



Square des Utilisateurs

# GCL et génie logiciel : l'effort vers la qualité

**Retour d'expérience d'un DSI : entretien avec Guy Lapassat**

*propos recueillis par Mireille Boris,  
Journaliste indépendante en informatique*

*mireilleboris@noos.fr*

*et par Pierre Fischhof,*

*Chef de projets en évolutions et migrations de systèmes d'information*

*pierre.fischhof@libertysurf.fr*

*Membres de la commission ADELI «Gestion de configuration logicielle»*

*Mireille Boris et Pierre Fischhof poursuivent leur série d'interviews de professionnels et managers sur le thème de la GCL.*

*Ils ont cette fois interviewé Guy Lapassat, DSI du CCF<sup>2</sup> de 1990 à 2000, et ensuite de la Générale des Eaux.*

*« La GCL est un sujet gigantesque » affirme Guy Lapassat en préambule. Elle fut son souci majeur dans ses différentes fonctions, et spécialement au CCF où son activité fut considérée comme exemplaire.*

## La GCL : un devoir de maîtrise du directeur informatique

### **Q : La GCL n'était-elle pas une préoccupation majeure dans vos fonctions ?**

G.L. : On ne peut imaginer que ce ne soit pas le souci d'un directeur informatique. De quoi se compose son système, comment il évolue, où sont les composants du SI, c'est vraiment fondamental. Mais cela a beau être fondamental, je ne suis pas sûr que tout le monde s'en préoccupe. Il y a des sociétés où je suis passé, en tant que consultant ou directeur informatique, où on ne gérait que la dernière version !

L'historique ou simplement la gestion de la date n'a pas été le premier souci du monde. Preuve, l'an 2000 qui a coûté des milliards de dollars tout simplement parce que le traitement de la date avait été négligé ... Et, si on faisait une petite enquête, on s'apercevrait qu'il n'y a pas beaucoup d'entreprises qui gèrent le problème des versions. Pour moi, c'est un souci fondamental, et je crois qu'il y a beaucoup de progrès à faire dans les entreprises. Malgré la préoccupation que j'en avais, je n'affirmerai pas que j'aie résolu le problème, loin de là. Je n'étais, partout où je suis passé, qu'à une étape de la gestion de mes problèmes. Je place la GCL dans un environnement beaucoup plus vaste qui est celui de la qualité du logiciel. Beaucoup de gens affirment qu'il est normal qu'un progiciel contienne des bugs<sup>3</sup> ! Je trouve cela scandaleux ! C'est comme si on vous vendait une Renault ou une Citroën en vous disant : « écoutez, si vous faites Paris-Marseille, il y a une forte probabilité que vous n'y arriviez pas avant d'être passé dans un garage ». C'est ainsi que nous vivons en informatique avec le constructeur le plus puissant de la planète. Un jour ou l'autre, on arrivera à des solutions de meilleure qualité. La GCL, c'est un pas vers cette atteinte de la qualité, tout simplement.

---

<sup>1</sup> Gestion de Configuration Logicielle

<sup>2</sup> Crédit Commercial de France

<sup>3</sup> Note de la rédaction pour les puristes : bogue, hélas peu utilisé dans le milieu professionnel informatique français.

## Les bugs : réalité inacceptable, traçabilité obligatoire

### **Q : La GCL est donc liée à la démarche Qualité de l'entreprise ?**

G.L. : Inévitablement. Une entreprise doit savoir où se trouve son patrimoine, combien elle a dépensé, Les services généraux doivent connaître les immeubles et les mobiliers qui composent l'entreprise (je sais que beaucoup d'entreprises ne le savent pas et ne sont pas à jour !). Il est indispensable de savoir de quoi se compose son stock informatique. C'est nécessaire pour arriver à gérer l'existant. Quant aux composants, si vous n'êtes pas capable de gérer les versions, le niveau de service va en souffrir. Sur 100% d'incidents, 50 sont dus à la production, 50 aux études, et les incidents d'exploitation qui sont dus aux études sont quasi exclusivement de la mise en production. Si une entreprise n'est pas capable de mettre en exploitation un produit sans erreur, elle n'est pas arrivée au niveau de qualité suffisant, c'est tout simplement inacceptable. Non, ce n'est pas normal.

