avoir besoin de solliciter de façon excessive les collègues ou les responsables et sans être contraints à des fouilles archéologiques dans des armoires et dossiers ; chacun d'eux souhaite mieux maîtriser son travail et avoir une vision la plus globale et la plus précise possible de la réalité.

Relativité et variabilité des objectifs en fonction des contextes

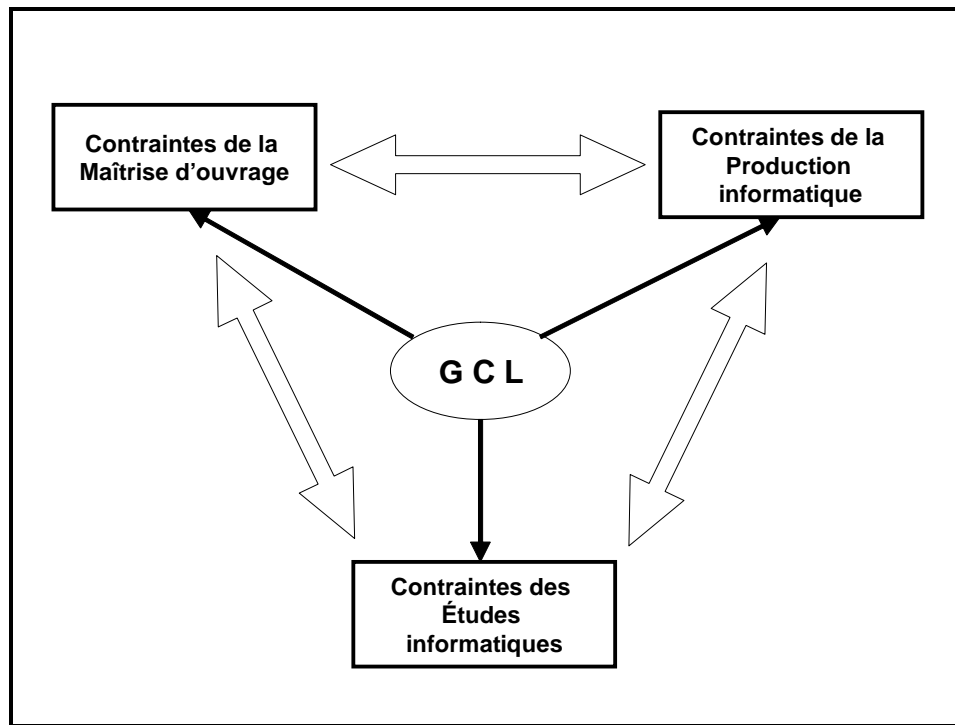
Périmètres : Complémentairement et symétriquement à ces objectifs, en examinant au départ le paragraphe concernant le « de quoi ne s'agit-il pas ? », nous avons ainsi implicitement défini les « non objectifs » de la GCL, et donc le périmètre de ce que nous pouvons et ne pouvons en attendre.

Mais en quoi ce que l'on peut attendre effectivement de la GCL peut-il fortement varier d'une organisation à l'autre ?

Tailles : il est clair, par exemple, qu'un artisan boulanger ayant informatisé sa comptabilité risque d'avoir des besoins en GCL plus modestes que le grand distributeur, le premier pouvant se contenter de bien étiqueter et d'archiver toutes ses disquettes et cédroms d'installation, le second ayant le plus souvent besoin d'un outil plus spécialisé.

Organisations : les quatre types d'objectifs énoncés ci-dessus vont donc sensiblement varier selon la nature de l'entreprise, son volume, les contraintes de son organisation et de sa culture. La nature des relations qui unissent et séparent la direction, les maîtrises d'ouvrage utilisatrices, la production informatique et les études, va jouer un rôle capital. On peut, par exemple, imaginer une entreprise de taille humaine où ces quatre métiers sont très intégrés, le mode de relations très direct et informel, et inversement, comme c'est de plus en plus souvent le cas, une très grande structure dont les métiers sont économiquement - voire juridiquement - séparés, constitués de multiples clients et fournisseurs mutuels, dont les rapports sont fortement contractualisés, soumis même à concurrence. Même si ce n'est pas le cas, l'éloignement géographique et culturel va entrer de plus en plus en ligne de compte, l'utilisateur pouvant être en Espagne, en Angleterre, en Allemagne..., la production à Poissy ou à Nantes, et les développeurs à Paris, New-York, Stockholm ou New Delhi (ce dernier cas devenant fréquent lors de l'utilisation de très nombreux progiciels).

Quels types de contraintes ?



Contraintes globales de l'organisation

Contraintes générales de la GCL

Les principales contraintes d'une gestion de configuration logicielle, pour une direction d'entreprise, seront en particulier **l'importance de l'effort humain et financier** à investir pour structurer et industrialiser la gestion des composants logiciels sources et exécutables, selon le point de départ de l'entreprise, la plus grande rigidité nécessaire des méthodes de travail et des calendriers de livraison des évolutions, offrant aussi plus de souplesse, tout en gérant naturellement une procédure d'urgence mais qui deviendra désormais, de par la règle, l'exception.

La **contrainte culturelle** est la forte à franchir qui implique partage du savoir et travail coopératif.

La contrainte de **l'hétérogénéité** et de la **dispersion** géographique des **plates-formes** matérielles et logicielles et de la complexité des **réseaux**.

Quelles sont les contraintes plus spécifiques à chaque corps de métiers ?

Contraintes de la GCL pour la production informatique et la qualification

Contraintes : la production devra d'abord empêcher - si nécessaire par un blocage de livraison voire un retour à la situation précédente - toute détérioration du service qu'elle offre à l'entreprise, dans le cas où elle serait tentée d'accepter une livraison logicielle défectueuse, sous la pression des études et de la maîtrise d'ouvrage.

Elle devra pouvoir ensuite garantir l'amélioration des performances en minimisant ses coûts, en demandant et obtenant les moyens nécessaires. C'est elle qui sera garante, au jour le jour, de la qualité de la logistique informatique de l'entreprise, et, donc par là même, de sa sécurité, et bien souvent de sa survie.

Rôle : la production devra donc jouer à la fois un rôle de frein de l'évolution pour maintenir la qualité de son service, et en même temps de moteur pour pouvoir améliorer son offre, ses activités et pouvoir suivre l'évolution des environnements économiques et techniques ainsi que l'état de l'art de la concurrence.

Exigence de la GCL : La GCL devient donc souvent pour l'exploitation, sous l'impulsion de la direction générale, un impératif de qualité et une exigence, imposée si nécessaire aux études informatiques et à la maîtrise d'ouvrage qui devront alors se soumettre à ses contraintes.

Contraintes de la GCL pour la maîtrise d'ouvrage et l'homologation

Contraintes : la maîtrise d'ouvrage doit, elle aussi, empêcher une régression des fonctionnalités du service, apportées au quotidien aux métiers concernés de l'entreprise, pouvoir identifier clairement les origines des dysfonctionnements ou insuffisances des logiciels pour pouvoir en rendre compte à ses propres clients en énonçant les solutions qui seront apportées et leur échéancement. Elle doit en même temps maîtriser, autant qu'elle le peut, les évolutions du système d'information en étant à même d'améliorer l'offre de service, apportée à l'entreprise et ses acteurs dans des délais et des coûts les plus acceptables possibles, quels qu'en soient les fournisseurs (études informatiques ou non).

Elle est, face aux utilisateurs finaux, la garantie de la nature et de la qualité fonctionnelles de ce qui est exploité, et exigera souvent une maîtrise globale de ce qui est livré en production et des performances offertes aux utilisateurs.

Rôle : son rôle sera donc à la fois moteur, en ce qui concerne l'évolution des fonctionnalités logicielles et des performances matérielles – en négociation constante avec ses clients et ses fournisseurs - et, dans une moindre mesure, frein en ce qui concernerait un risque de détérioration potentielle, par des aménagements non maîtrisés, de l'outil quotidien de l'entreprise.

Contrôle a posteriori : L'outil de GCL peut permettre, à la maîtrise d'ouvrage, de contrôler a posteriori l'absence de « livraisons et évolutions clandestines » en production et en recette, dans la limite de sa connaissance des composants physiques qui restera modeste et des explications qu'elle obtiendra des études, ainsi qu'un contrôle du respect, par les études, du calendrier des livraisons.

La maîtrise d'ouvrage subit aussi néanmoins les contraintes de la GCL dans la mesure où cela l'oblige à ne pas déroger – sauf urgence impérative - au calendrier des livraisons, permis par la production.

Contraintes de la GCL pour les études informatiques et la maintenance

Contraintes : l'équipe des études informatiques doit satisfaire les demandes de corrections et améliorations du système informatique pour justifier son rôle, vis-à-vis de la maîtrise d'ouvrage en premier lieu. Elle est de plus en plus placée, elle aussi, en situation de concurrence potentielle, vis-à-vis de ses partenaires qui peuvent élaborer sans elle des solutions (bureautique, progiciels...) ou, plus souvent encore, vis-à-vis d'autres acteurs possibles effectuant le même métier qu'elle (appels d'offres).

Rôle : les études informatiques vont donc souvent se trouver en situation de pousser en avant la livraison et la mise en place d'évolutions, sous la pression de la maîtrise d'ouvrage, tout en risquant la sanction lorsque les résultats ne correspondent pas aux attentes ou, a fortiori, produisent des régressions de services. Les études s'efforceront alors de temporiser et grouper les livraisons demandées par la maîtrise d'ouvrage – rôle de frein - pour ne pas détériorer les systèmes en place et ne pas risquer de décevoir et de mécontenter ses clients par des actions mal maîtrisées.

Exigences de la GCL : L'équipe des études informatiques, si elle est quelquefois à l'initiative de la mise en place de l'outil de GCL, se voit, le plus souvent, imposer cet outil comme une nécessaire industrialisation de ses méthodes de travail jugées trop artisanales. Une fois ce type d'outil installé, les études vont chercher à en tirer, de façon inventive, le meilleur parti possible au niveau de leur travail, de façon à pouvoir effectuer plus efficacement études d'impacts, contrôle des évolutions, etc.

Relativité et variabilité des contraintes en fonction des contextes

Périmètres : On a vu que les contraintes globales ne sont pas du tout les mêmes selon la nature et l'identité de l'entreprise elle-même : un artisan boulanger n'aura probablement ni les mêmes contraintes, ni les mêmes besoins, qu'un grand distributeur. Si tant est qu'une problématique de GCL soit véritablement pertinente dans le premier cas et qu'une boîte de disquettes vierges et un carton des cédéroms d'installation ne suffisent à répondre aux besoins, les contraintes pouvant se borner à sauvegarder tous les deux ans une version de logiciel et conserver logiciels et matériels à l'abri de la farine et de l'humidité.

Efforts Humains et financiers : l'effort humain et financier peut être très différent selon le point de départ de l'entreprise, sa taille, les ressources que l'entreprise est prête ou non à mettre en œuvre, le degré de motivations des directions et des équipes, etc.

Contraintes culturelles : Il s'agit là de freins éventuels ou non des mentalités, selon les entreprises, à la mise en œuvre de la GCL, et de la possibilité ou non de les compenser par de l'information et des échanges de communication et de formation. On se rappelle que, par le passé, il s'agissait là d'un des premiers freins, par exemple, à l'utilisation de la démarche fonctionnelle « objet » et des technologies de développement « objet » pour le développement de la réutilisabilité du logiciel, thèmes dont on nous avait pourtant abreuvés jusqu'à saturation complète pendant des années. Il faut aussi prendre en compte l'historique propre de chaque équipe, et la problématique des restructurations, fusions et réorganisations régionales ou internationales, perturbant parfois les habitudes diverses, dont l'utilisation des langues.

Hétérogénéité et dispersion des plates-formes : Les contraintes ne seront bien entendu pas les mêmes selon la complexité et la diversité des plates-formes matérielles et logicielles et des réseaux, voire selon leurs contraintes de répartition géographique, encore que cette dernière contrainte tende à s'estomper grâce aux télécommunications.

Organisation générale : Les contraintes seront enfin différentes selon l'organisation générale propre à l'entreprise, et en premier lieu selon le mode de communication plus ou moins facile ou non, fluide ou non, formalisé ou non formalisé, entre les rôles des maîtrises d'ouvrage, des études informatiques et des productions informatiques.

Conclusion

L'un des buts de cette présentation était donc principalement d'esquisser un panorama, lorsque cela est pertinent, de quelques enjeux les plus cruciaux comme de quelques contraintes majeures de la Gestion de Configuration Logicielle, pour l'entreprise et pour ses différents types d'acteurs.

Il était également de mettre en évidence les aspects pouvant être les plus fortement communs d'un contexte à l'autre, comme les aspects pouvant être les plus fortement variables, selon les entreprises et organisations concernées.

Cela nous permettra d'être à même, lorsque le besoin existe, d'esquisser, une analyse pertinente des besoins de chaque organisation et de ses contraintes, avec les acteurs concernés, et être à même de suggérer ou préconiser les pistes d'évolutions ou de transformations les mieux adaptées au contexte, à son histoire et à ses projets et devenirs possibles.

Cela nous permettra peut-être aussi de jeter des bases pertinentes vers l'établissement futur d'une classification des enjeux, des contraintes et des solutions possibles de la GCL – que ces solutions soient logicielles ou non - permettant de constituer l'une des pièces d'un « Scope » d'ADELI, même si ce n'est là qu'un des multiples aspects, parmi d'autres, du sujet éventuel d'un tel ouvrage.

Pierre Fischhof,
Consultant en Migrations des Systèmes Informatiques,
membre de la Commission GCL de l'ADELI,
pierre.fischhof@libertysurf.fr

Glossaire lié à la gestion de configuration

Configuration (de configurare : donner une forme)

Forme extérieure d'un ensemble, relief ; aspect général de quelque chose d'abstrait, caractère d'ensemble, physionomie. [GL]¹

Description des éléments d'un système informatique précisant les différentes unités et leur liaison. [GL/Informatique]

Une configuration constitue la référence d'une nomenclature d'un système, à un instant donné, en indiquant la liste des références de ses articles constitutifs.

Une configuration se caractérise par :

- une identification ;
- une description ;
- une liste de composants (chaque composant ayant un indice de révision) ;
- une disposition relative de ces composants.

[ADELI Lettre 44]

Référentiel de configuration

Ensemble d'articles de configuration (documents et composants logiciels) ayant fait l'objet d'une revue formelle et d'un consensus et qui, par la suite, sert de base à un développement ultérieur et ne peut être modifié que par l'utilisation de procédures de contrôle de changement, officielles. [SCTC2]

Configuration de référence

Configuration d'un produit, formellement établie à une étape de la vie du produit et servant de référence pour les activités ultérieures. [ISO 10007]

Article de configuration

Ensemble de matériels ou de logiciels (ou des deux) définis aux fins de la gestion des configurations et traités comme une entité unique dans le processus de gestion des configurations (IEEE-STD-6 10 Glossaire). [SCTC]

Gestion de configuration

Discipline de management de projet qui permet de définir, d'identifier, de gérer et de contrôler les articles de configuration tout au long du cycle de développement d'un logiciel (ISO 1007).

Elle consiste à mettre en œuvre et à utiliser des moyens organisationnels, techniques et administratifs pour obtenir, à tout moment de son cycle vie, une visibilité satisfaisante du produit, au travers de ses caractéristiques techniques et fonctionnelles, généralement décrites dans des documents.

La gestion de configuration est :

- soit dynamique ou adaptée au développement : elle permet, essentiellement, de coordonner les activités de développement des équipes, de suivre chaque composant logiciel au cours de son élaboration jusqu'à sa validation ;
- soit statique ou adaptée à la maintenance : ... elle est fondée sur la traçabilité des composants d'un système en production dont elle contrôle les modifications. [ADELI-Lettre 44] 3

¹ [GL] *Grand Larousse en 5 volumes.*

² [SCTC] *Secrétariat du Conseil du Trésor du Canada : www.cio-dpi.gc.ca*

³ *Adaptation du texte initial d'Alain Coulon : Petit guide méthodologique (1^{ère} partie).*

Gestion de configuration logicielle

La gestion de configuration logicielle (GCL) consiste à établir et à maintenir l'intégrité des produits du projet logiciel tout au long du cycle de vie logiciel du projet. Elle consiste à identifier la configuration (c'est-à-dire les produits de travail logiciels sélectionnés et leur description) à des points déterminés du temps, à contrôler les changements apportés à la configuration, et à maintenir son intégrité et sa traçabilité tout au long du cycle de vie logiciel.

Parmi les produits de travail placés sous gestion des configurations logicielles, se trouvent les produits livrés au client ainsi que les éléments nécessaires à la création de ces produits logiciels ou identifiés avec ceux-ci. [SCTC]

Produit de travail logiciel

Élément quelconque créé dans le cadre de la définition, de la maintenance ou de l'utilisation d'un processus logiciel, y compris les descriptions, les plans, les procédures, les programmes informatiques et la documentation connexe, qu'il soit prévu ou non de livrer l'élément en question au client ou à l'utilisateur final. [SCTC]

Gestion logicielle intégrée

La gestion logicielle intégrée vise à intégrer les activités d'ingénierie et de gestion logicielles sous forme d'un processus logiciel défini et cohérent qui est adapté à partir du processus logiciel standard de l'organisation et des acquis de processus connexes.

