

Dans l'industrie, la fonction représente l'élément de base pour l'analyse de la valeur.

Il en est de même pour le S.I. : la fonction est certainement l'élément le plus apte à établir le rapport coût/bénéfice, et par conséquent, à prendre les meilleures décisions de gouvernance du patrimoine informatique.

Bien entendu, cela suppose d'avoir établi un référentiel de fonctions, de manière à collecter les coûts de chacune, et d'identifier celles qui ont le plus de valeur. Contrairement à ce que l'on pourrait croire, cette démarche est beaucoup plus facile qu'il n'y paraît. Un bon architecte fonctionnel suffit !

Première étape : qu'est-ce vraiment qu'une fonction ? Nous avons tous le sentiment que le système informatique de notre entreprise, nous offre des centaines, voire des milliers de fonctionnalités. Il n'en est rien : l'analyse menée dans une banque de détail montre que le nombre de fonctions qui offrent une véritable valeur ajoutée aux clients, internes et externes, ne dépasse pas la trentaine...

Pléthore de fonctions : une idée reçue

Notre système d'information est beaucoup trop riche en fonctionnalités non rentables : c'est le reproche majeur qui ressort dans une enquête interne récemment menée dans un grand groupe bancaire.

Nous avons tous l'impression que nos systèmes d'information nous offrent des centaines, voire des milliers de fonctions. Une analyse détaillée montre qu'il n'en est rien : lorsqu'on se focalise sur les fonctions qui apportent une véritable valeur ajoutée, on arrive au plus à quelques dizaines de fonctions, sans que ni la taille, ni l'activité de l'entreprise n'aient d'influence marquante sur ce nombre.

D'où provient cette impression de multiplicité des fonctions ? Les causes en sont multiples, je me bornerai ici à analyser deux d'entre elles :

Confusion entre fonction et fonctionnalité

Selon mon Larousse en 15 volumes, une fonction est un ensemble d'opérations concourant au même résultat ; alors qu'une fonctionnalité est ce qui répond à une fonction déterminée.

Prenons l'exemple des phares de voiture. De nombreuses fonctionnalités y sont associées, dont la plus évidente : les allumer ou les éteindre. On peut y ajouter les dispositifs qui permettent de régler la hauteur du faisceau lumineux, de nettoyer les phares, d'alimenter automatiquement le feu de recul au moment où l'on enclenche la marche arrière, voire d'allumer les phares dès que le véhicule est plongé dans l'obscurité. Si toutes ces fonctionnalités peuvent posséder une certaine valeur – du moins pour une partie des acheteurs - il n'en reste pas moins que l'éclairage en lui-même n'offre que deux fonctions : voir, et être vu. Le reste est une question d'interface d'accès à ces deux fonctions.

Si on transpose la comparaison en informatique, les fonctionnalités sont les moyens d'accès aux fonctions, associées à des règles de gestion qui vont régir le comportement de la fonction (s'il fait noir, alors l'éclairage se déclenche). Dans plusieurs pays d'Europe, où il est obligatoire de conduire en feux de croisement, plusieurs constructeurs ont supprimé l'interrupteur. Les phares s'allument dès le démarrage : on économise un dispositif qui n'est d'aucune utilité et qui n'a donc aucune valeur.

Confusion entre manuel et automatisé

Cette confusion se rencontre très fréquemment. Pour l'illustrer, je prendrai l'exemple de la gestion de projet. Il existe de nombreux logiciels de gestion de projet sur le marché, nous en avons tous pratiqué. Certes, ces logiciels gèrent¹ des tâches, des ressources... mais ils ne se substituent jamais au chef de projet : c'est bien lui qui positionne les tâches, précise les contraintes d'enchaînement de celles-ci, et affecte des ressources sur chaque tâche. Tout bien considéré, la valeur ajoutée d'un logiciel de gestion de projet tient à 3 fonctions :

- le lissage : il optimise le positionnement des tâches, pour déterminer la date de fin optimale sans trop surcharger les ressources ;
- l'agrégation des coûts : à condition d'avoir affecté au préalable, un coût aux ressources, elle permet d'obtenir le coût cible du projet, puis de le comparer au coût estimé, au fur et à mesure de son avancement ;
- la restitution : certainement la plus utilisée en pratique... l'outil de gestion de projet nous sert surtout à communiquer le planning d'un projet et son avancement à tous les intervenants et aux parties prenantes !

¹ Il faudra un jour édicter une loi qui interdise l'emploi du verbe « gérer » dans tout document informatique de type cahier des charges, spécification, ou appel d'offres !