Si on veut améliorer la qualité, on doit savoir où est son stock. Je ne parle pas de ce qu'on devrait faire et qu'on ne fait pas parce que les outils ne sont pas sur le marché : garantir que les produits qu'on développe sont de bonne qualité. Dans la gestion du logiciel, il n'y a pas que la cartographie ou la gestion des versions, il y a la gestion des tests, le contrôle formel des programmes. La plupart des sociétés n'en font pas. Combien de sociétés font-elles du contrôle de non régression ?

Le CCF a été extrêmement sérieux. Mais ce qui est à faire est gigantesque. Pour l'instant quand on parle de configuration logicielle, on traite ce qu'il est possible de faire aujourd'hui, c'est-à-dire documenter, gérer les composants.

On peut noter aussi que l'on oublie trop souvent la loi de traçabilité comptable qui contraint théoriquement l'entreprise à pouvoir justifier et réexécuter à volonté tous les traitements impactant la comptabilité. Cette loi implique donc légalement une historisation complète des composants informatiques et l'archivage de toutes les données permettant ainsi de réexploiter à volonté. Mais dans la pratique, la rapidité d'évolution rend cela presque impossible ; aucune entreprise ne peut le faire entièrement. Cela pour la raison que, par exemple, quand les hardwares<sup>4</sup> évoluent, il est presque impossible économiquement de garder leurs anciennes versions; et pour les softwares<sup>5</sup> d'exploitation, c'est presque aussi insoluble ; quant à l'historisation de toutes les données, c'est très complexe à bien réaliser. La traçabilité est donc bien un devoir de l'entreprise mais l'imaginer intégrale serait une utopie.

## Différencier les composants du SI par une logique multicouche

### **Q : Préférez-vous parler de gestion de configuration tout court plutôt que de GCL ?**

G.L. : Je n'aime pas me battre sur le vocabulaire. Bien sûr, il faut faire la configuration de l'ensemble de ses équipements. Mais il y a tout de même une limite. On est obligé de spécialiser. Il faut des personnes qui gèrent les ordinateurs, des personnes qui gèrent les progiciels de production, et d'autres les logiciels applicatifs. Cette spécialisation est inévitable et je la recommande. On peut imaginer qu'on ne gère pas les équipements comme on gère les logiciels.

De même, c'est une erreur grave de mélanger le traitement et les données. On devrait être en mesure de séparer les deux. C'est une faiblesse critique de HTML, par exemple. Un système qui n'est pas modulaire et qui ne sépare pas correctement ses composants pose problème - quand vous devez changer le traitement des données, vous devez modifier votre dictionnaire de données, ou vice versa...

Dans un objet, il y a des données et des traitements, donc on doit gérer les deux, mais pas ensemble. Données et traitements n'évoluent pas de la même façon. Les traitements portent sur des données et tout ceci communique au travers d'API<sup>6</sup>. Est-ce qu'on parle des données qui sont dans les API ou des données sur lesquelles travaillent les traitements ? On a besoin d'outils pour décrire les données que

---

<sup>4</sup> En français : matériels

<sup>5</sup> En français : logiciels

<sup>6</sup> Application Programming Interface : Interface de programmation. Les API consistent en des appels de fonction, accompagnés des paramètres à communiquer à l'extérieur de l'applicatif.

l'on traite, d'outils pour décrire les API, on a besoin d'outils pour décrire les traitements, et on ne demande pas tout à fait la même chose à ces outils. Un outil de gestion de configuration logicielle va devoir gérer les dictionnaires de données. Au CCF, c'étaient Data Manager et MC Designer. Pour la gestion des traitements, il y a plusieurs produits, qui sont d'une part la cartographie proprement dite, et d'autre part tout simplement la gestion des bibliothèques, des produits qui sont capables de stocker des sources, des objets, et de dire qui appelle quoi. Il y a des outils de cartographie ou des outils de rétro-ingénierie qui vous permettent de reconstituer le source, et de refaire les organigrammes de programmes écrits il y a 5 ou 10 ans. Les mêmes logiciels ne peuvent pas tout traiter.