Le processus logiciel défini du projet est une version du processus logiciel standard de l'organisation, adapté pour traiter les caractéristiques particulières du projet. [SCTC]

Changement (de cambiare : troquer)

Action, fait de changer, de modifier quelque chose, passage d'un état à un autre ; Modification profonde, rupture de rythme. Tout ce qui rompt les habitudes, bouleverse l'ordre établi. [GL]

Conduite du changement

Transformer l'entreprise pour développer ses performances implique la mise en cohérence des hommes avec l'organisation et les technologies. La conduite du changement permet de vaincre les réticences, susciter l'adhésion et convaincre, afin que les comportements soient adaptés aux évolutions et que celles-ci puissent se développer de concert. Elle a aussi pour objectif de donner aux individus et aux équipes la maîtrise de leur organisation pour renforcer leur capacité opérationnelle et manœuvrière. [PEA Consulting]

Maîtrise de la configuration

Activités comprenant la maîtrise des évolutions des articles de configuration après établissement formel de leurs documents de configuration. [ISO 10007]

Documents de configuration

Documents nécessaires pour définir les exigences, la conception, la construction/production et la vérification d'un article de configuration. [ISO 10007]

Exigences système allouées au logiciel

Sous-ensemble des exigences système qui doit être pris en charge par les composants logiciels du système. Les exigences allouées constituent une des principales entrées du plan de développement logiciel. L'analyse des exigences logicielles définit et raffine les exigences allouées et donne comme résultat les exigences logicielles documentées. [SCTC]

Gestion des changements

La gestion des changements peut être décomposée en deux étapes :

- la détermination des formes ou descriptions de base ;
- la gestion des changements successifs de ces formes ou descriptions.

Les descriptions techniques sont couramment identifiées par des demandes et autorisations formelles suivant les principes d'ingénierie des systèmes et documentées dans un document de travail et selon des standards de documentation.

Les modifications de ces bases seront mises en œuvre par des demandes de changement documentées et autorisées.

La gestion du changement a pour objectif de réduire les efforts improductifs par la création d'un système de communication pendant la conception d'un système et ces composants, en permettant à l'ensemble des concepteurs d'être conscients des activités en cours pouvant affecter leurs propres travaux. [Ken Rigby 1998]

Gestion des changements du processus (logiciel)

La gestion des changements du processus (logiciel) comprend la définition des objectifs d'amélioration du processus (logiciel) et, avec le parrainage de la direction, l'identification, l'évaluation et la mise en œuvre, de façon proactive, systématique et continue, d'améliorations du processus logiciel standard de l'organisation et des processus logiciels définis des projets. [SCTC]

Gestion des changements technologiques

La gestion des changements technologiques comprend l'identification, le choix et l'évaluation de nouvelles technologies et l'intégration dans l'organisation de celles qui sont efficaces. Elle a pour objectif l'amélioration de la qualité logicielle, l'augmentation de la productivité et la diminution de la durée du cycle de développement des produits. [SCTC]

*Glossaire constitué par les membres de la commission
'Gestion de Configuration Logicielle' :*

Jean-Marc Bost

Bernard Decourbe

Michel Demonfaucon

Pierre Fishof



Square des Utilisateurs

PUMA

Proposition pour l'Unification des Méthodes Agiles

Nous avons le plaisir de vous proposer cet article aimablement transmis par Jean-Pierre Vickoff, publié dans Forum Logiciel (publication de Martinig & Associés) en avril 2002. Jean-Pierre Vickoff nous présente un panorama des nouvelles méthodes « agiles » et propose son approche pour unifier ces méthodes de développement.

Directeur de projet, architecte de SI, mais toujours concepteur et développeur (SWAT oblige !), Jean-Pierre Vickoff s'est spécialisé, en Amérique du Nord, dans la conduite de projets stratégiques sous fortes contraintes de temps.

Participant actif de l'évolution de l'état de l'art, il est l'auteur de nombreuses communications et d'ouvrages sur les méthodes. Ses travaux portent plus particulièrement sur les approches itératives incrémentales de type RAD.

Occasionnellement il réalise des conférences ou des formations pour les entreprises, les universités et les Grandes Écoles.

Un développement associant performance et qualité induit naturellement une conduite de projet itérative incrémentielle associée à un phasage simple. À l'origine de ces principes pragmatiques, la méthode RAD de James Martin¹. UML fut plus récemment le révélateur de cette nécessité méthodologique d'un couplage fort entre la forme de modélisation et le développement par prototypage. À ce jour, une dizaine de méthodes répondent à ces critères et se réclament du qualificatif d'Agile.

Les principales méthodes Agiles sont Adaptative Software Development (ASD), Feature Driven Development (FDD), Crystal Clear, Dynamic Software Development Method (**DSDM**), Rapid Application Development (**RAD**), Scrum, Xtreme Programming (**XP**) et Rational Unified Process (**RUP**). En terme de puissance commerciale les méthodes ne sont pas nées égales ; aussi, bien que RUP soit une méthode propriétaire, il est raisonnable de la citer et de prendre son influence en considération.

Heureusement compte tenu de leur nombre, ces méthodes sont globalement similaires et la plupart des valeurs et techniques qu'elles préconisent sont communes. Une étude des principes proposés révèle un tronc commun issu des racines du RAD. Seules des techniques complémentaires les unes aux autres ou mieux adaptées à des typologies et à des tailles de projets spécifiques les différencient.

Voici (en résumé) les 4 principes de base de ces méthodes Agiles :

- 1 - Les méthodes " Agiles " privilégient la **communication et l'interaction** qui en résulte à la contractualisation des spécifications.
- 2 - Les méthodes " Agiles " favorisent la **compétence et l'implication des ressources** plutôt que le respect de processus formel et d'une vision " outillée " à l'extrême des développements.
- 3 - Les méthodes " Agiles " privilégient la **livraison de fonctionnalités** réelles à la production d'une documentation pléthorique.
- 4 - Les méthodes " Agiles " favorisent l'**acceptation du changement** et la modification des **priorités** (*Time-Box, Task-Box*) plutôt que le respect d'une planification figée.

¹ Fin des années 80 pour le RAD et milieu des années 90 pour UML

Prédiction ou adaptabilité

Le paradigme des méthodes classiques est la *prédictivité*. Le paradigme des méthodes Agiles est l'*adaptabilité*. Soyons clair : aucune approche n'est réduite à une seule de ces visions. Toutes tentent de composer avec la contradiction d'une souple rigidité avec plus ou moins de finesse et avec plus ou moins de succès en fonction du contexte :

- Les méthodes prédictives tentent de réduire l'incertitude dès le début du projet par une planification très précise et très détaillée. Cette levée de risque implique que les exigences de l'application soient figées.
- Les méthodes Agiles préfèrent, partant d'une planification initiale, réévaluée régulièrement, s'adapter aux évolutions du contexte. La réévaluation servira de base à une prise de décision de type GO ou NO GO (figure 1) à chaque grand changement appliqué au projet initial.

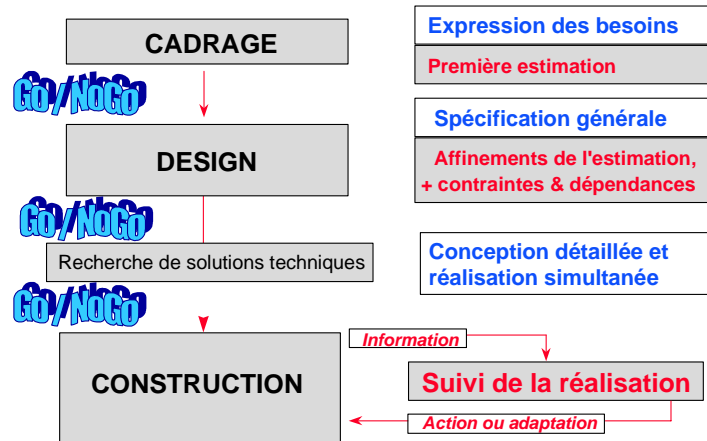


Figure 1 - Processus permanent d'évaluation / décision

Toutes les méthodes se situent concrètement à divers degrés sur une échelle les positionnant de la plus "prédictive" à la plus "adaptative". Si l'on souhaite disposer d'une vision relativement précise du champ d'application des diverses méthodes actuelles, il est nécessaire de les répertorier en fonction de ces critères (figure 2).

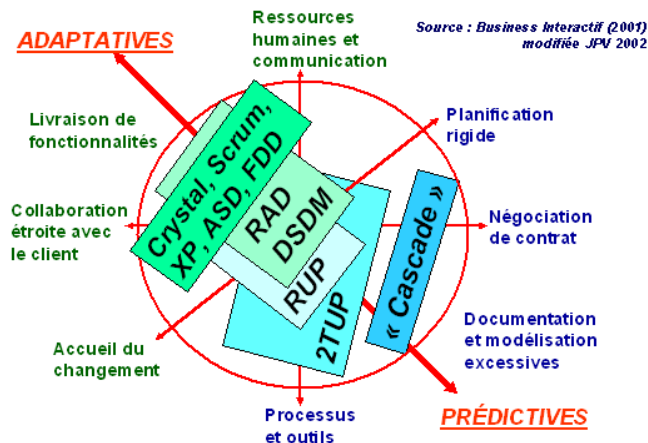


Figure 2 - Adaptativité / Prédictivité des méthodes

Les différences ainsi mises en exergue (figure3) justifient alors l'analyse fine de l'environnement du projet et du type de l'application afin de déterminer quels aspects de la méthode permettront de limiter les risques ainsi révélés et quels niveaux de service méthodologique et de qualité applicative seront mis en œuvre.

Complémentarité des méthodes

L'étude des méthodes agiles les démontrent similaires dans leurs fondements. Seules quelques techniques complémentaires entre elles ou mieux adaptées à des typologies et à des tailles de projets spécifiques les différencient. Par contre, ces méthodes offrent des couvertures plus ou moins complètes en regard des préoccupations génériques d'un chef de projet :

- 1 - **Respect de l'urbanisation** (positionnement du projet dans le système d'information).
- 2 - **Pilotage** (gestion des ressources, planning, suivi, qualité, reporting, visibilité).
- 3 - **Ingénierie de l'application** (gestion des Exigences, conception et développement, validation des livrables).
- 4 - **Conduite du changement** (impacts organisationnels et déploiement).

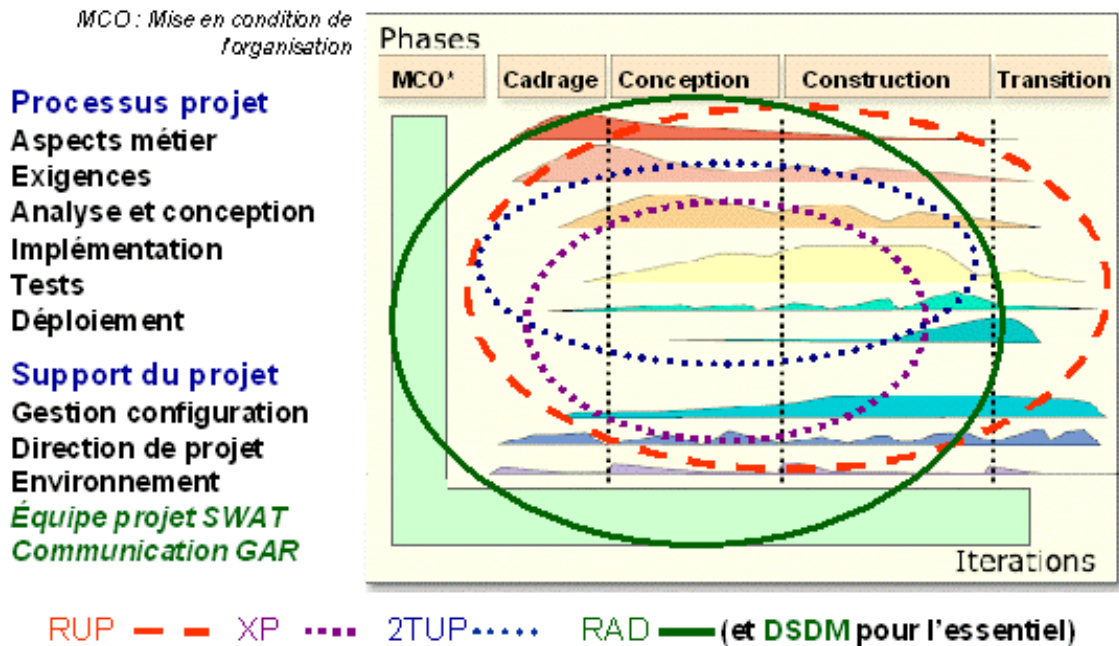


Figure 3 - Couverture de la méthode à l'ingénierie de l'application et à la conduite du projet

Tronc des pratiques communes

- Spécification et validation permanente des *Exigences*.
- Participation de l'utilisateur final aux groupes de travail.
- Interlocuteur ayant pouvoir de décision.
- Autonomie et organisation centralisée de l'équipe (motivation).
- Niveau méthodologique variable en fonction des enjeux du projet.
- Pilotage centré sur les enjeux et les risques.
- Recherche d'excellence technique de la conception.
- Vision graphique d'une modélisation nécessaire et suffisante.
- Vision d'une documentation intégrée nécessaire et suffisante.
- Planification stratégique globale basée sur des itérations rapides.
- Réalisation en jalons par prototypage actif itératif et incrémental.
- Normes et techniques raisonnables de qualité du code (métrique).
- Architecture à base de composants, gestion des changements.
- Recherche continue d'optimisation et d'amélioration des pratiques.

Les pratiques différenciatrices

Passons en revue les plus marquantes des pratiques différenciatrices.

La méthode **DSDM** se particularise par la **spécialisation des acteurs** du projet dans une notion de "rôles". Ainsi, l'on trouvera dans les réunions DSDM, des sponsors exécutifs, des ambassadeurs, des utilisateurs visionnaires, des utilisateurs conseillers, sans oublier l'animateur - facilitateur et des rapporteurs.

Le Groupe d'Animation et de Rapport représente d'ailleurs un sérieux apport du RAD à la communication de projet et à la formalisation des exigences applicatives.

La méthode **SCRUM** affirme sa différence dans des pratiques de courtes **réunions quotidiennes**. Ces temps de travail commun ont pour objectifs d'améliorer la motivation des participants, de synchroniser les tâches, de débloquer les situations difficiles et d'accroître le partage de la connaissance.

Pour **FDD**, la particularité nommée *Mission focused* réside dans une forte orientation vers un but immédiat mesurable. C'est en fait l'ambition globale d'une itération qui se trouve ainsi renforcée.

Cet aspect se retrouve aussi dans le **RAD** sous la forme des objectifs de Focus ou dans Scrum dans la notion de *Sprint*.

Comme son nom l'indique, FDD préconise le *Features Driven Development*. Cette technique se caractérise par des notions de *Feature* et de *Features set* (fonctionnalités et groupe de fonctionnalités). La priorité est donnée aux fonctionnalités porteuses de valeur.

Le **RAD** propose des techniques proches : livraison en fonctionnalité réduite ou livraison par thème.



Figure 4 - Le cycle projet de XP (les autres méthodes Agiles épousent un cycle identique)

La méthode **XP** est très axée sur la partie Construction de l'application. Une de ses originalités réside dans l'approche de planification qui se matérialise sous la forme d'un jeu intitulé *planning game* et qui implique simultanément les utilisateurs et les développeurs. On notera aussi des techniques particulières liées à la production du code comme la **programmation en binôme**, l'**appropriation collective du code**, le *refactoring* et l'**intégration continue**.

La méthode **RAD** préconise dans ce sens des revues de code personnelles, puis collectives et l'intégration avant chaque Focus. Par contre, le RAD limite la programmation en binôme aux parties les plus stratégiques ou les plus complexes de l'application.

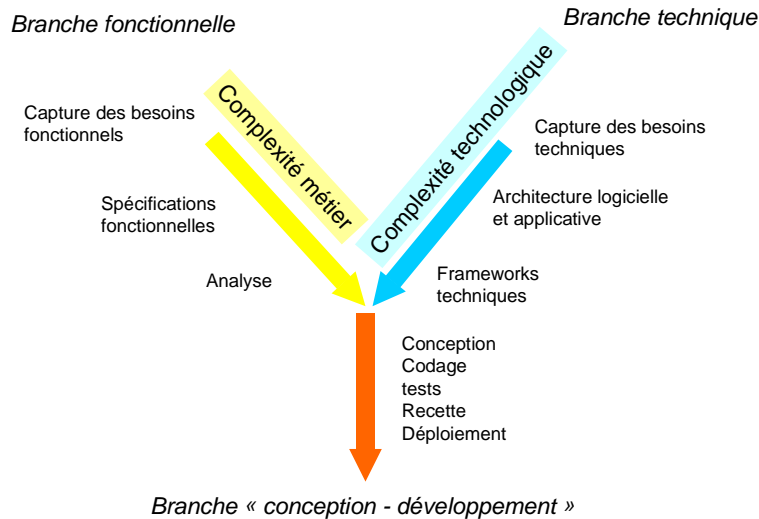


Figure 5 - Le cycle en Y de 2TUP

La méthode 2TUP² préconise un **cycle de vie en Y** qui dissocie et parallélise la résolution des questions fonctionnelles et techniques. Le cycle de vie de 2TUP s'apparente à un cycle en cascade mais introduit une forme itérative interne à certaines tâches. Il n'est pas certain que ce cycle s'apparente réellement à une approche Agile. Par contre, 2TUP préconise des formes de recherche de qualité et de performance intéressantes telles que les services réutilisables et la conception générique (*Framework* et *Design pattern*) proches des mécanismes architecturaux de RUP.

RUP³ se caractérise par une approche globale nommée "**Vue 4+1**". Les 5 composants⁴ de cette vue sont : la vue des Cas d'utilisation, la vue Logique, la vue d'Implémentation, la vue du Processus, la vue du Déploiement. RUP offre aussi, à l'identique du RAD, un **Processus guide** formel et adaptable comme guide d'activité. Dans le cas de RUP, il est malheureusement propriétaire et orienté outil.