En pratique, c'est bien cette dernière fonction de présentation qui est la plus utilisée. Et pour cela, point n'est besoin de logiciel sophistiqué. En caricaturant à peine, on pourrait affirmer qu'Excel dispose de suffisamment de fonctions pour construire tout un SI. Que celui qui n'a jamais glissé dans un document un planning élaboré avec Excel me jette la première pierre !

Si on y ajoute Word et un outil de messagerie, on peut tout faire avec Excel dans une PME. Et une PME remplit exactement les mêmes fonctions qu'une grande entreprise : la taille ici ne fait rien à l'affaire.

D'ailleurs, la liste des fonctions d'Excel, celles qui s'affichent en cliquant sur l'icône, ne comprend que 200 éléments environ (mathématiques, textuelles...). On est bien loin des milliers !

Mais qu'est-ce donc qu'une fonction ?

C'est d'abord le résultat qui compte !

Nous avons un peu tendance à l'oublier : une fonction, c'est uniquement un ensemble d'opérations conjuguées pour retourner un résultat ! En mathématiques, une fonction $y = f(x)$ est définie par la relation des x et des y . Elle matérialise une relation entre une (ou plusieurs) valeurs en entrée, et une valeur en sortie, selon des règles précises. Par exemple, pour calculer le montant des mensualités pour rembourser une voiture, il me suffit de connaître le montant emprunté, la durée de l'emprunt, et le taux.

Application aux fonctions du S.I.

Pour les fonctions assurées par un SI, il est utile de préciser quelques règles et propriétés complémentaires :

- une fonction porte toujours sur un objet Métier de l'entreprise et un seul : il s'agit d'une règle d'urbanisme de base, garantie de découplage.
- une fonction devrait toujours pouvoir s'exprimer sous la forme d'un verbe d'action portant sur l'objet Métier : par exemple, calculer un score, contrôler une quantité en Stock, imputer un règlement sur un compte, ...
- une fonction a pour objet de faire passer cet objet Métier d'un état stable à un autre de son cycle de vie (à l'exception des fonctions de restitution, qui ne changent pas le statut de l'objet).

- une fonction s'exécute de manière ininterrompible : soit elle réussit à retourner le résultat, soit elle ne retourne rien (en cas de panne par exemple). Dans ce cas, il sera obligatoire de reprendre toute l'exécution au début¹.

En cas d'interruption, la fonction devra prévoir de restaurer à l'état initial les données qui auront été transformées : c'est la célèbre intégrité transactionnelle.

C'est le cas par exemple de la fonction d'autorisation bancaire, invoquée lors d'un retrait d'espèces, et qui a pour effet de réduire d'autant le solde disponible sur notre compte en banque. En cas d'échec de la transaction, il est fortement souhaitable de re-créditer la position de gestion !

En résumé

- Une fonction est le plus petit élément du système d'information ayant un but explicite, permettant d'assister un utilisateur dans son travail.
- Son objectif doit pouvoir être exprimé de manière indépendante de la manière dont elle est implémentée.
- Chaque fonction est responsable des informations qu'elle crée, supprime ou modifie. Bien entendu, elle aura souvent besoin de consulter des informations dont elle n'est pas responsable, et appartenant à d'autres objets Métier.
- Il en découle qu'une information est obligatoirement placée sous la responsabilité d'une seule et unique fonction, du moins à un instant donné de son cycle de vie.
- Enfin, 'last but not least', le comportement des fonctions est régi par des règles de gestion.

En pratique, pour chaque fonction à automatiser par un composant logiciel, le Métier ou la maîtrise d'ouvrage devrait être capable de spécifier à la maîtrise d'œuvre : les données dont elle a besoin en entrée ; les données que l'on attend en sortie ; et au moins une règle de gestion qui régit le comportement de la fonction.

Cette règle doit être « validable » par l'utilisateur, et par conséquent, vérifiable au cours d'une recette.

¹ Bien entendu la fonction peut également retourner un message d'erreur si les paramètres qui lui sont fournis en entrée ne respectent pas les règles de validité : exemple, Log de zéro.

Petite chaîne de valeur du S.I.

En simplifiant volontairement la liste des actions à valeur ajoutée que l'on peut appliquer aux informations, il est possible de proposer la chaîne de valeur, schématisée ci-dessous. Comme indiqué, un S.I. qui n'assumerait que deux fonctions : collecter et restituer, peut très bien se suffire à lui-même. Les deux autres fonctions, contrôler et transformer, ne sont en effet pas obligatoires, même si elles possèdent probablement une grande valeur ajoutée pour certains Métiers. La plupart du temps, elles s'appliquent à un sous-ensemble des données.