Une vision du système d'information selon une logique multicouche (couche métier, couche logique, couche physique ...) alliée à une décomposition en fonctions (fonctions de présentation, fonctions de stockage, fonctions de calculs et transformation, etc.) va nous aider à discerner les types de composants du système et ainsi la façon d'optimiser la gestion de chacun d'eux.

Configuration des sources, des objets, des équipements sont trois choses distinctes. Il y a les données et les traitements au niveau des sources, et au niveau des objets, il y a ceux qu'on a développés soi-même et il y a les progiciels. Il faut croiser tout ceci avec d'autres préoccupations qui sont les phases de développement. Il faut les étudier alors qu'on est en phase de développement ou qu'on est en phase de maintenance. Au niveau de la conception, il y a la partie purement conceptuelle et la partie physique. Au niveau de la réalisation, il y a la partie développement, et la partie test. À chacune de ces étapes, ce ne sont pas tout à fait les mêmes outils qu'on emploie, ce ne sont pas tout à fait les mêmes problèmes. Pour chacun de ces problèmes, on va avoir à chercher des outils qui répondent à ces besoins là. On peut imaginer qu'idéalement un éditeur fournisse la réponse à toutes ces questions, mais il sera probablement très bon sur un de ces domaines, plus faible sur d'autres.

## L'importance d'une réelle politique de tests

### **Q : Pour s'assurer de la qualité de ses produits, la réponse n'est-elle pas dans les tests ?**

G.L. : Je considère que la gestion des tests est un sujet fondamental. Comment contrôler formellement que ce que j'ai écrit est correct en dehors des tests ? Pour Ariane 5<sup>7</sup>, ils en ont fait certainement une partie mais pas tout...

Quand je suis arrivé au CCF, en 1990, j'ai dit à mes équipes : « vous faites du très bon travail, mais au feeling, je dirai qu'il y a un domaine dans lequel nous ne sommes pas bons, ce sont les tests ». Ce qui me choquait c'est qu'on faisait des tests sur des fichiers réels ce qui est révélateur du fait que les jeux d'essai ne sont pas bien faits. Mes équipes, pour des raisons diverses, ont mis beaucoup de temps à accepter. On a commencé à s'intéresser aux tests dans la deuxième partie des années 90. On a acheté une grande partie des produits de Compuware. Quelque temps après, j'ai demandé au président de Compuware de me présenter des sociétés qui soient allées jusqu'au bout du travail de tests et qui puissent nous expliquer ce que nous pouvions améliorer. Il m'a répondu : « vous vous trompez, avec ce que vous avez fait, vous êtes en avance ! ». Je considérais qu'on avait à peine récupéré 20 ans de retard. On était bons, il fallait être meilleurs encore. C'est un énorme chantier, partout dans toutes les entreprises. J'aimerais savoir combien d'entreprises ont des outils qui vont jusqu'au bout de la logique et combien d'entreprises font des tests de non régression.

- Cela coûte très cher à mettre en place.
- Cela coûte très cher à maintenir.
- Ce n'est pas bien vu des DG et des utilisateurs parce que cela apparaît comme une dépense.

La qualité, ce sont des dépenses en amont. Qui veut dépenser en amont ? Personne ... Nous vivons dans un monde où on parle beaucoup de qualité mais où on en fait moyennement.

---

<sup>7</sup> Voir dans La LETTRE n°25 l'article d'Alain Coulon « Ariane 5 » qui analysait les causes de l'échec du premier tir.