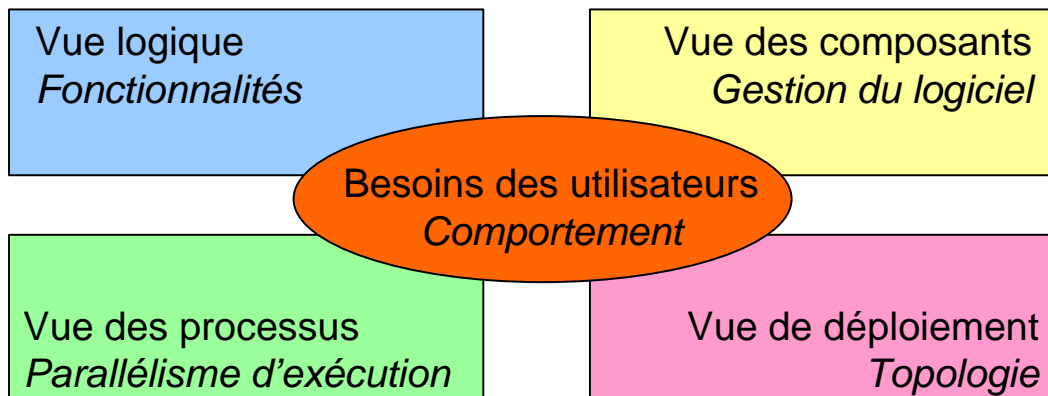


Figure 6 - La vue 4+1 de RUP

La méthode RAD (*cycle figure 7*) préconise un groupe d'animation et de rapport (**GAR**), ainsi qu'une équipe de développement particulière, le **SWAT**. Cette équipe est autonome, spécialement formée, concrètement motivée et outillée. Elle se compose essentiellement d'un profil unique de concepteurs-développeurs formés à des spécialités techniques complémentaires. Le SWAT travaille avec les utilisateurs dans une salle permanente spécialement équipée formant un plateau de communication (**salle RAD**).

² Un mélange de cycle en V et d'UDSP selon Miguel Moquillon ou un mélange de conduite de projet et de modélisation selon Jean-Pierre Vickoff.

³ Fondement du développement objet **USDP** (Unified Software Development Processus) a été développé par les auteurs d'UML pour spécifier les principes du modèle d'analyse et de conception. Ses principales implémentations sont **2TUP** de Valtech et **RUP** de Rational.

⁴ Ph. Kruchten proposa en 1995 5 différentes perspectives, indépendantes et complémentaires : la [vue \("4+1"\)](#) pour définir un modèle d'architecture qui servira de base à UML..

Le RAD recommande aussi la **variabilité de la taille et de la maturité⁵ des groupes de travail** en fonction des phases du projet afin d'optimiser l'engagement des ressources et de préserver leur intérêt par un travail adapté à leurs préoccupations. L'organisation performante des réunions est basée sur un mode opératoire des entretiens (**sessions en 3 étapes**) et sur des techniques de **validation permanente**.

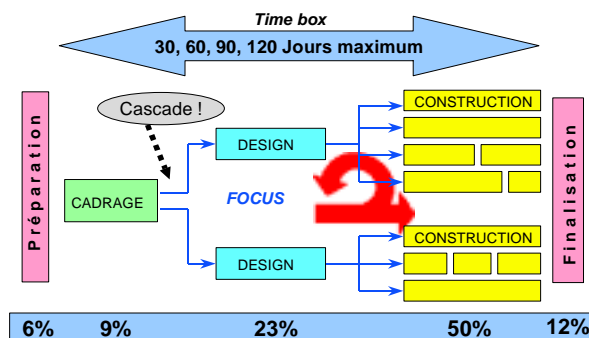


Figure 7 - Cycle RAD : parallélisation et sérialisation

Pour une agilité optimale

Si une tentative d'unification des méthodes Agiles devait se heurter à de trop grandes difficultés compte tenu de leurs origines respectives (USA), l'émergence d'une méthode unificatrice de l'ensemble des bonnes pratiques qui les caractérisent devrait cependant s'avérer possible.

Une fois les pratiques communes et les pratiques différenciatrices isolées, il est aisé d'imaginer ce que devrait être la méthode optimale en fonctions d'un type particulier de projet (*figure 8*).

La méthode Agile unifiée se composerait de l'ensemble ou d'une sélection des pratiques communes auxquelles il conviendrait d'ajouter la ou les pratiques spécifiques judicieuses en fonction du contexte. L'ensemble de ces aspects s'inscrivant obligatoirement dans un niveau variable de service méthodologique.

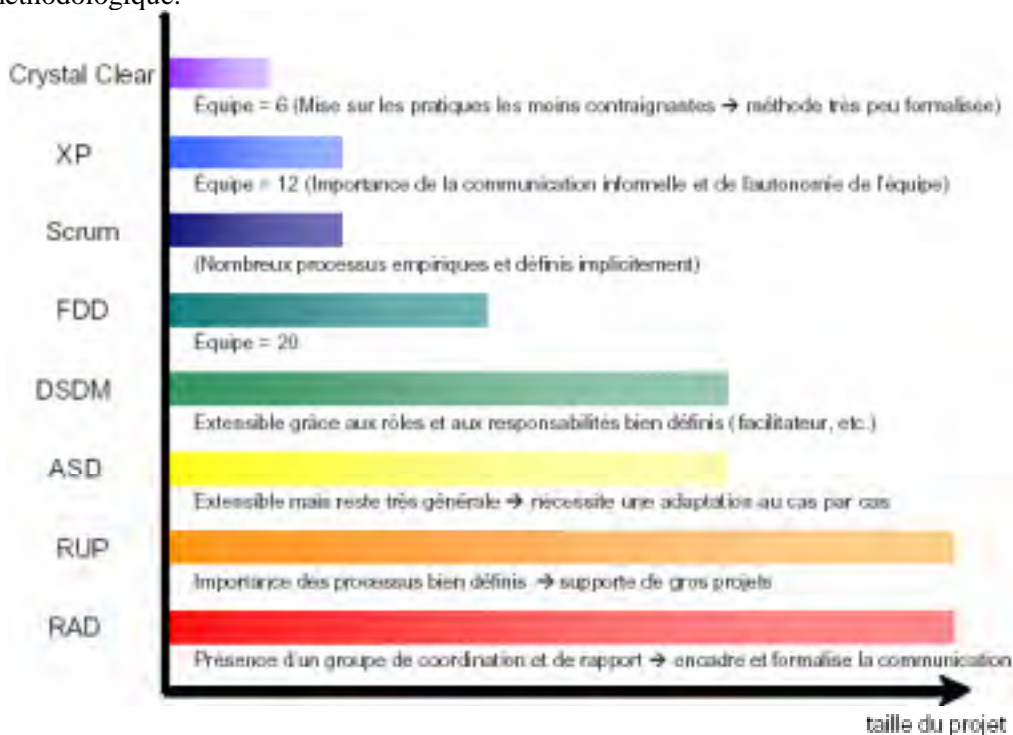


Figure 8 - Taille du projet / adaptation de la méthode

⁵ La mise en œuvre de ces distinctions est de la responsabilité de l'animateur - facilitateur.

Piloter des enjeux et des risques

Le pilote d'un projet de développement est confronté en général à la maîtrise de 3 processus distincts et concourants :

- Piloter un projet *rentable*.
- Construire une application *adaptée*.
- Organiser un changement *attendu*.

Chacun de ces aspects s'inscrit dans une problématique d'enjeux et de risques. Un processus sécurisé de conduite de projet couvrira donc l'ensemble de ces aspects (*figure 9*). La pathologie du projet en difficulté met globalement en évidence l'imbrication de multiples causes de défaillance. En revanche, les effets se limitent généralement à quatre types d'insuffisances :

- l'adéquation fonctionnelle,
- la qualité technique,
- les conditions économiques de réalisation,
- la préparation au changement.

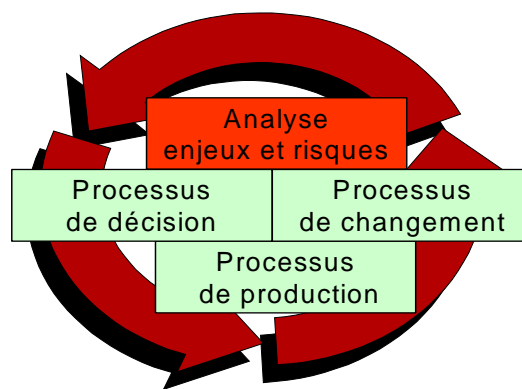


Figure 9 - Pilotage des processus par les risques

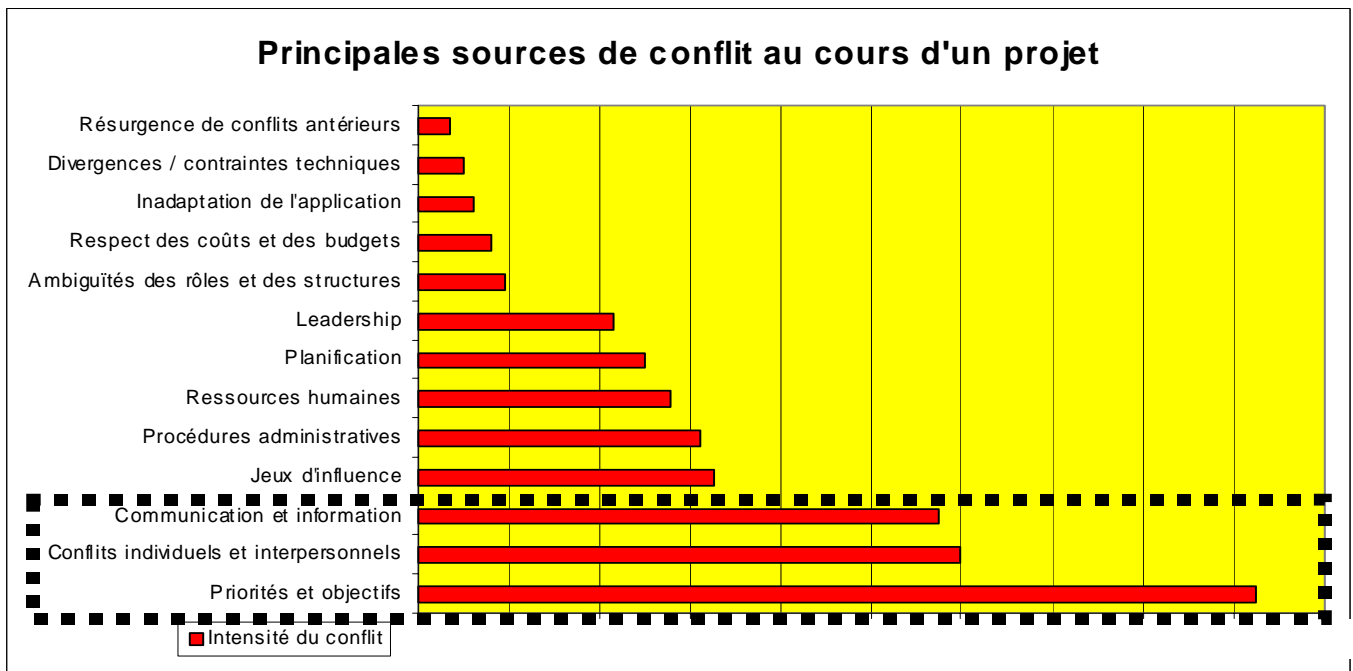


Figure 10 - Risques dans les projets (source Hervé Courtot)

Lorsqu'une, ou plusieurs de ces insuffisances se manifestent, le projet est en difficulté. Pour intervenir efficacement et recentrer un projet déviant, il faut nécessairement disposer d'une conduite de projet

efficace en regard des risques qui se sont matérialisés. Hervé Courtot, expert en ce domaine, détermine les 3 risques actuels les plus importants comme étant liés à la communication, aux rapports humains et à l'arbitrage des priorités (figure 10).

Une méthode agile unifiée devra donc s'attacher en priorité à réduire ces risques chroniques. L'apport du RAD dans son principe **animation facilitation neutre** est prépondérant pour traiter préventivement ces trois cas.

Relation MOA / MOE

Dans les applications actuelles le **métier** s'adapte à la compression du temps. Les solutions sont liées aux **technologies** et évoluent en cours de projet. La complexité des solutions actuelles impose alors une MOA sachant spécifier ses besoins actuels comme futurs et proposer une dynamique applicative. L'engagement planifié et si possible permanent des utilisateurs devient indispensable.

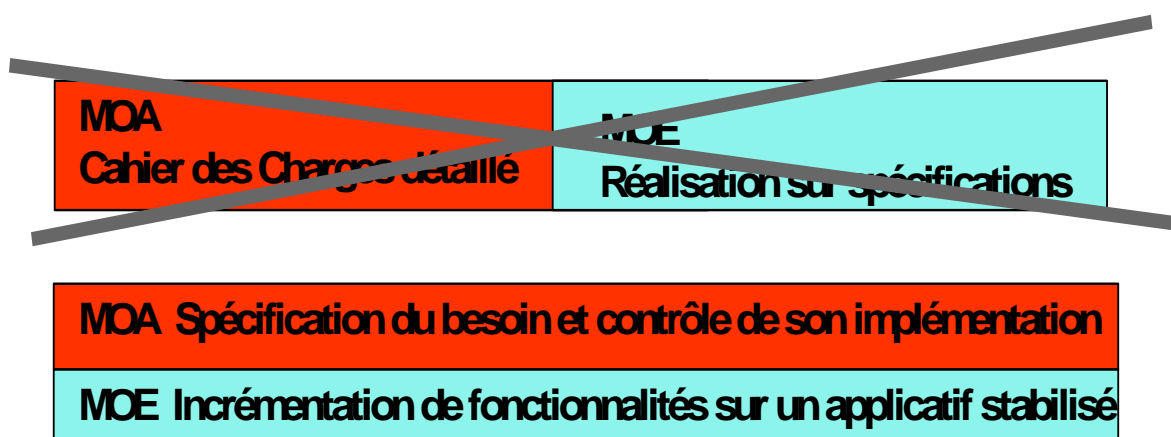


Figure 11 - La relation MOA / MOE actualisée

Pour sa part, la MOE doit se considérer comme une force de services de réalisation et de proposition technologique. De ce constat il découle que contrairement à certains discours plus que douteux, la relation maîtrise d'œuvre / maîtrise d'ouvrage n'est pas caduque ! Ce qui n'est plus adapté, c'est l'engagement séquentiel de ces deux parties (*figure 11*). Il est même possible d'affirmer que cette relation n'a même jamais été aussi indispensable. Surtout si l'on prend en compte la nécessité d'externaliser l'obtention de parties de qu'il faudra contractualiser. Ce modèle répond à une logique universelle⁶ et est d'ailleurs recommandé par le CIGREF. Méthodes Agiles ou non, la définition des responsabilités doit être formelle.

Pour mieux comprendre les distinguos introduits par ces concepts, en voici une concrétisation :

La Maîtrise d'Ouvrage a pour champ de responsabilité l'évolution des métiers de l'organisation. La MOA raisonne en termes de systèmes d'information, de processus et d'applications. La MOA dirige les projets qu'elle commande.

La Maîtrise d'Œuvre a pour champ de responsabilité la fourniture de la solution. La MOE raisonne en termes de systèmes informatiques, de conception et de réalisation. La MOE assure le pilotage et d'ingénierie de la solution.

Les choix « d'externalisation »

Compte tenu de la complexité croissante des applications, les possibilités et nécessités d'externalisation de la solution deviennent stratégiquement indispensables. Le modèle que le RAD propose au chef de projet se présente en 7 couches externalisables en fonction du projet :

⁶ Exemple aux USA : Owner / Chef Information Officer.

- 1 - développement de la partie **générique** ;
- 2 - développement de la partie **spécifique** ;
- 3 - stratégie de **maintenance** ;
- 4 - solution de **déploiement** ;
- 5 - stratégie d'**assistance** aux utilisateurs ;
- 6 - solution d'**exploitation** ;
- 7 - solution de **logistique** ;

Stratégie de planification Agile

Il existe globalement 4 types de contraintes pour un projet. Le RAD propose des techniques de conduite de projet (*figure 12 colonne de droite*) adaptées à chacune de ces contraintes. Malheureusement, ces 4 contraintes sont contradictoires et vouloir les gérer à l'optimum pose un sérieux problème de combinatoire. L'outil Évaluateur, gratuit sur le site www.RAD.fr, se propose de vous aider à maîtriser cette complexité grâce à un assistant de planification stratégique de projet.



Figure 12 - Les 4 formes de contraintes du projet RAD

Le PAQ « Agile »

Pour le RAD, l'Agilité n'est pas contradictoire avec le principe de l'assurance qualité. Par contre, la qualité doit être rentable, aussi le PAQ sera par principe « minimaliste » (mais normalisé CMM).

Le principe sous-jacent à sa constitution doit prendre en compte la réflexion suivante : « tout ce qui n'est pas directement la production d'une fonctionnalité est une action parasite aussi utile qu'elle puisse être ».

Voici les points d'un PAQ générique pouvant être formalisés dans le PAQ spécifique du projet :

- assurance qualité et niveau de service ;
- organisation, communication, pilotage ;
- gestion de la qualité du logiciel ;
- méthode de développement logiciel ;
- gestion des exigences ;
- gestion du plan de tests ;
- gestion des divergences ;
- gestion quantitative des charges ;
- gestion de la planification ;
- gestion de la documentation ;
- gestion " pragmatique " des risques ;

- gestion de la configuration et des versions.

À partir de cette liste, le chef de projet choisira les pratiques qui lui conviennent en terme de levé des risques pressentis. Pour résumer, ce PAQ de base contient le minimum requis pour un pilotage et un développement sécurisé.

À l'extrême de la simplification, l'assurance qualité peut se limiter à ces aspects :

- 1 - Plan de communication et d'engagement ;
- 2 - Gestion des exigences, tests et divergences ;
- 3 - Référentiel d'application et mémoire du projet ;
- 4 - Planification, évaluation, suivi du processus.

Cette simplicité en 4 points peut se traduire textuellement ainsi : (1) Les bons interlocuteurs (maîtrise d'ouvrage et utilisateurs) se donnent les moyens (2) d'exprimer les besoins et de valider la réalisation sous une forme conservée et réutilisable (3). Le pilote du projet et les techniciens de la maîtrise d'œuvre garantissent alors les conditions de fabrication (4).

