



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.

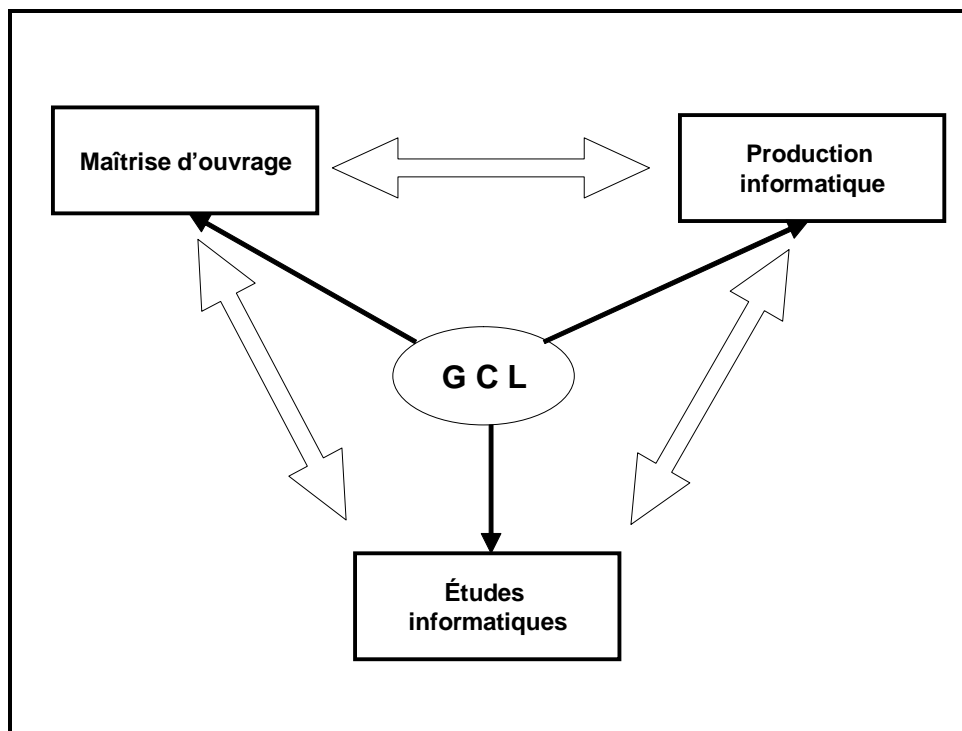
À minima, cette gestion des composants peut être rudimentaire, implicite, empirique : sauvegarde sommaire par les développeurs de leurs sources, à une date donnée, sur un support de stockage affecté à cet effet : disque, bande, voire disquette... Sauvegarde des exécutables à une date donnée par la production. 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).

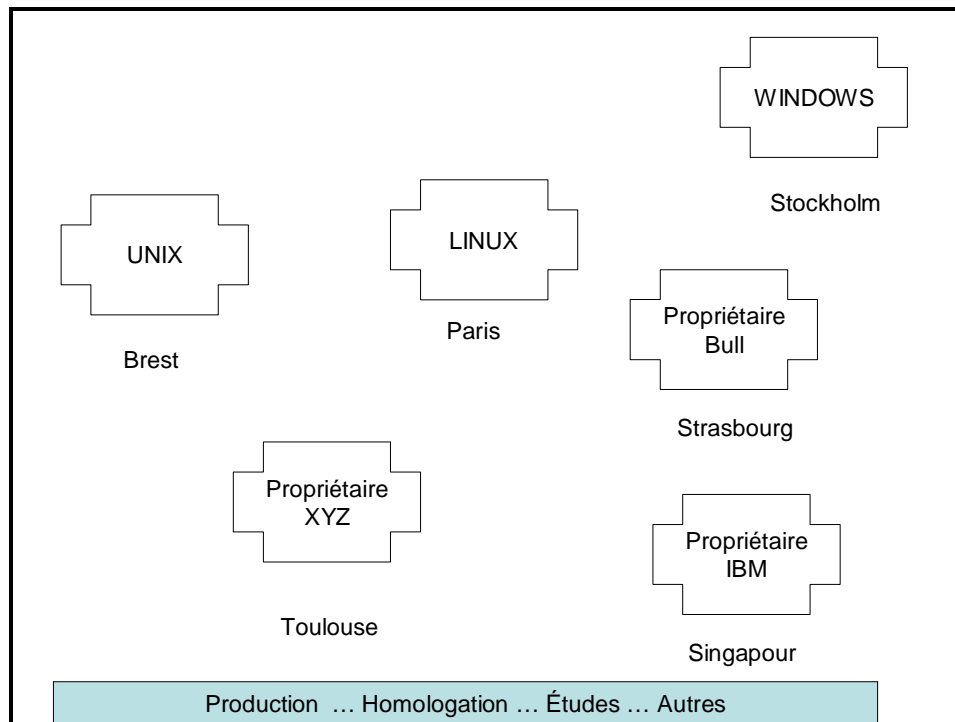
Chacun peut représenter, de plus, des centres de profits, des entités économiques, voire des entités juridiques indépendantes, voire concurrentes, dotées chacune, dans le cadre général, de finalités particulières, la GCL prend donc dès lors un rôle très contractuel.