## Une mauvaise période pour la qualité

### **Q : La période actuelle serait plutôt mauvaise pour la GCL ?**

G.L. : Elle n'est pas mauvaise, elle est catastrophique. Je connais le cas d'un DI qui est aux ordres et sur ordre a arrêté tous les projets d'infrastructure à terme et de méthode.

C'est inutile, ça ne sert à rien, ce n'est pas visible. Payer pour une équipe interne en train de gérer des outils de version ?... On attend que le système soit planté, quand il est planté on corrige. La tendance générale va aujourd'hui en ce sens. À quel niveau était la Bourse en mars 2000, à quel niveau aujourd'hui ? On est dans une période critique, on s'occupe médiocrement de la qualité, on s'occupe de survie. N'est-ce pas se condamner à plus long terme ? La période n'est pas propice aux investissements qui visent à de l'intangible, comme la qualité.

### **Q : En revanche, au cours de votre passage au CCF, vous avez maîtrisé la GCL ?**

G.L. : Ce n'est pas à moi de juger si ce fut un début de maîtrise ou une grande avancée. Au CCF, les équipes informatiques de l'époque ont fait un travail énorme. Pendant les dix ans que j'ai passés au CCF, on a mis en place des gestions de versions, comme Endevor et cc-Harvest, un outil de rétro-ingénierie qui s'appelle Wincap, qui a du subir beaucoup d'évolutions. L'éditeur a dû développer et rajouter des analyseurs à la demande d'un de nos ingénieurs. On a mis en place des outils de cartographie, de tests, et les outils de modélisation UML et RUP de Rational Rose. Voici les outils que l'on a mis en place, avec, autour, toute une série de processus de gestion comme vous pouvez imaginer.

## GCL et génie logiciel : quels retours sur investissements ?

### **Q : Avez-vous fait des calculs de ROI<sup>8</sup> ?**

G.L. : On ne peut pas faire de calcul de ROI à proprement parler.

Mais on peut faire un calcul. Une banque doit pratiquement répondre à la seconde près à un incident. On suivait les incidents de très près. À la Générale des Eaux, l'essentiel de la production était consacré à traiter les incidents. À chaque fois que vous réduisez le nombre d'incidents, vous réduisez la taille de votre équipe. Si réduire le nombre d'incidents de moitié vous permet de réduire de 30% la taille de votre équipe de production, il y a un vrai retour sur investissement.

Un budget informatique de banque c'est 10% du CA. Mettre 2 M d'euros d'investissement pour faire des tests est rentable. La cartographie ne coûte presque rien du tout et il n'est pas très utile de faire des études de rentabilité.

J'ai mis en place un système de suivi du nombre et du taux d'incidents. Chaque fois qu'un état sort en retard, c'est un incident. Le taux, c'est le nombre d'incidents divisé par le nombre de programmes qui tournent. Il y a chaque année 20% de programmes en plus. En 1990, quand je suis arrivé, il y avait 10 000 programmes qui tournaient, quand je suis parti, il y en avait plus de 40 000, et les incidents se réduisaient de 17% par an en quantité et de 33% en taux. C'est-à-dire qu'en 96, il y avait un incident sur 40 M de lignes de code traitées et que l'objectif était d'arriver en moins de 7 ans à un incident sur 400 M. Étant donné l'argent qu'on investissait là-dedans, c'était rentable !

Tous ces outils sont absolument fondamentaux. S'ils ne sont pas systématiquement mis en œuvre partout, c'est que le monde informatique n'a pas compris ce qu'est la qualité.

---

<sup>8</sup> Return On Invest, Retour sur investissement.

## Gérer les contenus, les torchons et les serviettes

### **Q : Les outils de GCL peuvent-ils gérer le contenu ?**

G.L. : Quand on parle de contenu, il s'agit plus d'arrivées d'informations non structurées que d'informations structurées, que les outils du marché savent traiter.