Arrêt d'urgence, mode opératoire

Toute méthode de conduite de projet devrait inclure un mode opératoire pour les arrêts d'urgence. Pourtant, l'étude des plus anciennes méthodes prédictives, semble mener à un constat d'optimisme : les auteurs n'ont jamais connu l'échec et leurs disciples doivent suivre cette voie.

Sur ce point, la force du RAD se situe dans la présence d'un animateur-facilitateur. Cette ressource, de préférence externe, doit être neutre en regard des autres intervenants. Elle répond à une autorité supérieure à tous les participants du projet. Ainsi, lorsque le contexte stratégique, économique ou technique d'un projet évolue, ou si les conditions de réalisation se dégradent, l'animateur reporte le problème au dirigeant qui l'a mandaté. Ce dernier peut alors prendre dans les meilleurs délais et avec des informations objectives les décisions qui s'imposent.

Dans un environnement classique de projet, les acteurs impliqués sont parties prenantes de la problématique et la dissimule dans l'ambition souvent louable d'y porter remède. Le retour sur investissement, les plannings, les budgets sont donc « améliorés » afin de lisser la problématique. Cette situation conduit à des blocages dans la remontée d'informations objectives au-delà d'un premier niveau hiérarchique et aboutit finalement à une rupture dans la chaîne de décision et de commandement. À ce niveau d'altération de la réalité, toute tentative de recentrage met généralement en évidence des dysfonctionnements dans l'organisation. La notion de *Courage*, une des quatre valeurs fondamentales de *l'eXtreme Programming*, prend alors toute sa signification et certains commencent à regretter qu'une autre de ces valeurs, le *FeedBack*, n'ait pas été prise en considération plus tôt.

Variabilité du niveau de méthode

Une solution méthodologique efficace dans un environnement donné peut ne pas être adaptée à un environnement de taille, de complexité ou de nature différente. Adapter méthode et qualité de la production à la dimension économique et au cycle de vie des applications est devenu un mal nécessaire. Voici, dans l'esprit du RAD et de *l'Agilité*, les éléments constitutifs du niveau de service méthodologique d'un projet avec lesquels il faut composer :

Niveau de service « pilotage »

- Planification et suivi de l'avancement.
- Planification et suivi des risques.
- Planification et suivi de la sécurité.
- Formalisation des activités de pilotage.
- Formalisation Exigences / Divergences
- Formalisation de la gestion de configuration.
- Formalisation de la gestion des tests.

Niveau de service « application »

- Qualité technique de l'application.
- Qualité fonctionnelle de l'application.
- Qualité de la documentation technique.
- Qualité de la documentation d'utilisation.

On notera la séparation faite entre le pilotage du projet et l'ingénierie de l'application.

Pour exemple : on ne documente pas à l'identique une application dédiée à quelques cadres permanents et une application prévue pour des centaines d'utilisateurs en fort turn-over.

Forme de modélisation

En fonction du contexte, la forme⁷ et le niveau de modélisation devraient pouvoir être choisis. À l'évidence, la plupart des méthodes récentes sont fortement influencées par l'Objet et plus particulièrement par UML. Par contre, la nécessité d'imposer sans discernement une modélisation UML complète⁸ est de plus en plus controversée.

Un courant de pensées simplificateur, l'*Agile Modeling*, considère cette théorisation à outrance comme inadaptée aux projets sous fortes contraintes de temps ou d'argent.

Le pragmatisme impose au pilote de projet de conserver à l'esprit que **l'objectif est une application** et non un modèle. De par mon expérience, je vois se profiler avec UML des excès identiques à ceux auxquels avait abouti Merise voici quelques années. Le plus inquiétant est le fanatisme des tenants de l'objet. Aucune critique des pratiques de leur secte n'est entendue.

L'Objet propose depuis de nombreuses années des techniques de conception qui ne sont pas liées à une forme de modélisation particulière. Citons parmi les plus connues ou utiles la dissimulation, la modularité, l'abstraction, l'encapsulation, la cohésion, le couplage, la hiérarchisation, l'héritage, le polymorphisme, l'algorithmique de base, la structuration (des données et des traitements).

Le but ultime d'une conception efficace est de faciliter le développement et la maintenance ultérieure mais aussi la conduite de projet dans sa rencontre avec le nerf de la guerre : le retour sur investissement. Parmi d'autres, le principe de « conception en vue de modifications » a été imaginé pour ces raisons. Il se base essentiellement sur la technique de « dissimulation d'information ». Pour la mettre en œuvre il est nécessaire de respecter les étapes suivantes :

- Définition exhaustive des fonctionnalités.
- Identification des règles de gestion instables.
- Encapsulation des parties modifiables.
- Définition formelle des interfaces d'appel.
- Programmation modulaire.

⁷ Merise avec ses niveaux d'abstraction et de préoccupation, les Flux associés à l'entité-relation, l'Objet, ...

⁸ Pas moins de 9 types de diagrammes

Les gourous du développement qui ne comprirent pas tout de suite l'intérêt de cette pratique furent contraints rapidement de faire amende honorable. Fred Brooks « La critique de cette technique est une des rares erreurs de la première édition de mon livre, *The Mythical Man-Month* ». Boehm « C'est une des rares techniques théoriques qui ait fait la preuve de son utilité pratique ». Quelle que soit l'expression de modélisation choisie, mettre en œuvre systématiquement de bonnes pratiques de conception serait déjà un grand pas. Selon des statistiques de Microsoft (*54 Règles pour un grand logiciel*) 97.5 % des informaticiens ne connaîtraient pas ou ne pratiqueraient pas ces techniques basiques de conception.

Engagement des ressources

La nouvelle nature des applications requiert l'expression d'un besoin dont la complexité s'accroît régulièrement. Dans le même temps, les technologies chargées d'apporter des solutions subissent, elles aussi, des évolutions majeures. De plus, la contrainte principale est de plus en plus souvent liée à la compression du temps. **De ces pressions découle le Mode Projet.**

Cette organisation concentre dans un même espace géographique et temporel l'ensemble des intervenants fonctionnels et techniques engagés dans une unique mission. Les projets menés suivant cette approche sont des succès. Au regard de ce constat, on est en droit de s'étonner de la continuité, voire du développement d'une gestion matricielle des ressources. Les fervents de cette approche, en général de purs gestionnaires, espèrent pouvoir optimiser l'engagement de leurs ressources sur de multiples projets simultanés. Théorie parfaite en ce qui concerne des petits travaux de maintenance, mais battue en brèche par une réalité différente dans les projets de développement : **le crime de dispersion ne paie pas !**

Les causes sont profondément humaines : perte de reconnaissance et de responsabilité dans l'aboutissement du projet. En pratique, il m'a été donné d'observer la totale absence de participation d'individus alors qu'ils étaient, théoriquement, affectés entre 5 et 15% de leur disponibilité ! D'autre fois la pléthore de ressources engagées aboutissait à des résultats semblables : « Lorsque tout le monde semble s'occuper de tout, en général personne ne s'occupe de rien ».

Une équipe RAD reste de dimension raisonnable (4 personnes à l'idéal, 8 au maximum), mais l'ensemble de l'équipe est engagée en bloc au début du projet (Cadrage).

Motivation de l'équipe projet

Les méthodes Agiles s'efforcent de travailler avec les spécificités des ressources humaines plutôt que contre leur nature. La meilleure garantie de succès d'un projet reste l'engagement d'une équipe motivée. La capacité de travailler en groupe et de communiquer avec l'utilisateur est primordiale.

Le RAD recommande un forme d'équipe particulière le SWAT (*Special Weapons and Tactics*) qui devient dans l'engagement d'un projet *Skill With Advanced Tools*.

Les membres de l'équipe ont un profil unique⁹ de concepteur-développeur mais sont surentraînés dans une spécialité et formés aux techniques d'entretiens RAD. Basant leur mission sur la communication avec l'utilisateur, ils se complètent et s'appuient.

Afin de motiver exceptionnellement une équipe, la méthode RAD préconise de :

- Définir précisément le défi à relever.
- Formaliser la récompense (en fonction de l'atteinte des jalons définis).
- Enrôler l'équipe en donnant le choix de l'engagement à chacun individuellement.
- Organiser l'autonomie de l'équipe et termes de moyens et d'organisation du travail.
- Comptabiliser (simplement) les actions d'assistance entre les membres du SWAT.
- Personnaliser les travaux en fonction des désirs et des possibilités des ressources.

⁹ L'introduction d'un élément fonctionnel dans un SWAT est l'idéal dans les développements modernes exigeants (High-Tech, High-Touch).

- Publier régulièrement l'avancement des résultats (sans langue de bois).
- Surveiller le tarissement de l'enthousiasme et récompenser les efforts par divers avantages.
- Fêter l'accomplissement des jalons significatifs à partir d'un budget dédié (pots, restaurants, sorties).
- Favoriser l'émergence de signes d'identité de l'équipe (pas très bien compris en France).

Le retour sur investissement de ces pratiques "à l'américaine" est immédiatement évident.

Sur le plan technique, comme le recommande la structure d'un SWAT, favorisez la formation de généralistes-experts¹⁰. Répertoriez ces expertises puis affichez un tableau des ressources à consulter en cas de questions pointues. Cela permettra de développer une culture d'entre aide entre les membres de l'équipe.

Afin que les plus compétents ne soient pas pénalisés par le nombre des interventions de soutien, il faut mesurer l'échange de compétences et de services. Ceci permet de gagner un temps précieux en recherche. Ceci permettra aussi d'éliminer les craintes du « demandeur » qui pourraient découler de réactions négatives de l'expert si ce dernier n'était pas dédommagé de son temps investi dans le transfert de compétences.

Ce dernier point acquis, **il faut focaliser la plus grosse partie des récompenses sur les résultats d'équipe et non individuels**, car le SWAT doit rester une synergie et une complémentarité de spécialistes engagés dans un succès d'équipe.

Les 10 commandements du RAD

Bien que la chance ait sa part dans un projet, comme dans la plupart des activités humaines, respecter des règles de bonne conduite (de projet) permet de s'assurer qu'elle ne sera pas gaspillée par négligence.

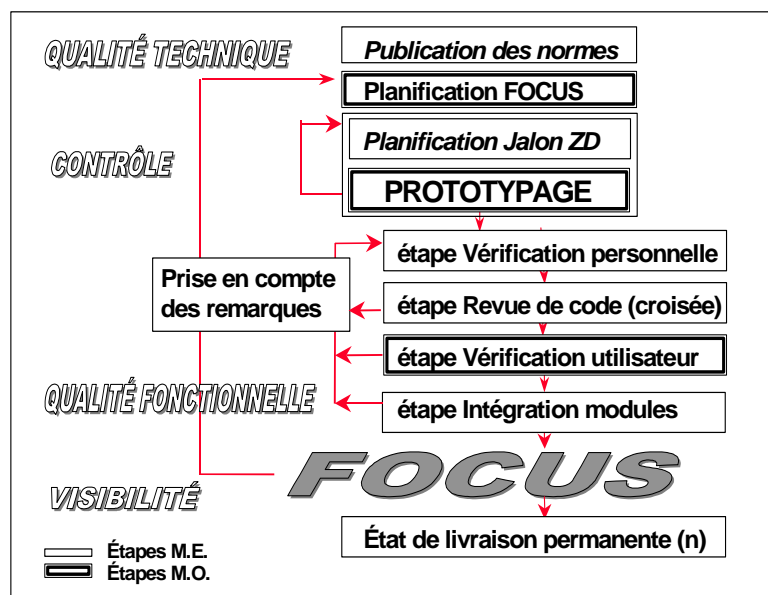


Figure 13 - Techniques de maîtrise de la réalisation

Voici les 10 commandements du RAD :

- 1 - Exigez un environnement organisationnel et technologique efficient.
- 2 - Pratiquez la validation permanente avec l'utilisateur réel.
- 3 - Planifiez de manière réaliste au début puis réévaluez régulièrement (GO/NOGO).
- 4 - Respectez le phasage et les pré-requis de poursuite.

¹⁰ généraliste de la conception et du développement mais expert dans un domaine complémentaire

- 5 - Respectez la dimension temporelle (optimum 90 Jours).
- 6 - Validez immédiatement la faisabilité opérationnelle (cycle en Y de 2TUP).
- 7 - Composez une équipe techniquement et fonctionnellement adaptée au défi.
- 8 - Appliquez les concepts fondamentaux de la conception Objet.
- 9 - Appliquez les techniques de maîtrise de la réalisation : Jalons ZD, revue de code (*figure 13*).
- 10 - Assurez le respect des livrables et des Focus de visibilité.

Rappels et conclusions

Comme le met en évidence le tableau « facteurs de succès et causes d'échecs » (*annexe 2*), dans la plupart des projets les échecs relèvent rarement de la technique à elle seule. De dangereux écueils apparaissent dilués dans les méandres des relations humaines et des impératifs économiques.

Cette constatation est particulièrement évidente lorsque le projet se situe sur le terrain de la nouvelle économie, de l'innovation, du trans-domaines et parfois même du trans-organisations. La réussite du projet nécessite alors d'associer simultanément deux aspects généralement contradictoires : la **créativité** indispensable aux ambitions d'innovation différenciatrice et la **rigueur** de contraintes économiques ou stratégiques. Le pilotage par **les enjeux, les risques et les contraintes** s'affirme alors comme une nécessité vitale.

De ce constat émerge une certitude : les nouveaux projets font évoluer en profondeur le cadre de travail des DSI, des chefs de projet et des maîtres d'ouvrage. Il leur faut se doter d'une structure de développement formelle en terme d'actions et de responsabilités.

Apparaît ensuite la nécessité d'une **méthode de développement semi-itérative** sécurisée par un processus fiable et adaptable.

Pour être économiquement viables, l'ensemble de ces aspects s'inscrit obligatoirement dans un **niveau variable de service méthodologique et de qualité requise**.

De plus, l'innovation, lorsqu'elle atteint un certain degré, affecte fondamentalement le cœur du métier. Le projet fait alors office de révélateur quant à la nécessité d'une **remise à plat des processus** que d'ailleurs l'Objet impose naturellement. À partir de ce constat, c'est souvent l'organisation complète qui se doit d'évoluer. Une telle amplitude de changement de mode de travail, lorsque ce n'est pas directement de culture, est encore exceptionnelle et représente pour l'encadrement « le temps de tous les dangers ». Les freins à l'évolution sont donc puissants.

La mise en œuvre de la méthode, dans ses aspects de communication élargie, doit en tenir compte.

Au delà de la maîtrise des projets, l'essentiel dans l'usage d'une méthode itérative et incrémentielle s'appuyant sur l'utilisateur, c'est le **dynamisme** qu'elle induit dans l'organisation parallèlement à l'acceptation du changement qu'elle favorise.

Jean-Pierre Vickoff

Bibliographie

Livre blanc : Méthode Agiles état des lieux, www.BusinessInterActif.fr

[Paulk], The Capability Model : Guidelines, SEI, 1995

[P. Roques et F. Vallée], UML en action, Édition Eyrolles

[Philippe Kruchten], Rational Unified Process, Édition Eyrolles

[Vickoff J-P], Piloter les projets informatiques de la nouvelle économie, Éditions d'Organisation, 2000

ANNEXE 1

Principaux facteurs de succès et causes d'échecs des projets actuels

Aspects	Facteurs de succès	Causes d'échecs
Politique	Le directeur du projet et/ou le responsable du groupe d'animation et de rapport répond à une autorité supérieure (à la direction générale dans un projet stratégique)	Le directeur de projet ne dispose pas d'un mandat suffisant pour arbitrer les divers intervenants, limiter leurs luttes d'influence, leurs ambitions ou visions partisans.
Pilotage	Piloter le projet <i>dynamiquement</i> essentiellement par les enjeux et les risques .	Piloter le projet <i>administrativement</i> essentiellement par les budgets et les ressources .
Organisation	Engager un mode projet centralisant sur un plateau unique tous les intervenants aussi bien sur le plan organisationnel que géographique. Formaliser dans le cadre de courtes missions de réelles délégations de responsabilité verticales et latérales.	Utiliser des ressources à temps partiel et/ou non dépendantes directement de la direction de projet. Ne pas disposer de l'agenda électronique de l'ensemble des intervenants. Négliger la couverture des participants en terme de responsabilité hiérarchique.
Financier	Justifier le projet par un plan d'investissement global mais simple , dont chaque élément est une base réaliste et acceptée.	Créer et maintenir un modèle financier dont le niveau de détail interdit ensuite sa remise en question et dont la complexité fera ensuite douter de sa pertinence .
Planification	Intégrer dès le début du projet l'ensemble des sous-projets et des contraintes dans un planning réaliste qui sera ensuite suivi jalon par jalon avec un outil professionnel léger.	Négliger un contrôle approfondi sur l'avancement des diverses parties sous traitées ou réalisées dans des sites éloignés de la direction du projet
Innovation	Viser l'utilisation de <i>technologies émergentes</i> dont la stabilisation est prévue pour la date du déploiement afin d'obtenir l'optimum d'efficacité stratégique	Se limiter aux technologies ou puissance de machine <i>disponible à la date du cahier des charges</i> ou de la réalisation pour des raisons administratives ou contractuelles
Communication	Employer un groupe d'animation et de rapport disposant de moyens modernes pour dynamiser la communication ainsi que formaliser et centraliser l'information.	Ne pas distinguer les groupes de travail de ceux de validation . Laisser les intervenants organiser en permanence des réunions <i>brainstorming</i> , non préparées ou non structurées et à participation variable.
Méthode	Cycle de vie itératif incrémentiel : réaliser des livraisons de résultats partiels sous la forme de plusieurs Focus de validation, suivis d'un prototype final, puis d'un site pilote.	Cycle de vie cascade classique : avant même de lever les risques organisationnels ou techniques, viser directement un système totalement finalisé dans la vision d'un déploiement total et parfait.