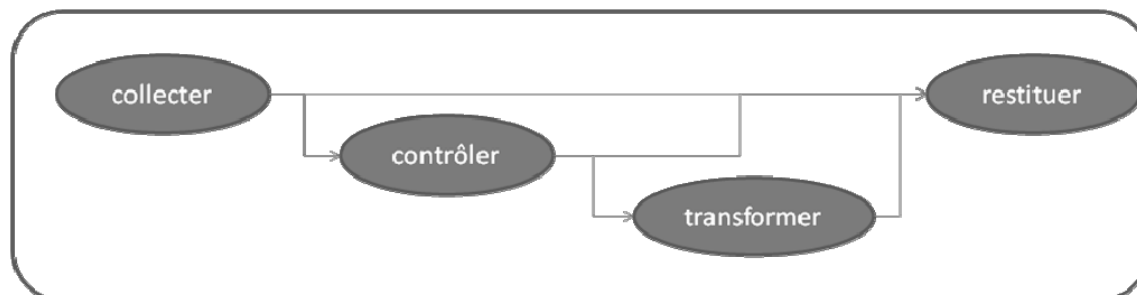


Fig 1. Chaîne de valeur du SI

Restituer : le fondement des S.I.

Une fois n'est pas coutume, commençons par la droite : la principale valeur ajoutée de nos S.I. tient à leur capacité de restitution de l'information. On estime en effet que de tout le patrimoine d'information manipulé par une personne dans l'exercice de ses missions courantes, moins de 3% a été créé par cette personne elle-même : tout le reste a été collecté ou créé par d'autres.

Par ailleurs, s'il n'était besoin de mettre à disposition des uns des informations fournies par d'autres, nous pourrions dès maintenant mettre au rebut tous nos coûteux serveurs, baies de disques et autres routeurs, et nous contenter d'un bon vieil ordinateur personnel !

Collecter : rien ne se perd si rien ne se crée

Sans collecte, point de restitution possible ! Cette fonction, qui peut paraître simple, a beaucoup plus de valeur qu'il n'y paraît lorsque l'on prend en compte son objectif fondamental : transformer de l'information en données manipulables dans un système informatique. En effet, dès que l'on commence à prendre en compte les problématiques de multi-langue, multi-alphabet, on se heurte rapidement à des subtilités techniques.

Il existe plus de 5000 langues sur notre planète...Le fait de savoir exprimer la sémantique de 24, XXIV, vingt-quatre, ou encore 二四 pour la Chine, en le codifiant sous la forme d'une suite de bits immédiatement exploitable par toute une famille de systèmes – par exemple 1111 0010 1111 0101 pour l'EBCDIC - possède une valeur indubitable.

Contrôler : l'information prend de la valeur

On a pu dire qu'une information n'a pas de valeur intrinsèque : celle-ci varie selon les utilisateurs et l'usage qu'ils sont en position d'en faire. Toutefois, on sait depuis longtemps qu'une information fiable a plus de valeur qu'une information approximative.

On distingue traditionnellement les contrôles de surface des contrôles sémantiques, lesquels sont exécutés par examen d'un référentiel ad hoc. Le contrôle d'adresse constitue un exemple classique de fonction à valeur ajoutée : si l'on veut contacter une personne, mieux vaut disposer de la bonne adresse, qu'elle soit postale, numéro de téléphone ou courrier électronique !

Plusieurs entreprises ont quantifié le coût du célèbre 'NPAI' (n'habite pas à l'adresse indiquée) et mesuré le ROI d'un investissement informatique pour le réduire. En ont résulté les systèmes experts utilisés dès les années 80 par les sociétés de VPC pour dédoublonner leurs contacts, ou le service Unigeo qui permet de vérifier l'existence d'une adresse postale.

Transformer : marginal, mais fondamental

Je regrouperai sous ce terme les transformations et les créations pures d'information : la frontière est ténue. Et surtout, la création ex-nihilo de données dans un système informatique ne se rencontre que très exceptionnellement¹. Sur des points très ciblés, un investissement minime peut dégager une grande valeur pour l'Entreprise et pour ses clients.

¹ à part générer un nombre aléatoire, il ne m'en vient pas à l'esprit, mais je fais appel à tous les Adéliens pour fournir des exemples !