Si l'entreprise considère que toute l'information, tous les messages qui s'échangent sont des contenus qu'il faut gérer, oui, pour l'instant, on le gère très mal. On en est aux balbutiements d'une véritable gestion de contenu.

Ce qui est nouveau, c'est que c'est non structuré et que les volumes sont gigantesques. Il faudra le faire. Les outils faits pour gérer du code et des données structurées, auront du mal à gérer de l'image, par exemple. On est en train de faire du « web sémantique », d'essayer de retrouver ce qu'il y a derrière un texte qui a été stocké, restructuré, et peut être classé, Ce ne seront manifestement pas les mêmes outils. Wincap déjà n'analysait pas tous les langages qui pouvaient être utilisés dans une entreprise comme le CCF. De la même façon, les futurs outils vont devoir gérer un autre type d'information. Pour l'instant, ce n'est pas un sujet critique pour l'entreprise - on a de la marge.

## Les hommes et leurs outils

### **Q : Facteurs techniques, facteurs humains, lesquels doivent dominer ?**

G.L. : C'est l'un et l'autre...

En cartographie, les informaticiens découvrent que la vision de la production et la vision des études ne sont pas exactement les mêmes, parce que la production rajoute des tris, des points de contrôle, refait des blocs, en fonction de ses contraintes d'exploitation, pour pouvoir faire des reprises, des sauvegardes, pouvoir séquencer ses traitements, parce qu'une application attend l'arrivée d'une autre, parce qu'il y a des équilibrages de charge. Elle a une cartographie pratiquement imposée qui n'est pas la cartographie conceptuelle et presque pas de principe des études.

Études et production n'ont finalement pas le même langage. Donc, la cartographie les éclaire. Ceux qui sont encore plus éclairés, sont les utilisateurs ! La cartographie est un outil extrêmement bien reçu par les utilisateurs, mais il faut la mettre à jour. Il y a un travail considérable à faire pour convaincre les différents responsables de mettre à jour leur cartographie. Sur le principe, tout le monde est d'accord. Mais une volonté de mise à jour d'informations est indispensable.

Pour les tests, la première fois que vous mettez en place des tests, il y a des réactions extrêmement négatives. C'est plus lourd, c'est inutile... On s'aperçoit que ce n'était pas le cas. Assez rapidement, dans les entreprises qui marchent bien, les utilisateurs réalisent très vite qu'il y a une amélioration de la qualité, de la compréhension, une rapidité de résolution qui est extraordinaire.

Mon expérience est simple : au bout de quelques mois de mise en place des tests, ce sont les utilisateurs qui incitaient les équipes d'informaticiens à les utiliser quand ces équipes ne voulaient pas le faire ou n'allaient pas assez loin. Facteurs techniques, facteurs humains, il faut les deux à la fois. Il y a un travail à faire sur le terrain pour lancer les opérations, pour convaincre – la solution est simple, on commence par les plus motivés, et si l'opération est bonne les autres suivent. Mais il est évident que si vous n'avez pas l'adhésion des équipes, des utilisateurs, des informaticiens, vous n'arriverez jamais à mettre en place des outils techniques.

Mon expérience est plutôt positive : quand vous mettez tout cela en place, il y a des gains de qualité et tout le monde suit finalement.

## L'atelier de génie logiciel intégré : mythe ou réalité ?

**Q : Tout ce dont nous avons parlé jusqu'ici fait bien partie du génie logiciel ?**

G.L. : Tout à fait.

Ce qui me préoccupe c'est que le concept d'AGL qui était un bon concept ait un peu disparu. Les AGL, comme les schémas directeurs, ça n'existe plus. Le problème des AGL est qu'on vendait des objets complètement intégrés qui étaient en fait des générateurs avec dictionnaires de données, et outils de cartographie. Ils débouchaient sur de la génération avec des structures rigides et imposées. Évidemment, ça ne marchait pas. Je discutais avec une société qui vendait un AGL et je leur disais : « votre dictionnaire de données, je le trouve bien, est-ce que je peux vous l'acheter ? » – « Non. C'est tout ou rien, nous vendons l'AGL. » m'a répondu le responsable. Et bien, ce fut rien.