ANNEXE 2

Analytique cartésien (prédicatif) / Pragmatique empiriste (adaptatif)

Comme le met en exergue le tableau suivant, les approches prédictives et adaptatives se réclament de postulats épistémologiques fondamentalement différents.

Méthodes	Dites classiques ou « <i>Lourdes</i> »	Dites nouvelles ou « <i>Agiles</i> »
Paradigme fondateur	Prédicativité	Adaptabilité
Fondement	Analytique cartésien	Pragmatique empiriste
Cycle projet	En cascade (sans rétroaction)	Incrémentiel et Itératif (adaptatif)
Forme de levée du risque	Descriptive et documentaire	Recherche - action - expérimentation
Raisonnement	Discursif (prémisses conclusions)	Systémique et heuristique
Vision sous-jacente	Isoler pour structurer une partie d'univers figé	Exécuter pour comprendre la dynamique des interactions
Pensée	Réductionnisme et hypothèses mécanistes	Vision holistique des phénomènes (ressources humaines, communication, environnement etc.)
Philosophie d'analyse	Considère la nature des interactions	Considère les effets des interactions
Structuration méthode	Sur la base de niveaux isolants d'abstractions et de préoccupations figés	Sur la base d'un phasage simple et souple prenant en compte les contraintes du projet
Axe de recherche	L' analyse de la structure	L' aboutissement des actions
Limites et possibilités	Réduction de systèmes simples par l'analyse	Appréhension de systèmes complexes par leurs finalités
Conduit à des systèmes	À forte entropie	À forte rétroactivité , « cybernétique »
Aboutissement	Recherche l' exhaustivité de la solution	Accepte un « rendement satisfaisant »
Philosophie d'action	Conduit à une action totalement détaillée et programmée	Conduit à une action par objectifs et flexible
Aboutit concrètement	À la reconduction de structure existante	À l' amélioration des éléments de performance
Validation	Comparaison théorique à base de jeux d'essais en fin de parcours	Confrontation permanente du modèle avec la réalité (prototype)



Square des Utilisateurs

Structures et processus

La flexibilité et la réutilisation dans les systèmes

*Après la présentation faite en Assemblée Générale sur le thème **Processus d'affaire et maîtrise du Système d'Information** et avant les Assises prévues en fin d'année, nous commençons ce qui sera, nous l'espérons, une série d'articles sur les processus. On pourra plus tard évoquer les processus de développement et de configuration, les facteurs de *Qualité* ou la modélisation. Pour cet article introductif, nous avons choisi de nous concentrer sur la flexibilité et la réutilisation, thèmes très longtemps délaissés avant de devenir une antienne dans les méthodes Orientées Objet, demain peut être dans les services Web.*

Si l'on considère le concept système comme un outil de modélisation, c'est à dire de représentation de la réalité perçue avec un point de vue déterminé, l'entreprise peut être représentée comme un ensemble d'éléments constitutifs en interaction. Une entreprise est donc un système qui comporte des éléments matériels, économiques et humains en rapport les uns avec les autres. Une fois les relations d'ordre logique et chronologique prises en compte, on peut le qualifier de système organisationnel ou SO. Il est totalement ou partiellement décrit par un ensemble de modèles comme l'organigramme des responsabilités, les nomenclatures de produit, les budgets par centre de coûts ou de profits, le plan comptable, les règles de gestion. Une application ou un agrégat de programmes informatiques est également un système au sein duquel les relations sont le plus souvent exprimées en recourant aux structures de bases de données ou de classes d'objets métiers, d'une part, et de traitement, d'autre part. Soulignons que le notion de Système d'Information est volontairement délaissée dans ce qui suit ; SI doit bien être entendu comme Système(s) Informatique(s).

Les structures du type des associations « client – commande – produit » sont stables dans le temps, voire statiques¹. La cartographie des applications l'est tout autant puisque basée sur ou inspirée par une décomposition selon les grandes fonctions de l'organisation (voir figure 1), elles-mêmes définies par les métiers des personnels (vendeurs, acheteurs, administratifs, financiers, productifs, etc.) depuis le début du XX^{ème} siècle avec Fayol.

L'effort plus récent d'urbanisation des systèmes respecte toujours cette orientation métier – tout en attribuant à ce terme une acception partiellement nouvelle qui est celle du secteur économique au sein duquel l'entreprise développe l'essentiel de ses activités² –. Il essaye de cantonner les fonctions métier aux invariants supposés tandis que des fonctions de service recomposables sont identifiées, toutes échangeant des messages sur des artères de communication de différents gabarits. L'échange complète donc la notion initiale de traitement.

De nouveaux invariants structurels pour traiter des associations variables

Pourtant les systèmes évoluent.

Les bases techniques des SI, leur architecture, leur localisation, leur sécurité, leur administration, évoluent ; leur déploiement devient adaptatif et se configure. Les SO incluent les missions globales ou particulières, la définition des produits propres ou distribués, l'appareil productif et logistique, les canaux de distribution et leurs relations contractuelles sous-jacentes, les formes et l'extension des

¹ Au gré des événements, des occurrences des entités type ou des associations type sont bien créées mais la structure du modèle de données ou de traitement n'est pas pour autant modifiée.

² On peut remarquer que l'idée d'urbanisme est née à la même époque que les progiciels se sont répandus, dans le domaine de la paye ou de la gestion de fabrication par exemple. Il devenait moins nécessaire de structurer le SI de l'entreprise exclusivement selon les fonctions générales. On s'est donc focalisé sur les fonctions spécifiques aux métiers de l'entreprise. L'abus du terme métier ne laisse pas d'inquiéter ici, ne serait-ce que parce qu'il suppose une stratégie figée en la matière.

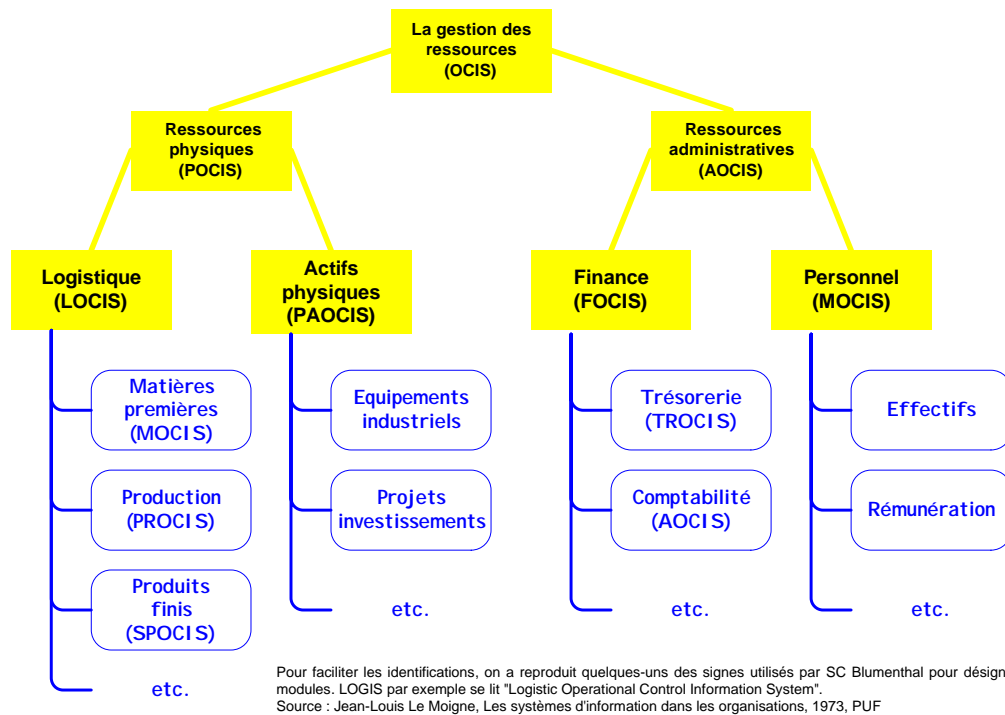


Figure 14 – L'alignement historique des applications sur les fonctions de l'entreprise

partenariats et externalisations. Mais en toute hypothèse, il existe des missions, des produits, des opérations, des canaux, des partenariats, des plates-formes. *Les SO évoluent pourtant parfois plus vite que les SI³.*

L'important à ce sujet est bien d'identifier les *structures type* sous-tendant les caractères évolutifs des objets de gestion et pas seulement les instances. Il s'agit de l'expression d'associations ou règles de composition comme la hiérarchie des unités organisationnelles, la nomenclature des sous-ensembles et des composants, la gamme ou procédé opératoire, la commande ou contrat, les restitutions entre partenaires. Ce qui a été fait avec les bases de données ou les objets métiers spécifiques à l'entreprise ou aux ERP. Un système de gestion de bases de données mutualise en effet la gestion des structures de fichiers auparavant incluse dans chaque programme d'application. C'est maintenant le cas des interpréteurs (« parsers ») de structures documentaires XML ; ils traitent aussi bien les documents que les schémas génériques de ces documents. *Les programmes génériques sont ceux qui travaillent sur des structures ; la variabilité de celles-ci permet la flexibilité des SI utilisant les premiers.*

Pour s'intéresser à la flexibilité des systèmes SO et SI, il est donc de bonne méthode de considérer leurs caractéristiques structurelles plutôt que fonctionnelles. Parmi ces caractéristiques, tout en assumant une grande diversité dans la situation des entreprises, on peut sans doute citer :

1. Le fondement *transactionnel* de la gestion, c'est-à-dire que les événements extérieurs au SO ou au SI sont générateurs d'informations qu'il convient d'enregistrer dans les bases de données et déclenchent des opérations humaines ou automatiques. La gestion n'a qu'un rapport lointain avec un corpus de règles déterministes et programmables même si certaines opérations sont, pas commodité, regroupées en lots et traitées par des automates informatiques. Le caractère événementiel du SO est patent ; le SI doit le plus possible conserver ce caractère si on veut l'aligner en permanence sur le SO ;
2. L'implication *humaine* qui est nécessaire dans la quasi-totalité des situations pour négocier, choisir, approuver, commenter ou s'informer, en particulier avec des règles floues ou sans règles⁴. Les toutes dernières annonces de plates-formes supportant les solutions à base de services Web

³ Il est piquant de considérer que les formats de fichiers et les logiciels eux-mêmes sont maintenant souvent plus durables que les matériels informatiques qui sont capables d'en faire usage.

⁴ On peut évoquer le « structuralisme » de PIAGET pour lequel la structure de tout modèle dépend de la réalité observée mais aussi de l'observateur et de leur interaction.

insistent d'ailleurs sur cette nécessité comme celles issues de l'intégration d'applications d'entreprise (EAI)⁵. Un acteur humain n'a usuellement pas à être défini comme « servant » une application ; il est souvent l'auteur des actes à valeur ajoutée ce que n'est pas souvent un programme ;

3. La nécessité d'un *suivi* ou historisation des événements au fil du temps, en particulier pour assurer un meilleur service à la clientèle, le pilotage des opérations mais aussi pour mieux connaître la vie des produits ou services durables livrés ;
4. Le caractère *documentaire* des informations échangées aux frontières du système si on définit le document comme un *ensemble coordonné et évolutif dans le temps de données*, fixes et variables, sans méconnaître sa forme de restitution ou le caractère potentiellement multimédia de ces données. C'est bien aux frontières des SO et SI que souhaits, commandes, confirmations, factures et paiements accompagnent les produits et services livrés ou reçus. Par extension, c'est bien aux frontières des sous-systèmes organisationnels et informatiques que s'échangent et se partagent dossiers et documents, papiers ? ou électroniques. Avec une granularité plus fine encore, l'échange entre unités organisationnelles traverse les frontières de zones de responsabilité. Les ensembles de données circulants dont la durée de vie est limitée au traitement d'un événement déclencheur, qu'on les appelle documents, messages ou formulaires (électroniques), sont aussi importants pour le SO que les ensembles de données référentielles ;
5. les fonctions de supervision et contrôle nécessaires au pilotage des systèmes qui doivent survivre, certes en s'adaptant, aux mouvements divers dans l'environnement en particulier leur (dé) centralisation ou distribution. Il est frappant d'observer combien les changements dans la répartition des missions laissent intactes la nature et la forme des interactions élémentaires, encore plus la supervision.

Pour résumer, les éléments de structure des SO et SI qu'il semble pertinent de relever pour une flexibilité mieux garantie sont d'ordre événementiel, humain, historique, documentaire et managérial.

Remarquons enfin que les périmètres des SO et SI ne se superposent maintenant que rarement, sauf à se référer encore aux modèles historiques qui tracent des frontières qui se voulaient naturelles entre fonctions de l'entreprise. La matricage des responsabilités, l'accélération des cycles et l'agilité souhaitée ont non seulement rendu le SO plus mouvant que le SI mais aussi et surtout décalé par rapport à lui. Le pilotage en souffre. En toute hypothèse, c'est un fait d'expérience que cette non concordance soit devenue permanente. Elle s'ajoute aux cinq caractéristiques structurelles mentionnées plus haut pour rendre particulièrement pertinente la progressivité de la mise en œuvre. En terme d'ingénierie, il y faut des possibilités d'assemblage et de réassemblage des composants des SI. C'est donc une exigence de réutilisation qui se présente

Récurtivité et réutilisabilité

Le processus est une notion fort ancienne qui, pour ne parler que de ses dernières acceptions, recouvre le progrès des maladies, le déroulement des actes juridiques puis les procédés de fabrication. Il est pour nous maintenant la représentation d'une structure organisationnelle et informationnelle supplémentaire qui s'ajoute à la liste non limitative déjà mentionnée des missions, des produits, des opérations, des canaux, des partenariats, des plates-formes. Le processus est une suite ordonnée d'opérations concourant à la même fin. Le processus est un outil de modélisation complémentaire de l'outil système⁶. Il exprime la dynamique du système ; il est mis en œuvre quand des entités en interaction ont été identifiées dont on veut restituer des relations d'ordre logique et/ou chronologique. Il s'agit à proprement parler d'un modèle des traitements, ceux déclenchés par les événements⁷.

⁵ A titre d'exemple, citons l'introduction du « workflow humain » chez WebMethods ou des processus dans Biztalk chez Microsoft.

⁶ Sur le plan formel, la modélisation du processus relève des réseaux de Petri. Elle peut être décrite textuellement, par exemple grâce au méta langage qu'est XML et avec des balises adaptées (Wf-XML, ebXML, etc.)

⁷ Définir le processus comme la (fonction de) transformation d'intrants en extrants n'est pas incompatible avec celle retenue ici mais elle est singulièrement plus étroite puisqu'elle ne peut induire le caractère récursif dont nous parlons plus loin.

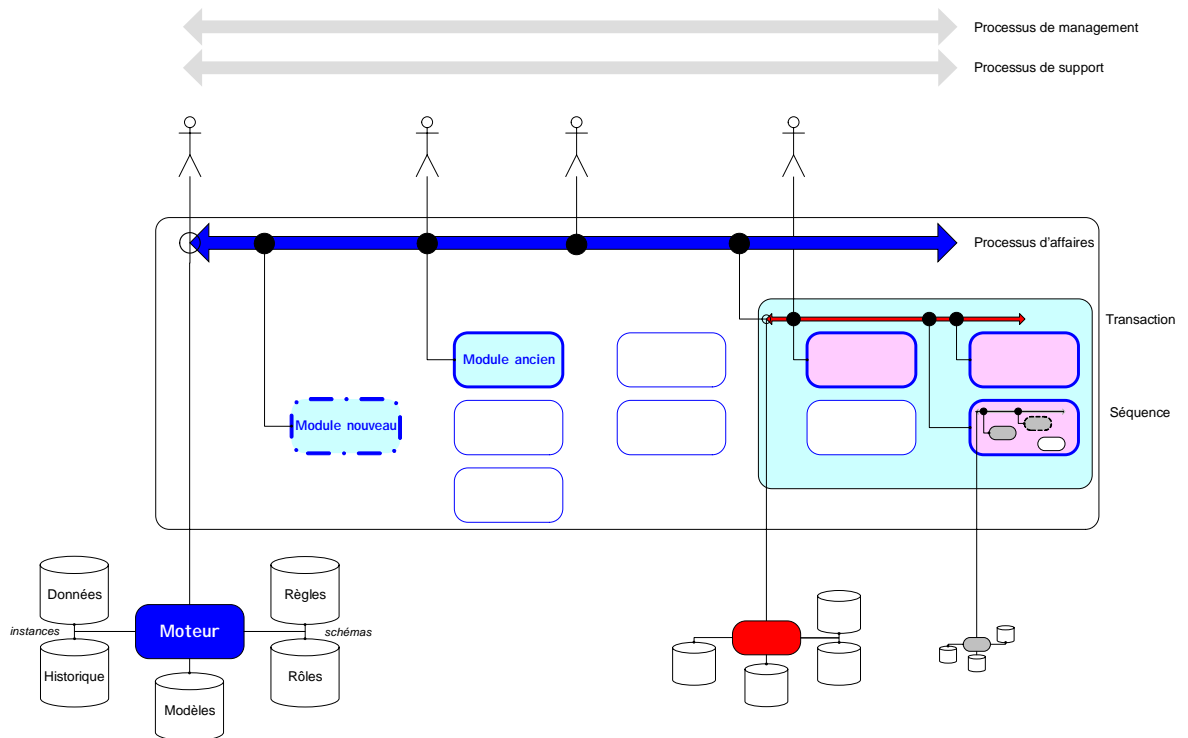


Figure 15 – Le réalignement des applications sur les processus transversaux à l'organisation

Applications existantes, modules nouveaux et activités humaines sont rassemblés, à plusieurs niveaux, en processus modélisés, instanciés, actés, historisés et supervisés sous le contrôle de moteurs ou séquenceurs qui en traitent les règles de structures. Les transactions informatiques peuvent être regroupées pour correspondre aux transactions économiques ou aux collaborations entre unités organisationnelles. Les processus peuvent s'étendre à l'entreprise virtuelle. Le système Informatique est une ressource pour le processus qui est un guide de travail et/ou une commodité pour les acteurs.