Application à la Banque

Les principales fonctions du S.I. dans la Banque de Particuliers peuvent se réduire à une courte liste : dans un organisme de Crédit nous sommes arrivés à une soixantaine.

Sur ces soixante, la moitié sont génériques : on trouve les mêmes dans d'autres métiers. Il s'agit par exemple de collecter des données personnelles (client/prospect), de planifier une action (dans un agenda), de numériser un document, d'imprimer, ... Au final, une trentaine seulement s'avère spécifique au métier de la banque. Ces dernières peuvent se regrouper en 6 catégories :

- autoriser une opération (par exemple un retrait d'espèces dans un guichet automatique) ;
- tenir la position de gestion : sans position de gestion fidèle et fiable, impossible d'autoriser la moindre opération ;
- accepter – ou refuser - d'accorder un crédit ;
- recouvrer les fonds prêtés, c'est-à-dire collecter les remboursements de ces mêmes crédits ;
- calculer les montants des ressources financières à se procurer à court, moyen et long terme, afin de se les procurer au meilleur coût ; c'est le rôle de la gestion de bilan actif-passif ou ALM en anglais ;
- restituer des situations, ce qui est tout autant utile pour le client que pour le banquier, le premier cherchant à savoir où il en est, le second quels produits additionnels il peut espérer lui vendre ! Sans oublier les Tiers et en particulier les autorités de tutelle.

Attardons-nous un instant sur les fonctions 3 et 4. Dans un organisme de crédit, elles s'avèrent particulièrement critiques : un dysfonctionnement prolongé aurait un effet immédiat sur la rentabilité de l'entreprise. Comme l'a démontré la crise des « sub-prime », une erreur qui consisterait à octroyer des crédits sur une population à risque, peut même se traduire à moyen terme par la mise en péril de l'organisme financier.

Il en est de même pour le recouvrement : la capacité de la banque à ramener un client dans le droit chemin en cas d'impayé, et ce le plus rapidement possible, a un impact direct sur son résultat et se mesure au jour par le jour par le coût du risque.

Il n'est donc pas étonnant de voir que dans la banque, et tout spécialement pour les organismes de crédit, ces fonctions font l'objet d'une recherche permanente d'optimisation, de mise en œuvre de nouvelles stratégies, et de mesure d'efficacité.

Pour autant, il est intéressant de noter que ces fonctions ne sont pas et de loin, les plus coûteuses. Mais pour en arriver à ce constat (et par conséquent, essayer de diminuer le coût des autres fonctions, moins critiques, encore faut-il se donner les moyens de mesurer combien elles coûtent !

Mesurer le coût des fonctions

De tous les 'objets' manipulés par le S.I., la fonction est certainement le plus pertinent à la fois pour en analyser la valeur, mais aussi pour en mesurer les coûts.

L'analyse des exigences, qui revient à la mode, se pose en concurrente. Certains de ses promoteurs militent pour l'estimation et le suivi des coûts des exigences. Mais peut-on coter une exigence sans avoir une idée de la solution ? En pratique, le coût de l'exigence dépend de la manière dont les fonctions destinées à la satisfaire, vont être mises en œuvre : comme il est plus économique de fabriquer un vase en matière plastique qu'en métal, il est plus coûteux de développer avec un L3G qu'avec un L4G...

Chaque exigence portant sur le produit logiciel peut être ramenée à une fonction : le temps de réponse, le taux de disponibilité, n'ont de sens que par rapport aux fonctions sur lesquels ils portent.

Contrairement à ce que l'on pourrait croire, il est relativement aisé de collecter le coût des fonctions, qu'il s'agisse des investissements en développement, mais aussi de la maintenance, de l'exploitation, ou de support.

A partir des référentiels de bonnes pratiques ITIL et CoBiT, Conix Consulting a mis au point une matrice de collecte des coûts des fonctions. Basée sur la méthode ABC¹, elle comprend 29 activités réparties entre la planification, la construction, l'exploitation et l'administration des S.I.

Parmi ces 29 activités, 21 d'entre elles peuvent être directement imputées à des fonctions. Les 8 restantes représentent moins de 13% des coûts d'une DSI : il est donc légitime de les répartir au-pro-rata des autres coûts, de la même manière que dans une entreprise, on répartit le coût de la Direction Générale au prorata du Chiffre d'Affaires des Centres de Profit.

¹ *Activity-Based Costing : méthode détaillée d'analyse des coûts de revient, qui consiste à examiner les activités consommées par chaque produit/service, en la découplant de la consommation des ressources par ces mêmes activités. Elle repose sur la détermination préalable des inducteurs de coût d'une activité.*

Un enjeu de gouvernance

L'analyse de la valeur appliquée à l'informatique reste un exercice subtil, qui demeure encore peu répandu, malgré quelques initiatives plus ou moins abouties¹.