Nous avons mis en place au CCF Data Manager plus MC Designer, UML, RUP, Compuware, la cartographie Wincap, cc-Harvest, Endeavor. Il n'y a pas de produit intégré meilleur que cet assemblage...

Les AGL sont morts de vouloir être intégrés au lieu de vouloir intégrer des fonctions. La nouvelle génération d'AGL ne fait pas d'Upper Case<sup>9</sup> comme il faudrait. Autre défaut des AGL en général, on voit sortir des dizaines de langages différents, ce qui est une hérésie. Dans un langage, l'essentiel est d'intégrer des fonctions techniques. On nous embrouille avec les langages. Par exemple, en ce moment, des chapelles se battent pour dire que PHP est meilleur que Java, Microsoft sort sa solution propriétaire au lieu de s'orienter vers un langage commun et on régresse.

Le génie logiciel est encore une jungle et nous éloigne de l'objectif de l'entreprise.

Merci, Guy Lapassat, d'avoir répondu si ouvertement et franchement à nos questions.

---

<sup>9</sup> *Atelier d'aide à la conception*

## **Biographie de Guy Lapassat :**

Guy Lapassat est actuellement consultant.

Il a partagé sa carrière entre l'entreprise :

- en tant que Directeur informatique (Générale des Eaux, Crédit Commercial de France pendant 10 ans, Paribas Capital Market à Londres) ;
- Directeur Stratégie, Contrôle de Gestion, Informatique (Olida-Caby) ;

et le service :

- 13 ans chez Arthur Andersen & Cie (industrie, banque, presse) ;
- puis directeur d'agence Cap Sogeti (progiciels) ;
- et directeur de division Sema Group (ECS, Banque).

Il est membre du Conseil d'Administration de l'ASTI. En tant que membre du Conseil d'Administration du CIGREF, il a animé plusieurs groupes de travail (progiciels, workflow, marketing interne des DSI, Club Nouvelles Technologies).

## **Publications**

- « **Architecture technique informatique** »(Hermès 1997)  
explique comment structurer et écrire un programme en informatique de gestion en architecture multi-couches. Ce livre propose des principes, des règles et des solutions à tous ceux qui développent des programmes traditionnels comme à ceux qui utilisent des méthodes et des outils récents (objet, client-serveur, workflow, Internet, etc.) ou qui ont besoin simultanément de toutes ces diverses techniques, en environnements hétérogènes. Il montre comment il est possible à chaque entreprise de mettre en œuvre les règles et les solutions proposées en fonction des moyens dont elle dispose.
- « **Urbanisme informatique et architectures applicatives** » (Hermès 2003)  
définit le rôle et les méthodes de l'urbaniste et de l'architecte des SI, ainsi que les règles qu'ils doivent appliquer pour définir la structure des systèmes d'information d'un univers classique, Objet ou Web, en particulier pour les architectures multi-couches.  
Ce manuel comprend 3 parties : L'urbanisme fonctionnel, son environnement et son domaine d'action, Principes et règles de l'urbanisme fonctionnel, Typologie des systèmes d'information.
- « **Architecture fonctionnelle des logiciels** » (Hermès 2003)  
est un manuel de référence pour tous ceux qui doivent concevoir une architecture fonctionnelle de logiciel ou de progiciel, qu'il s'agisse de faire évoluer ou refondre un système d'information, de faire converger plusieurs systèmes ou d'assurer l'interopérabilité de progiciels, de définir les objets et les règles réutilisables, d'imposer des normes, qu'elles soient aussi simples que les dates ou les montants, ou aussi complexes que celles qui régissent les référentiels tiers ou annuaires, ou d'intégrer le monde Internet.