Ici encore, il s'agit d'une structure associative. Celle qui lie, nous semble-t-il, les éléments des SO et SI déjà mentionnés qui subsistent quand tout change : l'état d'un « dossier » dont on doit suivre l'évolution pour finaliser un acte de gestion au travers des diverses activités humaines ou opérations automatisées utilisant des ressources. Cette structure pallie à la non concordance entre SO et SI puisqu'elle tient compte de la répartition des activités et opérations entre les fonctions du SO et les programmes du SI. Ici le dossier est presque une métaphore puisqu'il s'agit des documents portant la trace des événements survenant aux frontières des (sous) systèmes. La perspective est manifestement temporelle, celle qui a toujours manqué dans les représentations informationnelles pratiquées comme d'ailleurs dans les systèmes d'exploitation.

La notion de processus permet de construire un artefact, le SI, cohérent avec les opérations de l'entreprise représentées et avec les règles de gestion choisies et modélisées elles aussi par des processus. Il est très important de formaliser par des modèles explicités les représentations retenues pour les opérations de l'entreprise et les étapes de pilotage ; ces représentations ne sont pas uniques pour la même réalité mais doivent être comprises et partagées par tous les acteurs concernés.

Comme pour toutes les structures précédemment évoquées, la structure temporelle permet un découplage entre les entités et les associations, ici entre les acteurs et les activités ou opérations, disons les étapes pour aller vite. Ce découplage permet donc une description autonome des étapes par rapport aux processus qui les utilisent. Comme une étape peut être détaillée et ainsi devenir elle-même un processus, le processus a un caractère récursif. La récursivité qui a tât fait de devenir ingérable dans les représentations sémantiques ou logiques, y compris les modèles de traitement informatiques, est à comprendre comme une instanciation dans les représentations temporelles. Pour la modélisation, la récursivité s'exprime par l'imbrication des processus et sous processus.

Toute la puissance de la notion de processus au regard de l'exigence de réutilisation est ici définie. Pratiquement, un cycle d'approbation peut très bien devoir instancier un autre cycle similaire ou identique dans l'échelle des responsabilités, pour le même dossier ou pour un nouveau dossier subsidiaire. On comprend donc l'émergence actuelle du thème de la gestion des processus d'affaire,

BPM (« Business Process Management » en anglais). En effet, l'ingénierie organisationnelle ou informatique est potentiellement très favorisée par un référentiel de processus correctement identifiés, tant économiquement que techniquement. Contrairement aux bibliothèques pléthoriques d'objets, paquetages ou patrons définis sur des bases sémantique et logique, la réutilisation des structures temporelles ou cycles est nettement favorisée par leur généricité. Celle-ci provient directement de la nature organisationnelle de ces structures. Au-delà de l'approbation déjà évoquée citons l'escalade (des problèmes), la publication (sur le Web), la consultation (d'un expert), la collaboration ou la notification (d'un partenaire interne ou externe).

La réutilisation dans les SI porte donc sur la gestion technique mutualisée des structures processuelles, les « workflow » ou chaînes de tâches, comme des structures de données ou documentaires mais aussi sur la gestion organisationnelle et fonctionnelle des processus eux-mêmes.

De forts enjeux économiques et professionnels

Dans cet article, nous avons présenté la notion de processus comme une variante de la notion de système qui s'intéresse à sa temporalité et à d'autres aspects de sa structure. Elle permet de construire progressivement un SI aligné avec les formes organisationnelles et règles de gestion de l'entreprise et apte à évoluer en rapport avec elles. En rapport aussi avec la généricité de ses modes d'échange et de fonctionnement, elle permet de prolonger les efforts d'urbanisme par la mise en œuvre de logiciels génériques permettant d'espérer un haut niveau de réutilisation de ses composants. Les coûts de constitution et de maintien en condition opérationnelle des systèmes informatiques doivent donc pouvoir baisser considérablement, certainement plus que ce qu'avaient laissé espérer les techniques programmatiques les plus récentes, à tort d'ailleurs.

Si les processus d'affaire peuvent être très marqués par le ou les métier(s) de l'entreprise, les processus de support et de management ou les sous processus d'affaire plus détaillés ne le sont pas souvent. Le même processus d'escalade se pratique aussi bien dans les services de support technique que dans les services financiers pour l'octroi de prêts ou le traitement des sinistres. Les grands éditeurs l'ont bien compris, par exemple pour la « verticalisation » à peu de frais de leurs solutions de CRM⁸, SRM ou SCM ou encore en ce qui regarde l'offre des EAI et autres « middleware », sans parler des voies ouvertes pour la collaboration entre systèmes et pour la mutualisation de services Web.

Gérard SACCONI
Professeur, ESIEA



Les référentiels de connaissances

Travail collaboratif et intelligence collective

La gestion des connaissances est considérée aujourd'hui comme un enjeu majeur de l'entreprise, enjeu économique, et plus encore, condition de sa viabilité dans un environnement évolutif et complexe. Les nouvelles technologies de l'information et de la communication y sont naturellement associées et accréditent l'idée d'un déploiement rapide sur le terrain. Cependant, ce domaine que l'on rattacherait par sa dénomination aux sciences de la gestion, relève bien plus des sciences cognitives et des sciences sociales. Du coup, les problématiques qui sont abordées ne sauraient être réduites à de simples considérations de gestion de ressources, de stocks, ou d'échanges, et surtout si l'on s'intéresse à l'impact attendu de ce processus sur l'organisation : ajouter l'intelligence collective à l'expertise individuelle. La conception même des outils devant aider à atteindre ce niveau de performance, doit être fondée sur des concepts non triviaux¹ et exige de véritables expérimentations en grandeur réelle. C'est dans cet esprit que nous avons travaillé sur le concept de « Référentiel de connaissances » dont les fondements sont ici rappelés.

Introduction

Les technologies de l'information facilitent aujourd'hui la communication et l'échange de données, indépendamment des localisations géographiques des serveurs et des acteurs. La vague Internet, qui en constitue la partie la plus visible avec ses technologies conviviales, a remis au premier plan des objectifs ambitieux pour les organisations : incitation à l'innovation, utilisation généralisée du retour d'expérience, partage des savoirs individuels, argumentation et décision collectives.

Cependant, les performances technologiques des systèmes d'information ne suffisent pas à l'émergence d'un réel mode de travail collectif, où l'ensemble des processus cognitifs d'apprentissage, d'argumentation et de décision, attribués à un groupe d'acteurs, dépasseraient un niveau de performance atteint isolément par chacun des individus.

Il semble que l'objectif d'une meilleure performance collective nécessite de repenser la dialectique individu / collectif ; il s'agit de mieux reconnaître l'existence et l'intérêt des actions et interactions individuelles, tandis que dans nos organisations domine généralement l'idée d'une norme organisationnelle composée de principes stratégiques, de procédures et de consignes prédéterminés, applicables uniformément par tous. D'un autre côté, l'innovation que l'on tend parfois à considérer comme seulement issue du génie individuel repose tout autant sur la performance collective que sur les compétences individuelles.

Pour atteindre le niveau d'une intelligence collective, il est nécessaire d'instrumenter ce sujet collectif qu'est une organisation constituée en vue de l'action (une équipe de projet par exemple) afin de le doter de capacités analogues à celles – naturelles – d'un individu cognitif. On peut en effet parler d'une entité collective - une entreprise, par exemple - comme étant susceptible d'apprendre, d'avoir une mémoire, de faire des expériences, des choix, de prendre des décisions, d'agir, etc. - c'est à dire, comme étant en quelque sorte dotée d'une cognition individuelle [1].

En tout premier lieu, il s'agit de doter le collectif d'une « mémoire commune », et ce, quels que soient les métiers concernés, les spécialités, les fonctions et la nature des connaissances nécessaires à l'exécution des tâches. C'est sur la base de cette mémoire partagée (et non pas commune) que pourront s'édifier des processus collectifs d'apprentissage, de conception et de décision. Comme le remarque très généralement Pierre Lévy [2], « le savoir de la communauté pensante n'est plus un savoir commun, car il est désormais impossible qu'un seul humain, ou même un groupe, maîtrise toutes les connaissances, toutes les compétences, c'est un savoir collectif par excellence ».

¹ Action située et cognition collective

La nature de cette mémoire collective mérite cependant d'être discutée, la cognition individuelle étant bien différente de ce que l'on peut appeler la cognition collective.

Dans le cas d'un individu, il est établi que l'activité mémorielle n'est pas totalement consciente, c'est-à-dire qu'elle ne se réduit pas à des processus de remémoration délibérée et intentionnelle. Certains mécanismes automatiques de l'esprit humain sont à l'œuvre, (on se souvient mieux d'un mot si on a eu à l'utiliser auparavant) et leurs effets sont pondérés par d'autres mécanismes : la construction de l'image mémorielle est tributaire des conditions de sa production (les faits inhérents au contexte d'action sont associés aux faits initialement mémorisés).

Dans le cas d'un « sujet collectif », la situation est tout autre ; il n'existe pas de mécanismes automatiques, en dehors d'un inconscient collectif jouant le rôle d'un système de valeurs, et dans le processus mémoriel, la phase de reconstruction dynamique est prédominante : les connaissances communes doivent être recapturées une par une pour être utilisées dans le jugement et la décision. Et ceci d'autant plus que l'encodage de l'information n'est plus réalisé par le même individu. Il s'agit pour chacun d'utiliser aussi « les souvenirs des autres » en participant à un apprentissage collectif.

En résumé, la cognition, et encore moins la cognition collective, ne peut pas être réduite à un stockage symbolique des informations sous forme brute (mémoire-entrepôt) ou sous forme de représentation (mémoire-empreinte). Elle se fonde toujours sur une boucle d'actions et nous nous intéresserons donc en priorité aux connaissances relatives à l'action, que l'on peut désigner par l'expression connaissances actionnables.

Connaissances actionnables

Ce néologisme, comme le rappelle J.L. Le Moigne [3], a été introduit dans la littérature organisationnelle par D. Schön en 1983 (actionable knowledge), afin de dépasser le distinguo habituel entre savoir et savoir-faire, c'est-à-dire la séparation entre la composante épistémique (la connaissance) et la composante pragmatique (l'action). C. Argyre, qui dès 1974 a proposé avec D. Schön des modèles organisationnels pour l'action, parle en 1995 de « savoir pour agir » (knowledge for action). Il ressort de ces travaux que c'est par la réflexion sur ses actions, sur ses savoirs d'expérience, que le praticien (le sujet connaissant engagé dans l'action) peut mieux prendre conscience des stratégies d'actions qu'il a élaborées, et donc pourra les améliorer. La traduction de savoirs tacites en savoirs d'action constitue le cœur même du processus d'apprentissage.

En développant le concept dual d'action intelligente, H.A. Simon avait attiré l'attention sur la légitimité d'une rationalité procédurale, qui ne se réduit pas à la logique et au calcul fondé sur des faits ayant valeur de vérité, et qui tient pour tout aussi satisfaisants les modes cognitifs que sont la délibération et l'argumentation. Il s'agit alors de concevoir et construire des connaissances qui soient représentations d'expériences, d'actions, de réflexions, de délibérations.

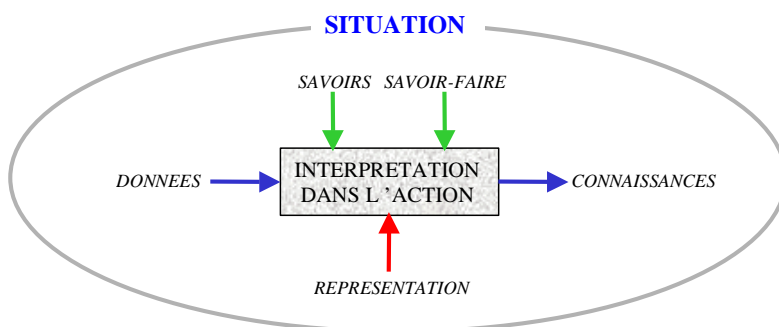


Figure 1: Connaissance actionnable.

Dès lors, la gestion des connaissances actionnables n'est pas une encapsulation du savoir, mais un suivi dynamique d'un corpus de connaissances en expansion. Elle est définie comme « processus de partage dynamique de connaissances [tacites] utiles à l'action [collective] ».

Les référentiels de connaissances

La problématique que nous devons traiter est celle de la gestion des connaissances scientifiques et techniques utilisées par les organisations pour leurs décisions d'action. Parmi toutes les démarches envisageables en gestion des connaissances, on constate deux tendances :

- la capitalisation des savoirs et du savoir-faire, dont l'objectif principal est de consigner les connaissances stratégiques, et qui porte donc l'effort sur la sélection et la structuration des connaissances ;
- le partage dynamique des connaissances qui ne préjuge pas de leur utilisation future, et n'élimine pas des connaissances dont l'intérêt pourrait être révélé plus tard.

Parmi les sciences relatives à la cognition², c'est la discipline appelée « Intelligence Artificielle » qui a développé la première voie avec une ingénierie des connaissances fondée sur la constitution de structures de données exploitables et validables dans les domaines techniques où une expertise stable peut être dégagée. La mise en forme de cette expertise est conditionnée par la finalité : l'implémentation en machine à traiter l'information.

L'utilisation massive des réseaux d'information et de communication a relancé récemment l'intérêt de constituer des corpus d'éléments de connaissances dynamiques partageables entre acteurs humains. Mais alors, c'est la capacité à suivre l'évolution des connaissances qui est recherchée dans les systèmes d'information.

Les référentiels de connaissances explorent en priorité cette seconde voie par une organisation de l'information à deux niveaux : les sources d'information et les éléments de connaissances. Un corpus d'éléments de connaissances recouvre un domaine de connaissances qui est aussi un domaine d'action (de décision).

Dans le domaine de l'ingénierie des connaissances, on appelle savoir élémentaire une « donnée informative qui prend du sens dans un contexte ». Un savoir élémentaire se distingue donc d'une simple information par le fait qu'il comporte une part d'interprétation liée à la personne qui l'énonce. Si cette interprétation tend à rendre l'information utile à l'action (on lui associe une pertinence et une utilité), on peut parler d'élément de connaissance [actionnable].

L'élément de connaissance explicite : une donnée informative jugée utile, une trace de raisonnement, une unité de sens partageable et réutilisable.

L'élément de connaissance est identifié par un triplet daté ECt {objet informatif, source, rédacteur), où l'objet est l'information issue d'une source documentaire ou donnée directement par le rédacteur. Cet objet d'information est interprété par le rédacteur, sous forme d'un commentaire où il exprime un point de vue ou une recommandation. Sous sa forme la plus simple, il s'agit d'un paragraphe factuel, avec référence à la source d'information. On peut si nécessaire lui donner une structure plus formelle, en référence à un modèle de représentation.

Le corpus de connaissances est ainsi géré de façon dynamique sous forme de graphes se développant dans le temps, dont les nœuds sont les éléments de connaissances et dont les arcs figurent les relations entre les ECt.

² au sens restreint : processus des couches supérieures de la pensée humaine, c'est-à-dire manipulant des concepts, des schémas, des connaissances et des processus explicites (langage...)



Figure 2: graphe d'éléments de connaissances.

Un ECT peut être relié aux autres éléments de connaissances (issus d'autres rédacteurs) par des relations d'association (complément, opposition, contradiction) à des ECT+i postérieurs, des liens de chronologie (lien voir_aussi vers un ECT-j antérieur) ou des liens de mise_à_jour (révision de son point de vue par le rédacteur même de l'ECT).

Pour chaque domaine de connaissance, on construit un référentiel, mais selon la problématique abordée dans le projet, on parlera de référentiel de sûreté, de référentiel qualité, de référentiel technique ou encore de référentiel de conception.

Un environnement logiciel de gestion de référentiels de connaissance doit faciliter le parcours du corpus de connaissances (navigation) mais aussi permettre de suivre l'évolution du corpus de connaissances (graphes dynamiques), d'améliorer la visibilité du domaine abordé (cartographie des connaissances) ou encore d'organiser l'utilisation des connaissances pour l'action (composition des connaissances, synthèses).

Retour d'expériences : une typologie

Afin d'expérimenter sur le terrain les concepts relatifs à l'intelligence collective, l'URC a conçu un environnement informatique de travail collaboratif³ fondé sur les objectifs suivants :

- Transformer les échanges d'informations en argumentation utile à l'action (aide à la décision) ;
- Organiser le partage des connaissances (corpus dynamique d'éléments de connaissances) ;
- Catalyser l'émergence d'une intelligence collective et l'amplifier.

Cet environnement a été utilisé dans des contextes différents de travail collectif où chaque acteur a des activités orientées vers les autres (négociation, reformulation) pour contribuer à des activités collectives (recherche, conception, décision...), mais toutes fondées sur la construction de représentations partagées [4].

³ cet environnement a reçu le prix INTRANET'99 décerné par le Ministère de l'Industrie.