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.

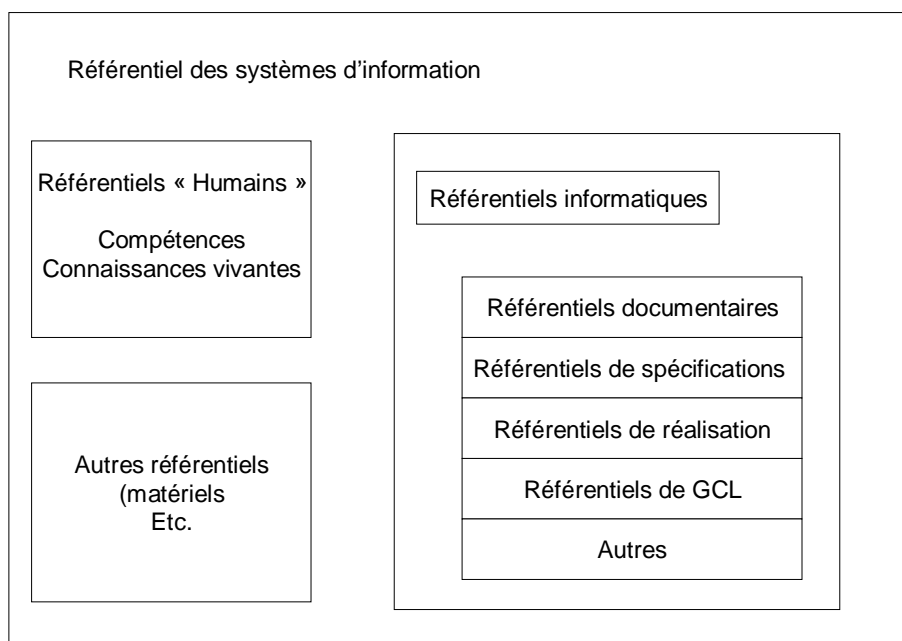
Notons d'abord que nous ne traitons pas ici des logiciels strictement industriels, qui sont soumis à des enjeux, des exigences, contraintes et compétences débordant de notre cadre d'informatique de gestion.

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

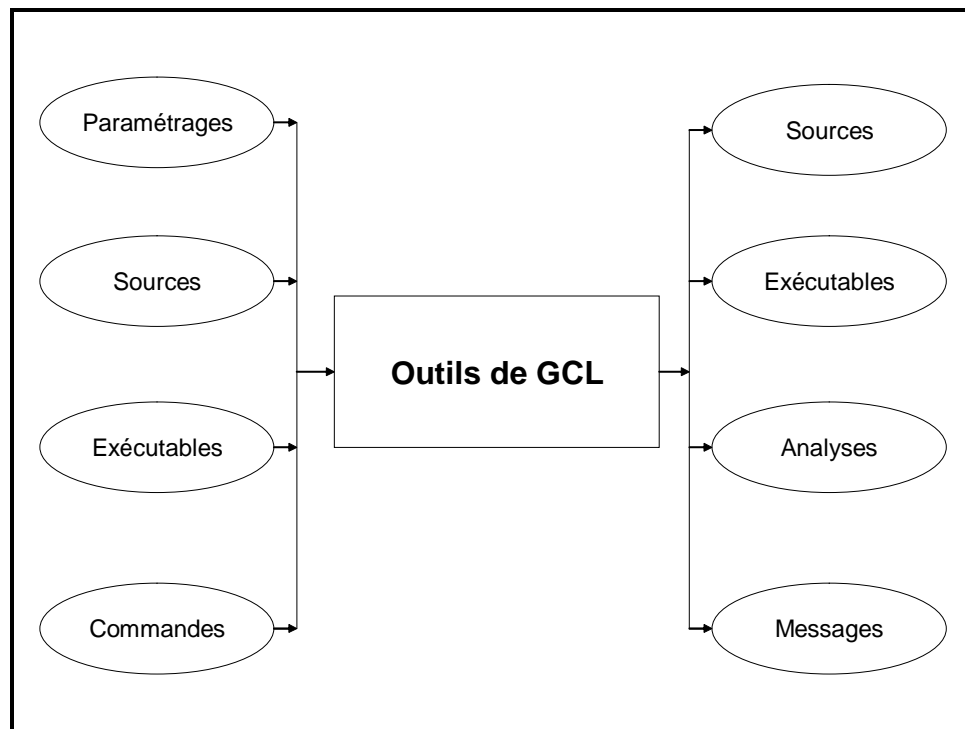
Quelles seront les sorties attendues de la GCL, pour reprendre la démarche claire, chère à Warnier ?

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.