Comme le définit l'AFNOR, la valeur se mesure au rapport besoins satisfaits/coûts. Dans l'industrie, cette analyse de la valeur repose sur l'étude des fonctions. Une certitude donc : la fonction matérialise le meilleur niveau de maille pour évaluer la valeur d'un patrimoine informatique.

Prenons un exemple : les systèmes de pilotage. Ils font appel à deux types de composants logiciels : les ETL, qui permettent d'alimenter les entrepôts de données, et les outils de restitution, qui permettent à l'utilisateur de naviguer dans ces données.

Il est clair que pour les utilisateurs, seuls les seconds ont de la valeur. Les ETL ne sont là que pour éviter « d'attaquer » directement les bases de données des systèmes opérationnels, ce que l'état de l'art actuel ne nous permet pas encore d'envisager.

Conclusion : en attendant ce jour, il faut donc investir le moins possible dans les ETL.

Forts de ce constat, nous avons mis en place une technique pour industrialiser ce processus d'alimentation. Cette démarche a permis de dégager une économie substantielle.

Une mission pour l'architecte fonctionnel

Que faire pour mettre en place une véritable analyse de la valeur du SI ?

La première étape consiste à établir une liste standard des fonctions du S.I., applicable à toute l'entreprise, et recouvrant tous les Métiers, toutes les unités et tous les pays où celle-ci est implantée.

Pour cela, il est souhaitable de faire appel à un architecte fonctionnel expérimenté. Sa contribution consistera à identifier et à normaliser les véritables fonctions, leurs variantes, les possibilités de paramétrage et les cas d'utilisation. Pour reprendre l'exemple du calcul de crédit, une seule fonction suffit pour calculer soit le capital qu'il est possible d'emprunter pour un remboursement donné, soit la mensualité en fonction du capital : en effet, la formule à appliquer est strictement la même.

Ce référentiel de fonctions aura de multiples usages, dont celui de servir de check-list prête à l'emploi permettant aux maîtrises d'ouvrage de 'faire leur marché', et aux maîtrises d'œuvre de savoir ce qu'elles auront à coder, ou à prendre sur étagère parmi les solutions déjà disponibles.

Stop à la prolifération !

Une fois identifiées les fonctions, l'enjeu est de réduire au maximum les composants applicatifs qui assurent ces fonctions. Mission bien plus difficile, car il s'agit maintenant de lutter contre une tendance naturelle aux organisations : avoir sa propre solution ! On le sait, chaque Métier se considère comme particulier, et n'envisage donc qu'à contrecœur de partager la solution d'un autre.

Pour ne citer qu'un exemple vécu, tel Directeur du recouvrement amiable souhaitera absolument mettre en place 'son' progiciel. Or, l'analyse montre que 95% des fonctions du recouvrement sont identiques à celles de la vente.

Une liste de fonctions standardisée permet de détecter les redondances, et par là-même, d'identifier les possibilités de mutualisation, pour réduire le nombre de ces fonctionnalités non rentables !

Standardiser les flux et les interfaces

Dans toute entreprise, les systèmes informatiques dépensent beaucoup d'énergie à envoyer des informations, en interne, ou à des systèmes externes : clients, fournisseurs, autorités de tutelle.

Si tous ces flux avaient vraiment une valeur, alors à périmètre fonctionnel égal, un système qui échange 1000 flux aurait plus de valeur que celui qui n'en échange que 200... Du point de vue de l'utilisateur, il faut se rendre à l'évidence : c'est tout le contraire. Ces flux et interfaces n'ont aucune valeur ajoutée, et l'idéal serait d'en user le moins possible.

L'architecte fonctionnel a un double rôle à jouer : déterminer les bons découplages applicatifs, et standardiser le contenu des messages échangés.

Un standard de représentation

Un référentiel de fonctions fournit un modèle standard de représentation des systèmes d'information. Il s'avère particulièrement utile pour comparer des solutions, et documenter des projets.

Il fournit un appui aux décideurs pour appréhender les choix, et orienter les évolutions du patrimoine informatique. En cela, il contribue de manière significative à la bonne gouvernance du S.I.

guy.boizard@conix.fr

¹ On peut citer MAREVA qui intègre une analyse qualitative de la valeur ajoutée, couplée à des évaluations du ROI, de la VAN, ...