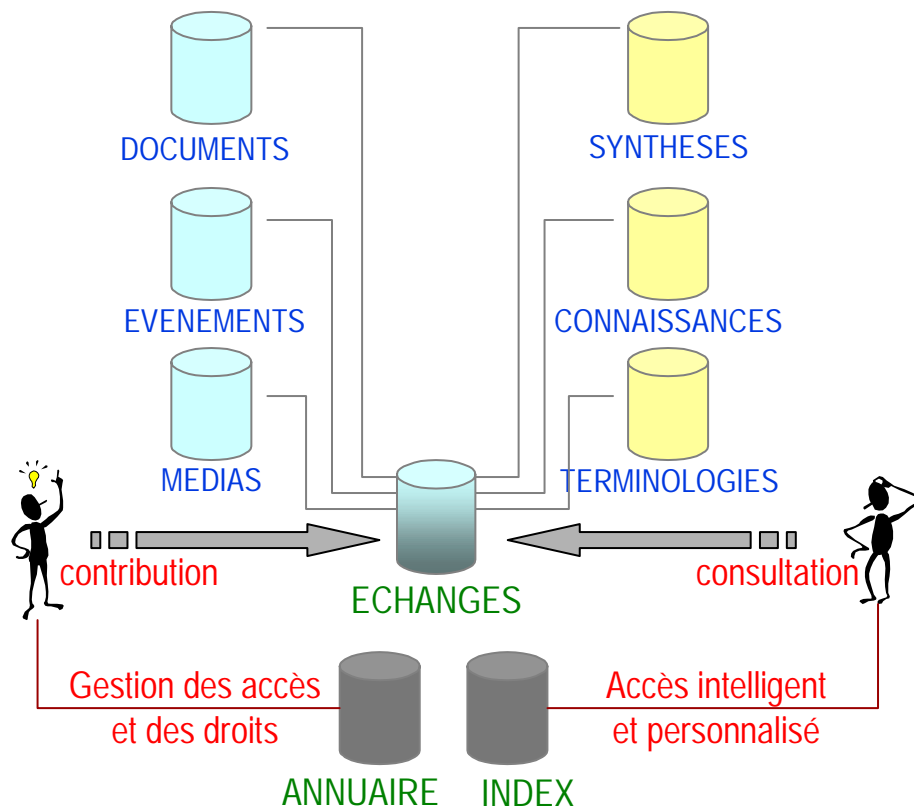


Figure 3 : l'architecture d'un référentiel de connaissances

Dix-huit plates-formes au total ont été déployées dans huit organisations différentes, mais qui ont en commun les caractéristiques suivantes :

- Travail distribué entre équipes ou individus dispersés géographiquement ;
- Projets, cycles industriels ou technologiques de durée notable (un an au moins) ;
- Connaissances produites dans un contexte organisationnel peu structuré.

Cet ensemble d'expériences en grandeur réelle (collectifs de 15 à plus de 200 personnes, serveurs hébergeant jusqu'à 3 000 documents partagés) a permis de construire une typologie des modes de travail collectif, et une différenciation fonctionnelle en trois classes de référentiels. Dans chaque classe, des distinctions peuvent être faites selon le critère de cohésion.

Pour chaque mode de travail collectif, on a pu constater d'importantes variations dans le niveau de partage et même dans la nature de ce qui est partagé : une identité, une culture, des biens et des services, des connaissances, des informations, des expériences, des risques ou bien simplement... de l'estime.

Travail en groupe

Le groupe est pris ici comme rassemblant des individus fortement dépendants, c'est-à-dire en interaction (parfois continue) dans l'action. L'identité du collectif est ici donnée par le contexte (l'institution), et on peut distinguer trois sous-classes.

Le mode-projet correspond au travail coopératif dont la cohésion repose sur l'adhésion de chacun des acteurs. Un cas particulier réside dans le mode-plateau qui regroupe tous les acteurs en un même lieu et donc minimise les fonctions de gestion de communauté virtuelle (forums de discussion, identification des acteurs).

Plus général est le mode-équipe dont les membres peuvent contribuer à des projets différents, mais qui partagent une culture et des ressources communes ; le critère d'appartenance fonde ici la cohésion. Dans le travail de groupe, les fonctions de sécurisation des échanges sont minimales, les règles de partage étant explicitement fixées par l'institution.

Travail en communauté

Une communauté regroupe des individus autonomes engagés dans un processus de socialisation, volontaire ou provoqué. L'identité commune doit ici être construite et émerge de l'interaction des acteurs. Le critère de cohésion est ici très différenciant.

Fondé sur la reconnaissance de la valeur de chaque membre, le travail collectif s'effectue alors en mode concurrentiel : partage de ressources communes (financements, ressources humaines, matériel). Les fonctions de gestion du partage, de gestion des accès et de sécurisation doivent être particulièrement développées.

Le travail en communauté peut également procéder du mode associatif où la motivation des membres soutient l'œuvre commune. Les fonctions de synthèse des connaissances, et de cartographie dynamique des contributions sont ici très recherchées.

Le seul critère de convivialité peut suffire pour constituer des communautés émergentes en mode interactif (les forums internet en sont le meilleur exemple). La règle est ici du partage total, et le processus de socialisation y est le plus apparent.

Travail en réseau

Le travail en réseau repose sur la mise en relation d'individus à la fois autonomes (ils ont chacun leur sphère d'action propre) et indépendants, visant des objectifs individuels.

Le critère de regroupement peut être la cooptation, caractéristique des réseaux d'experts fonctionnant sur le mode collégial. Paradoxalement, le niveau de partage des connaissances peut être dans ce cas assez faible, l'échange d'informations restant privilégié. Le sujet collectif laisse la place à une collection d'individus opérant en fonction de leurs représentations personnelles.

Le réseau d'échanges adopte le critère de valorisation pour fonctionner sous le mode de la réciprocité. Les échanges interpersonnels finissent par former un construit collectif. Ce type de réseau suppose des processus d'authentification stricte des acteurs.

Enfin, le rassemblement d'acteurs réunis par des intérêts communs constitue le cas particulier du mode coalition (exemple des consortiums, ou des programmes d'intérêt collectif). Les acteurs adoptent un objectif général, mais fonctionnent selon le principe de subsidiarité. L'objectif général n'est pas censé être atteint par la planification centrale et détaillée des tâches, mais grâce à la contribution des projets particuliers, avec leurs objectifs respectifs.

Conclusions

La diversité des expérimentations menées a permis de dégager quelques principes généraux, mais également de « bousculer » certaines idées reçues. Parmi celles-ci, la plus évidente est que la collaboration ne se crée pas spontanément même lorsqu'on partage un même espace de travail. La seconde est que la seule interaction entre acteurs n'est pas suffisante pour l'émergence d'une identité collective, et la réalisation d'un construit collectif constitue le premier pas vers l'intelligence collective.

L'impact sur la conception d'une plate-forme de travail collaboratif est qu'il faut compléter les fonctions de gestion documentaire et de gestion des connaissances par des fonctions de gestion de communauté tout aussi importantes : gestion du partage, de la confidentialité, des accès, services de communication instantanée et différée.

Enfin, l'intérêt a été souligné d'implanter sur la plate-forme des mécanismes analogues aux mécanismes cognitifs de traitement dynamique des connaissances : extraction, filtrage, recomposition.

J.M. PENALVA - URC EMA – CEA
LGI2P, Parc G. Besse, 30035 Nîmes cedex 1
Jean-michel.penalva@ema.fr

J. MONTMAIN - URC EMA – CEA
LGI2P, Parc G. Besse, 30035, Nîmes cedex 1
Jacky.montmain@ema.fr

L'Unité de Recherche sur la Complexité (URC), unité commune EMA – CEA, a été créée par le commissariat à l'Énergie Atomique (CEA) et l'École des Mines d'Alès (EMA) pour étudier prioritairement l'impact des sciences de l'information et de la communication sur les processus organisationnels (distribution des systèmes d'information, partage des connaissances et décision collective), et sur le facteur humain (interaction homme-système et défaillance systémique, risque décisionnel).

Bibliographie

- E. Andrieux (1998) Langage et construction de la cognition collective. Grand Atelier MCX, Poitiers, 19-20 novembre 1998
- P. Lévy (1991) L'idéographie dynamique, vers une imagination artificielle. Éditions La Découverte, Paris, 1991
- J-L. Le Moigne (1998). Connaissance actionnable et action intelligente, Grand Atelier MCX, Poitiers, 19-20 novembre 1998
- R. Teulier-Bourguin : les représentations, médiations de l'action stratégique, in La stratégie chemin faisant, M.J. Avenier, Ed. Economica, 1997



STIC

De quoi parlons-nous ? Comment en parler ?

La communauté informatique et, plus largement, celle des STIC (sciences et technologies de l'information et de la communication) est une grande créatrice de vocabulaire. Maîtriser les mots, n'est-ce pas un des points-clés de la maîtrise des systèmes d'information, objet même d'ADELI. Dans le dialogue qui s'établit entre un maître d'œuvre et un maître d'ouvrage, les deux partenaires doivent trouver les mots pour se parler. Entre le jargon « métier » et le jargon « technique », il faut non seulement trouver des correspondances, mais innover. Les « objets » que manipulera le futur système d'information ne sont jamais tout à fait donnés à l'avance !

Mais charité bien ordonnée commence par soi-même, et le thème de la journée organisée par l'ASTI¹ le 15 mai 2002 entendait dresser un panorama des problèmes de vocabulaire rencontrés par la communauté tant pour parler d'elle-même et de ce qu'elle fait.

Ouverture

Yves Lecourtier, professeur à l'Université de Rouen (spécialiste de l'analyse de documents et de la reconnaissance de l'écrit) montre d'abord la variété des enjeux portés par les mots, depuis la propagande commerciale jusqu'aux options philosophiques. C'est ce qui rend difficile une épistémologie du domaine, et qui pousse certains à refermer le problème sur le domaine, a priori plus sûr, de la triade signal/information/concept.

Informatique et communication

Dans un premier temps, l'ASTI, qui vise d'abord à rassembler, a largement ouvert les frontières en accueillant largement tous les mots de la communauté dans son « répertoire », disponible en ligne. On y trouve aussi bien des termes historiques oubliés depuis longtemps (qu'est-ce qu'un « binoquet ») que des sigles tels que XBRL... Le cas échéant, le répertoire laisse entendre qu'il s'agit d'un pur « buzzword » et en suggère un équivalent moins branché peut-être, mais plus pertinent et plus précis.

Les plus hautes instances de la recherche spécialisée en STIC apportent leur point de vue. Francis Jutand, directeur du département STIC du CNRS, présente notamment ses programmes thématiques, dont les intitulés mêmes balisent le champ de nos recherches.

Claude Guegen (Groupe des grandes écoles de télécommunications) insiste sur la « logique de l'usage » bien en phase avec l'accent mis aujourd'hui sur « la France d'en bas ». Jean-Pierre Verjus confirme ces approches en donnant le point de vue de l'INRIA.

Les problèmes de terminologie

Thérèse Hardin, au nom de SPECIF (.Société des personnels enseignants et chercheurs en informatique de France) cherche à définir l'informatique à partir de la « mécanisation » du calcul. Mais elle élargit ensuite son propos : « Si notre métier est mal perçu, c'est aussi parce nous n'assurons pas suffisamment la communication sur ce métier. Donc il nous faut mieux expliquer notre sujet, les résultats de notre activité de recherche, avec nos mots techniques. Notre vocabulaire doit certes s'adapter au public, mais les définitions doivent rester suffisamment précises. ».

Jean-Yves Gresser, un des fondateurs de la SFT (Société française de terminologie) et informaticien d'entreprise (Groupe Euler-Sfac) insiste sur le fait que tout professionnel est aujourd'hui un terminologue. Les informaticiens, en particulier. Et qu'ils doivent donc « être pros » dans cette fonction, et pour cela se brancher sur les sites spécialisés en terminologie.

¹ Association Française des Sciences et Technologies de l'Information - <http://www.asti.asso.fr>
Les Adéliens peuvent participer aux activités de cette association en assistant aux manifestations ou en adhérant.

Colette Hoffsaes, co-fondatrice du CREIS (Centre de coordination pour la recherche et l'enseignement en informatique et société) développe les impacts sociaux et met l'accent sur le fait que « le vocabulaire technique tend à masquer les enjeux politiques et sociaux ».

La recherche des fondements logiques

Gilles Dowek (INRIA), logicien, situe l'informatique dans les sciences et leur rapport avec la réalité, en partant du schéma de Kant pour montrer la situation tout à fait originale de l'informatique dans ce paysage.

Gérard Montsény, automaticien (LAAS-CNRS), rattache à l'information et son traitement à ses bases physiques et en particulier à la thermodynamique. Une vue abstraite qui trouve pourtant son illustration dans la vie concrète des entreprises et des organisations comme le montre l'exposé suivant de Pierre Descoins (Ministère de la Santé).

Des « sciences dures » à celui des « sciences molles ».

Eddie Soulier (Université de Troyes) brosse un panorama des approches. Thomas Baudel (Ilog, président de l'Afihm) analyse la question du point de vue de l'ergonomie.

Enfin, deux grandes catégories d' « utilisateurs » de notre vocabulaire concluent la journée.

Jean-François Pépin, délégué général du Cigref, montre comment les directions informatiques voient la question et note : « Les mots clés de la DSI sont : alignement stratégique métiers, urbanisation des systèmes d'information, gouvernance, management des processus ». D'autres mots, certes, que ceux de l'enseignant en informatique au sens traditionnel !

Jacques Baudé, président d'honneur de l'EPI (Association Enseignement public et informatique), conduit l'assistance au coeur des salles de classes, donne des exemples du vocabulaire que les élèves (enseignement secondaire) sont censés assimiler, mais regrette « la très insuffisante formation des enseignants aux STIC, à la fois en formation initiale (au moment où l'on s'apprête à renouveler la moitié du corps professoral) mais aussi en formation continue. ».

Outre le fait que les travaux de la journée sont ou seront disponibles en texte intégral sur le site de l'ASTI, ceux-ci avaient pour objectif de fournir un programme d'action pour le groupe de travail « Portail et dictionnaire » de l'ASTI. Un projet ouvert et communautaire, auquel, bien entendu, les animateurs du groupe espèrent bien qu'ADELI apportera sa contribution.

Pierre Berger
Vice-président de l'ASTI (Communication)
www.asti.asso.fr



De l'idéal à la réalité

Cet article présente quelques réflexions autour de la question suivante : « Comment présenter des plans d'action réalistes, sans renoncer aux ambitions de la cible ? »

Cette question se traite sur deux plans : le premier, évident, est l'expression formelle des objectifs, le second, plus subtil et diffus, est celui de l'interrelation.

Il faut toujours suivre sa pente naturelle, pourvu que ce soit en montant.

André Gide, *Les Faux-monnayeurs*.

Introduction

Le contexte des grands projets

Par construction, les grands projets s'inscrivent dans une perspective à long terme. En eux, tout est grand : l'échelle du temps comme la dimension spatiale, étendue à la totalité du système d'information. Dans un tel contexte, la perception des objectifs par les acteurs impliqués fait souvent problème. En effet, il revient à chacun de situer son action, aussi ponctuelle et immédiate soit-elle, dans la trajectoire de l'ensemble des travaux. Faute d'une telle vision sans cesse réactualisée, les acteurs perdent de vue les enjeux qui, seuls, donnent le sens de leur action. La démotivation finit par se répandre dans la troupe.

Le contexte des projets applicatifs

Les projets applicatifs¹ rencontrent, également, cette problématique. Certes, leur objectif est plus rapproché que celui d'un projet au niveau du système. Mais, à travers leurs nécessaires communications, ils interagissent avec des catégories d'acteurs qui ne pensent qu'à la perspective à long terme. Ils sont, en permanence, confrontés à des univers de pensée qui peuvent paraître, sommairement, contradictoires par rapport à leurs préoccupations. Il s'agit, en tout premier lieu, des utilisateurs, lesquels ne jugent pas un projet sur ses objectifs officiels (le bon produit sous contrainte de délai et de budget), mais par rapport à leurs propres attentes et à leur intuition, pas toujours formulée, de la solution idéale. En second lieu, les projets applicatifs doivent compter, de plus en plus, avec d'autres acteurs dont la fonction consiste à représenter et faire avancer les visions à long terme. Ces gardiens du temps sont les architectes, les organisateurs, les stratèges ...

Le problème soulevé

Dans l'un et l'autre de ces contextes, projets systèmes ou projets applicatifs, le problème se pose d'une formulation claire et réactivée des finalités et des enjeux. En l'absence d'une telle formulation, les acteurs impliqués rencontrent les plus grandes difficultés à se situer les uns par rapport aux autres. Le risque est grand de voir l'énergie collective gaspillée en vaine agitation, gâchée par la démotivation ou, au moins, sous-exploitée. De nombreux acteurs prétendront s'occuper des mêmes aspects ; ils se lanceront dans des actions redondantes, voire contradictoires.

¹ J'entends par « projet applicatif », un ensemble de travaux aboutissant à la fourniture d'une application logicielle. L'application est supposée, de nos jours, non seulement fonctionner pour son propre objectif déclaré, mais aussi au service du système d'information, dans son entier.

Le syndrome est fréquent sur les grands projets : si un dispositif rigoureux ne précise pas le détail des responsabilités et missions, chacun voudra dire son mot sur tel et tel aspect. La confusion s'installe plus vite et plus sûrement que l'ordre ne s'instaure.

Que ce soit dans l'économie du projet ou dans le déroulement d'une simple réunion, maîtriser la communication suppose la coordination des points de vue. Pour cela – c'est une évidence –, il importe d'exprimer les objectifs. Encore faut-il que cette expression s'obtienne en termes assez précis pour être concrets et convaincants, mais assez ouverts pour n'être pas réducteurs. Le danger permanent est d'étouffer toute créativité collective ou individuelle sous le poids de la formule rigide qui fait loi. Au contraire, la maîtrise d'une communication ouverte et créatrice suppose de faire place aux représentations portées par les différents acteurs.

L'actualité du sujet

Ces considérations, posées ici avec la sécheresse du raccourci théorique, émergent à partir des expériences de grands développements, et elles apparaissent, aussi, avec une acuité particulière, dans les nouveaux types de projets que la société nous demande : ouverture sur le public (internet, CRM), fructification du patrimoine des compétences (*knowledge management*), évolution vers les architectures ouvertes et intégration de composants, prise en compte des évolutions structurelles et de la stratégie des entreprises ... Dans tous les cas, le bon déroulement des travaux demande de faire coïncider des points de vue très divers.

L'analyse du problème

Ménager leur place aux divers univers mentaux impliqués dans une action collective ne semble pas si difficile. Pourtant, de nombreux freins bloquent une si bonne intention. Tous, plus ou moins, sont de nature idéologique.