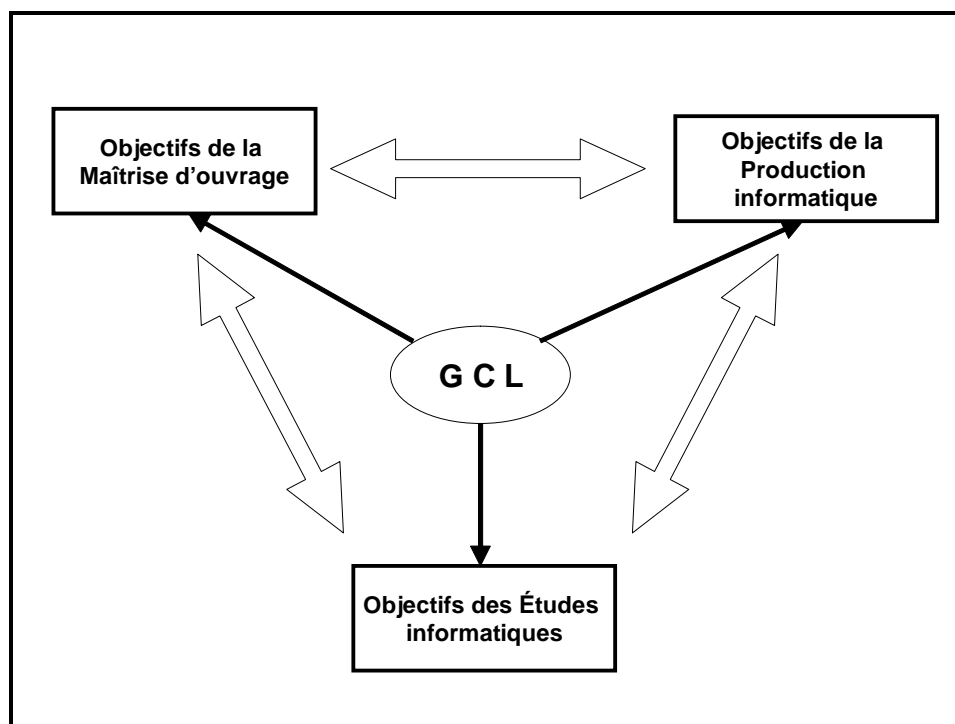
La première vue, plus simplificatrice, peut être néanmoins grossièrement acceptable, auquel cas il faudrait alors considérer tous les compilateurs (au sens le plus large) et procédures de compilations nécessaires comme constituant des entrées à fournir à l'outil de GCL faisant pleinement partie du paramétrage. 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

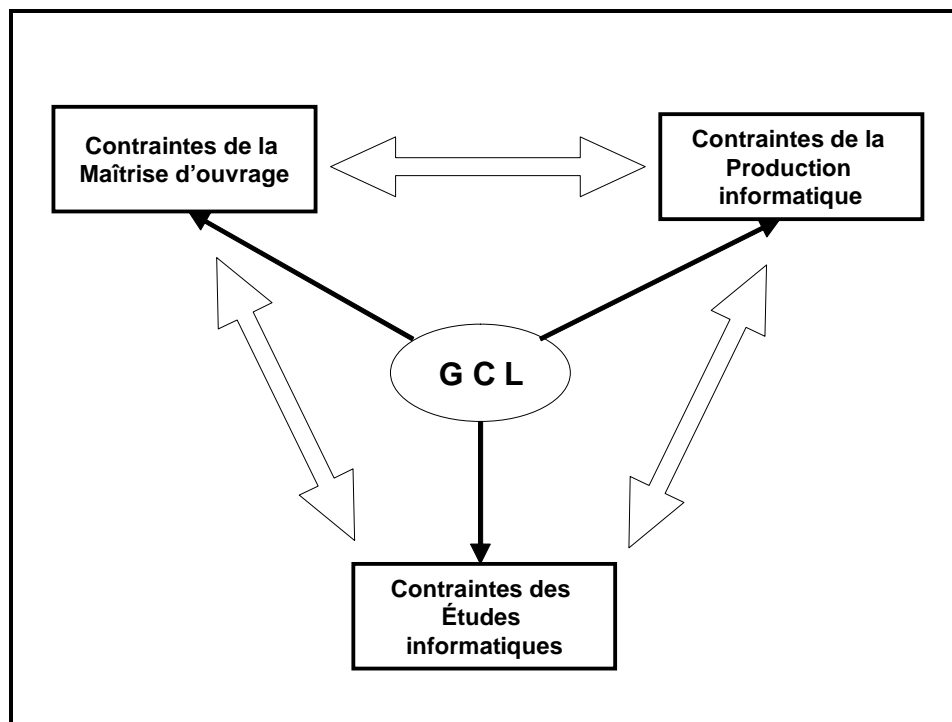
Périmètres : Complémentairement et symétriquement à ces objectifs, en examinant au départ le paragraphe concernant le « de quoi ne s'agit-il pas ? », nous avons ainsi implicitement défini les « non objectifs » de la GCL, et donc le périmètre de ce que nous pouvons et ne pouvons en attendre.

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