- L'analyse stratégique (Michel Crozier) nous sensibilise au jeu des intérêts divergents. Dans tout organisme, il y a toujours un point de vue dominant. Si on n'y prend pas garde, ce point de vue – le dogme – finit par inhiber toute pensée critique et, donc, par castrer la créativité. Ce processus fragilise l'organisation².
- Une perception plus anthropologique montre l'autocensure organisationnelle qui se met en place dans les organismes (avec des effets de seuils patents : plus l'organisme est grand, plus se restreint l'espace de la pensée libre).
- La culture d'entreprise est un facteur essentiel d'identification, donc d'adhésion. Mais, toute culture peut facilement fonctionner comme fermeture. Elle a, alors, l'effet néfaste de décourager l'expression et de stériliser l'imagination.

La trajectoire de progrès

Le principe

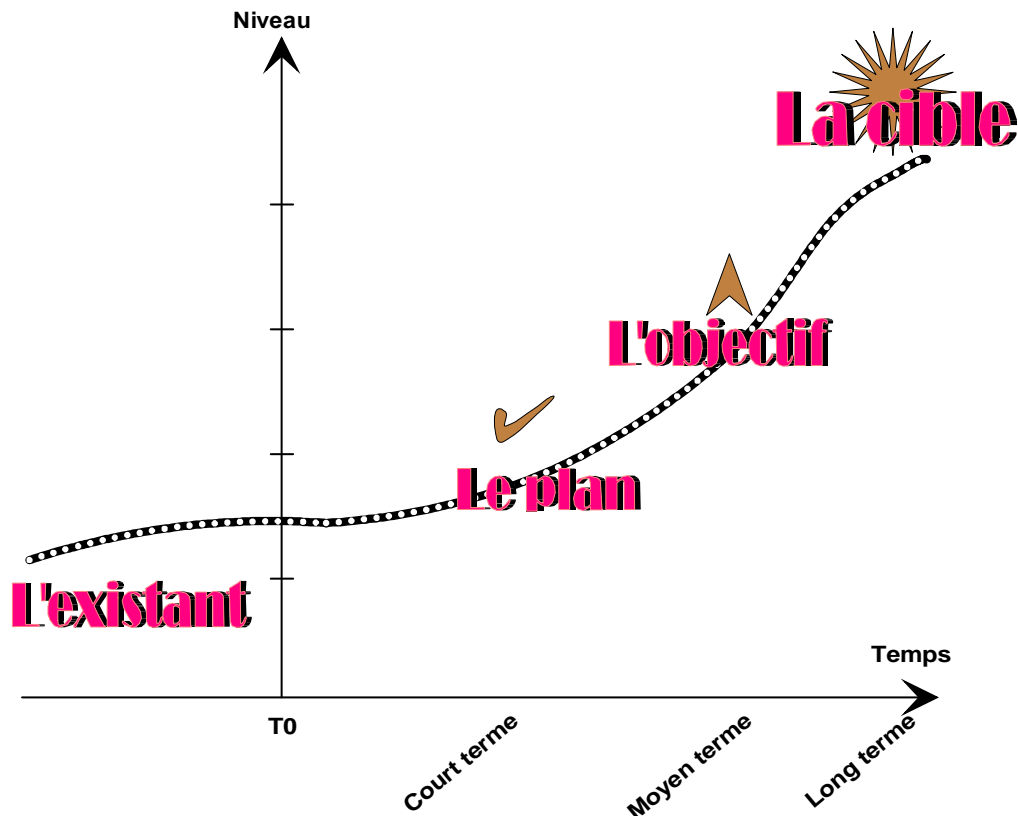
Le principe que nous proposons ici, comme tous les principes, est simple et évident³. En tirer les conséquences le sera moins. Il consiste à préserver, dans l'exposé des motifs d'une action, la hiérarchie qui va de l'idéal pressenti à la réalisation concrète. L'application de ce principe répond au souci de maintenir l'ambition et d'éviter la frustration. À la fois dans les moments d'un projet et à l'occasion d'une micro intervention telle qu'une réunion, nous devons nous entraîner à séparer les éléments en les accrochant à l'un des quatre objets suivants :

- La cible : « Les principes sont éternels. »
- L'objectif : « Les objectifs sont ambitieux. »
- La trajectoire : « Le plan d'action est réaliste. »

² Il est vrai qu'à l'inverse, une créativité débridée mettrait, également, en péril l'organisme. Tout est question de mesure et d'équilibre entre forces centripètes et centrifuges.

³ Pour citer Paul Valéry : « Ce qui est simple, est faux ; ce qui ne l'est pas, est inutilisable ».

- Les moyens : « Les moyens sont adaptés. »



La cible : « Les principes sont éternels »

Notre principe condense une première décision : il faut, à tout prix, éviter l'inhibition dans la définition des cibles. En cherchant, avec raison, à échapper aux délires complètement utopiques, nous risquons toujours de rogner les ailes de l'imagination. Nos entreprises ont, au contraire, besoin de cibles ambitieuses.

C'est en visant loin qu'elles se dépassent. Est-il nécessaire de rappeler une telle évidence ? Il semblerait que oui. En effet, dans les organismes, de puissants phénomènes contraignent la réflexion et stérilisent la créativité. Au sommet de la pyramide, sans doute les managers gardent-ils le cap sur une vision stratégique à long terme. Mais, au fur et à mesure que l'on descend le long de la ligne hiérarchique, les traductions successives à travers les univers dégradent le message initial. De plus, la vision stratégique ne suffit pas : il faut encore qu'à chaque niveau, elle « s'incarne » dans des formulations qui lui donneront de la consistance. La culture des entreprises est rarement favorable à ce processus. Au contraire, elle condense l'histoire de l'organisme et la sédimente à un niveau quasi inconscient. Son contenu officiel exprime les succès, sous la forme figée d'un catéchisme qu'il ne faut surtout pas questionner. Elle a aussi un contenu officieux, bien plus agissant et présent au quotidien. Cette petite musique parle, au contraire, de renoncement, d'autocensure, d'impossibilité ... Elle intervient au cours des réunions pour clore les débats par des sentences péremptoires qui terrorisent l'audience et renvoient chacun au silence : « On a déjà essayé. Chez nous, ça ne marche pas. Les décideurs ne comprendront pas, etc. ».

À rebours de cette tendance à l'inhibition, nous devons nous libérer : quand il s'agit d'élaborer une cible, c'est l'imagination qui commande. Ceci ne signifie pas que nous refusons les objections, mais simplement que nous les traiterons le moment venu. Ce premier principe (appelons-le « principe de dés-inhibition ») ne fait que généraliser, à l'échelle de l'organisation, la consigne principale des réunions de créativité (remue-méninges).

L'objectif : « Les objectifs sont ambitieux »

Si la cible doit être définie en termes idéaux et brossée de façon assez générale, elle doit s'analyser en objectifs précis. C'est le deuxième temps de notre démarche. Les objectifs sont coordonnés et rattachés à la cible : leur ancrage fait l'objet de communication. En effet, communiquer sur la cible seule, c'est souffler dans un pavillon sans embout. L'auditoire, incrédule, ne retiendra de l'exercice que l'idée d'une utopie creuse – au mieux. Les objectifs, au contraire, sont plus spécialisés, réduits à une dimension de la réalité collective : ils matérialisent les bonnes intentions de la cible en lui donnant un contenu tangible. Pour autant, les objectifs doivent insuffler quelque chose de la fièvre de l'ambition. En quelque sorte, ils font office de Janus : ils aident le passage entre l'idéal – perçu comme abstrait et inaccessible –, d'un côté, et le terrain de la lutte réelle, de l'autre.

La trajectoire : « Le plan d'action est réaliste »

À partir de la cible et en référence à celle-ci, un ou plusieurs objectifs sont définis. Certains sont retenus, d'autres repoussés ou différés. Il est difficile de s'engager sur une trajectoire générale qui mène à la cible : trop d'incertitudes rendent la planification illusoire. Si ce n'est pas le cas, c'est sûrement que la cible n'est pas assez ambitieuse ! En revanche, aux niveaux des objectifs, on retrouve l'univers habituel du management des projets. Sur ce terrain, les responsables bâtissent des plans d'action réalistes et rigoureux.

Les moyens : « Les moyens sont adaptés »

Enfin, la réflexion aborde le domaine le plus concret : celui des moyens. En tout premier lieu, il s'agit du budget, mais d'autres types de moyens sont à considérer, particulièrement la disponibilité des compétences attendues par les plans d'action. En fonction de ces calculs, les responsables effectuent les arbitrages qui, sans remettre en cause les objectifs, ni la cible, peuvent modifier les priorités et la trajectoire.

Par exemple, si, dans une hypothèse de trajectoire, les compétences nouvelles exigées sont peu assurées, le plan doit être révisé. Une solution peut consister à décaler l'objectif dans la liste des priorités, soit parce que les compétences sont en cours d'acquisition, soit parce qu'un autre plan d'objectif doit les construire. Bref, une trajectoire reste relative, ajustable ; on peut même, sans fausse honte l'abandonner (pourvu que l'on reste fidèle aux objectifs annoncés). Cela se nomme pragmatisme.

Les conséquences

La première motivation qui fonde ce principe de planification est la « dés-inhibition ». Sans tomber dans l'utopie stérile, l'imagination et la créativité doivent pouvoir jouer à fond pour bâtir des cibles vraiment originales, pour inventer des concepts novateurs, sinon pour créer de toutes pièces les futurs marchés.

Les organisations humaines sont ainsi faites qu'il est infiniment plus difficile de stimuler la créativité que, une fois activée, de la canaliser.

D'un point de vue formel, cette chaîne linéaire est assez bien traduite par les méthodologies de plan stratégique, schéma directeur et autre phase « *inception*⁴ » (première phase du RUP, *Rational Unified Process*). En revanche, sur le terrain, les choses sont un peu plus délicates. L'application du principe de dés-inhibition entraîne un impact important sur les relations dans l'entreprise. Nous nous plaçons, ici, au niveau microsociologique voire psychologique. La recette consiste à ménager sa place à chaque profil individuel. On cherche, en effet, à mobiliser tous les profils, à faire concourir toutes les compétences. Or, nous n'agissons pas dans une culture parfaitement ouverte et nous ne pouvons pas considérer que la rationalité démocratique ait atteint un haut niveau de maturité. Nous devons compter avec le choc des représentations. Avant même qu'interviennent les enjeux de pouvoir et les luttes d'influence, nous nous heurtons à un obstacle de taille : la différence des univers cognitifs. Par

⁴ Pour nous éviter un nouvel anglicisme, traduisons par « *impulsion* », terme qui rend bien l'idée.

exemple, pour l'architecte de SI et l'utilisateur ou le directeur Marketing, les mots ne signifient pas les mêmes choses ; le monde n'est pas structuré de la même façon. Pire : entre deux architectes, l'un autodidacte prudent, l'autre technolâtre futuriste, l'avenir n'a pas la même saveur ; de même entre un utilisateur qui répugne à se servir d'un « système » pas toujours fiable et un autre qui passe ses week-ends à « surfer ». Toutes ces populations se mélangent dans l'entreprise et leurs points de vue se heurtent avant que l'on puisse arrêter les décisions dont dépendra l'avenir.

Dans le quotidien des séances de travail comme dans l'événement des grandes déclarations, il importe de faire la place à chacun de ces points de vue : il n'y a pas une Raison, majeure, évidente et dictatoriale, mais de multiples rationalités, partielles, qu'il s'agit de drainer et concilier. La démarche proposée ici est une hygiène collective qui peut nous faire gagner beaucoup de temps et économiser beaucoup d'énergie. Au lieu de laisser s'étripier le chef de projet et le promoteur d'une idée, celui-ci gardant les yeux fixés sur l'avenir, celui-là focalisé sur les délais et budgets, disons-leur simplement que leurs discours portent sur des objets différents : le plan et la cible. Ne frustrons pas l'éclairé en lui disant qu'il n'est pas réaliste : son apport peut transformer l'avenir de l'entreprise, à long terme. N'humilions pas le berger en lui reprochant de penser exclusivement à l'étape du soir : son souci préserve le présent et, ainsi, rend l'avenir possible.

Être réaliste n'implique pas de se comporter en *ideal-killer* ! Réciproquement, chercher à améliorer la réalité, ce n'est pas la nier.

La méthodologie qui répondra aux besoins des nouveaux projets et aux aspirations de la société, devra intégrer cette dimension psychologique et, plus précisément, ménager sa place à chacune des postures essentielles de l'homme face au temps : de l'idéalisme au réalisme.

Dominique Vauquier
UNILOG – DSO
dominique.vauquier@unilog.fr

À César : Je dois à Hervé Pont (Azur-GMF) l'idée centrale de cet article et les expressions associées aux quatre objets (cible, objectif, trajectoire et moyen). Remerciement et amitié.



Le CathédralOscope de Dol-de-Bretagne

À l'approche des vacances estivales, il était urgent de parler du « CathédralOscope ». Ce n'est pas, comme le lecteur pourrait le penser, le titre d'un nouvel ouvrage d'ADELI. Quelques explications s'imposent, explications qui nous permettront peut-être de comparer, toutes proportions gardées, les bâtisseurs de cathédrales de l'ancien temps et les bâtisseurs de... systèmes informatiques des temps modernes !

Le musée de Dol-de-Bretagne

Il y a, sur la grand'place de Dol-de-Bretagne, une très belle cathédrale, comme beaucoup d'autres en France, et juste en face, un « CathédralOscope ¹ ». Le concept de ce musée semble calqué sur celui de nos ouvrages en « scope » diffusés par ADELI. Je me suis d'ailleurs laissé dire que des pourparlers étaient en cours entre le Bureau d'ADELI et les ministères de la culture et du tourisme pour le versement de redevances !

La genèse des cathédrales

La visite de ce musée apporte réponse à « tout ce que vous avez toujours voulu savoir sur les cathédrales, sans jamais avoir osé le demander ».

Leurs principes, leurs points communs, leurs différences, leurs procédés de fabrication, leur symbolique, leur décorum, leur sens, leur destination : tout cela est présenté, aux adultes et aux enfants, de fort belle façon, ludique et intéressante, notamment au moyen de remarquables maquettes, explications, reproductions et animations.

Parlons simplement de ce qui m'a le plus particulièrement frappé. On peut visiter de très belles cathédrales, lire les commentaires explicatifs des guides, apprécier l'architecture extérieure, puis intérieure, admirer les merveilles des vitraux, des sculptures et des peintures, éprouver le calme et la force de l'ensemble, mais il est beaucoup plus rare, avouez-le, de voir une cathédrale en construction. Dans ce musée, on peut en voir beaucoup, et s'imaginer à la place des bâtisseurs de l'époque.

Une construction au gré des événements

Je m'étais souvent demandé sur quels principes et quelles bases étaient construites ces cathédrales, imaginant, par exemple, à tort, qu'une cathédrale se construisait en une seule fois selon un plan clair et défini une fois pour toute. J'ai donc été surpris de découvrir que, le plus souvent, tel n'était pas le cas, ce qui explique les asymétries bizarres, les différences de styles, etc.

Finalement, c'est un peu comme un système informatique : en général, un très grand projet de départ, souvent sur l'emplacement d'un ancien édifice plus petit, un projet de construction grandiose qui peut durer plusieurs années ; puis plus tard, beaucoup de modifications, de destructions accidentelles ou d'agrandissements nécessaires, de changements et évolutions. Parfois, on n'a plus le budget, alors on réduit les ambitions. Parfois, les ressources s'accroissent et on enrichit les extensions. Je ne sais pas si cela vous rappelle des souvenirs...

¹ Visiter le magnifique site www.dol-de-bretagne.com

La résistance aux épreuves

Je m'étais aussi souvent demandé comment ces cathédrales parvenaient à maintenir leur fier équilibre, sans subir de fatales dégradations, face aux assauts des éléments et des conflits.

On comprend que les robustes églises romanes puissent résister aux épreuves du temps. En revanche, les cathédrales gothiques semblent plus vulnérables. Pour permettre l'éclairage de la structure centrale, on a percé de nombreuses ouvertures en fragilisant les murs centraux. Pour souligner leur légèreté, on les a habillées des fines dentelles de pierre.

Mais, en parfait contraste avec cette finesse, une structure externe d'arcs-boutants, lourde et massive, assure le maintien et la solidité de l'édifice.

Un peu comme une gracieuse princesse entourée d'une grande troupe armée, ou comme le PDG d'une entreprise, entouré de ses directeurs, eux-même entourés de leurs collaborateurs. Ou encore comme un serveur central entouré et soulagé par une multitude de serveurs secondaires, stations, périphériques, ou autres datawarehouses...

Impossible de regarder Notre-Dame de Paris de la même manière après cela : un précieux et fragile réceptacle ajouré et immense, entouré d'énormes et massifs blocs de pierre. Non un bloc uniforme, mais le Yin et le Yang. Et le plus surprenant est que ces contreforts massifs paraissent relever de la plus harmonieuse des élégances !

Du roman au gothique ...

Le seul tout petit reproche que l'on pourrait faire, actuellement, à certaines explications du CathédralOscope, c'est de présenter à, certains endroits, comme beaucoup de livres d'histoire, l'église gothique comme plus belle et plus élevée spirituellement que l'église romane, sous le prétexte qu'il y pénètre une plus grande quantité de lumière.

Il me semble qu'un tel point de vue serait une erreur. Il suffit de pénétrer dans maintes églises romanes et d'y chercher à méditer pour constater qu'il s'en dégage une paix profonde, aussi forte voire parfois plus que dans certains édifices gothiques moins réussis...

Et chacun d'entre nous sait que les plus beaux, les plus performants et les plus impressionnants systèmes informatiques que l'on découvre ne sont pas toujours ceux fondés sur le tout dernier cri de la modernité et de la technologie, loin s'en faut...

Bref, si vous souhaitez, pendant vos loisirs, mieux comprendre l'informatique, vous pouvez découvrir, églises et cathédrales et, surtout, visiter aussi le CathédralOscope !

*Pierre Fischof,
membre de l'ADELI,
pierre.fischof@libertysurf.fr.*