

L'INFORMATIQUE

P R O F E S S I O N N E L L E

Dossier - Valeurs et performances

Combien ça coûte ?

Roland Chanut, Yves Constantinidis, Jean Supizet, Catherine Leloup, Christophe Legrenzi, André Schwob, Frédéric Hervé, Fabien Cleuet, Olivier Le Gendre

Gestion des connaissances

Un bénéfice indéniable

Denis Meingan

KM et communautés de pratique

Un retour d'expérience

Eric Crouzet

Pré-contrat d'infogérance

La lettre d'intention

Isabelle Demnard-Tellier

sommaire

L'Informatique Professionnelle n° 209 décembre 2002

DOSSIER VALEURS ET PERFORMANCES

- 4** **Management par la valeur**
Maximiser la performance et motiver
En s'appuyant sur l'analyse de la valeur, le management par la valeur permet de maximiser la performance de l'entreprise, de renforcer les synergies et de motiver. Explications !
Roland Chanut
- 10** **Logiciel**
Négocier et construire la valeur d'un produit
La valeur d'un logiciel n'est pas la même pour tous. Négocier par la valeur permet de rapprocher les points de vue. Utiliser un langage commun de qualité maximise les résultats.
Yves Constantinidis
- 15** **Total Balanced Scorecard**
Un pilotage aux instruments
Le Total Balanced Scorecard permet de remettre la valeur au centre des préoccupations des acteurs.
Jean Supizet
- 21** **Projets**
La valeur naît de l'efficacité
La valeur d'un projet ne se résume pas à ses coûts, mais s'étend à son efficacité. La création de valeur résulte d'une démarche transverse fondée sur les processus.
Catherine Leloup
- 24** **Méthode**
Comment mesurer la valeur des SI ?
Que l'on soit au niveau du fonctionnement ou des études, il est devenu indispensable de savoir gérer ses coûts et de maîtriser les mécanismes de création de valeur.
Christophe Legrenzi
- 28** **Management**
Dire "non", souvent ... pour dire "oui", vraiment
Il faut savoir refuser des projets pour ouvrir à nouveau le champ des possibles, laisser de la place à l'expérimentation et atteindre les objectifs fixés. C'est le seul moyen pour créer de la valeur et être performant.
André Schwob
- 31** **Management**
Mieux valoriser les investissements
Immobiliser la production informatique, c'est lui reconnaître une valeur marchande. La performance d'une équipe de projet peut alors être analysée, en termes de résultats.
Frédéric Hervé
- 32** **Maintenance**
Préservons la valeur du SI
La valeur et la performance d'un système d'information dépendent pour beaucoup de son efficacité sur la durée. Pour préserver la valeur, il faut aussi savoir assurer des maintenances de qualité.
Fabien Cleuet
- 36** **Stratégie**
Le clair et l'obscur
La valeur de l'informatique est toujours difficile à prouver. Trois règles d'or permettent pourtant de s'y retrouver : identifier les objectifs "valables", remplacer les investissements informatiques par des investissements business, établir une métrique de contrôle a posteriori.
Olivier Le Gendre

...ET AUSSI...

GESTION DES CONNAISSANCES

- 38** **KM et communautés de pratique**
Un bénéfice indéniable
Diverses dans leur forme et leur objet, les communautés de pratique concrétisent les notions de management des connaissances. Les expériences réussies se multiplient.
Denis Meingan
- 45** **KM et communautés de pratique**
Un retour d'expérience
Dans cette entreprise, la mutualisation des savoirs est devenue priorité. Très vite, les deux premiers projets ont fait des petits et ce sont quatre communautés de pratique qui ont été définies. Retour d'expérience sur une approche réussie.
Eric Crouzet

ARRETS ET TENDANCES

- 48** **Pré-contrat d'infogérance**
La lettre d'intention
Le pré-contrat est indispensable à la mise en œuvre d'une infogérance. Mais attention, une lettre d'intention peut, en fait, cacher un véritable engagement contractuel.
Isabelle Demnard-Tellier



J.M. Atzel

COMITÉ ÉDITORIAL :

Jean-Pierre Corniou
Catherine Leloup
Jean-Claude Maury
Christian Morfouace
Jacques Pantin
Pierre Lora-Tonet
André Schwob
Serge Yablonsky

**DIRECTEUR
DE LA PUBLICATION :**
Norbert Miconnet

RÉDACTEUR EN CHEF :
Jean-Marc Berlioux (1502)

RÉDACTEUR EN CHEF DELEGUE :
Jean-Michel Atzel

GESTION DES ABONNEMENTS :
Sylvie Garofalo (1518)

SIEGE SOCIAL :
Gartner
Immeuble Défense Bergères
TSA 40002
345 avenue Georges Clémenceau
92882 Nanterre cedex 9
Tél : 01 41 35 15 15
Fax : 01 41 35 15 10

TARIFS ABONNEMENTS :
France 410 € (tva 2,10 %)
Hors France 430 €

ISSN 0750-1080
Commission Paritaire 61050
RC 350 624 102
SARL au Capital de 162 000 €

IMPRIMEUR :
Imprimerie Moderne de Bayeux
7 rue de la Résistance
BP 133
14401 Bayeux cedex
Tél. 02 31 51 63 20

Combien ça coûte ?

Certaines choses ont un coût, d'autres ont de la valeur. Rapportée à l'informatique, cette distinction sémantique montre tout le chemin qu'il reste à parcourir pour passer de la notion de budget à la notion d'investissement et pour passer de la notion de dépenses à celles de gains, d'usage et de valeur d'usage.

Car la différence entre coût et valeur cache, de part et d'autres, bien des malaises et des incompréhensions. Ce qui coûte ici, à la direction des systèmes d'information, à la direction financière ou à la Direction Générale n'a pas forcément la même valeur là, dans les directions opérationnelles et chez les utilisateurs.

Résultat : ce qui pour l'utilisateur peut se révéler d'une grande valeur, c'est-à-dire capable de générer une meilleure performance et une plus grande efficacité, pourra, faute de compréhension de l'enjeu et de valorisation préalable des fonctionnalités, être passé à la trappe du développement ou reporté ultérieurement afin de tenir des délais et des coûts purement financiers.

De la même manière, ce qui à première vue peut apparaître comme une dépense urgente et incontournable par les maîtrises d'ouvrage peut se révéler sans valeur et sans usage une fois le temps passé et l'application installée.

Le coût, la valeur et la performance : voilà donc le triptyque salutaire qui devrait être inscrit en lettres rouges au fronton de nos directions des systèmes d'information.

Trois notions extrêmement imbriquées, parfois difficilement mesurables, et qui ne peuvent aller les unes sans les autres, au risque de perdre de vue la stratégie et l'efficacité.

C'est cette combinaison des termes de l'échange entre valeur, coût et performance, entre la pratique officielle et l'usage de terrain qui est au cœur de ce nouveau numéro de l'IP. Construire la valeur du système d'information, la négocier, maximiser la performance, accroître l'efficacité, mais aussi mesurer, valoriser et pérenniser la valeur sur la durée, tels sont les principaux thèmes traités.

Mais la réflexion sur le vocabulaire et les enjeux, sur le management par la valeur ou la valorisation de la chaîne de production, ne peut, en ces temps de "vaches maigres" faire l'impasse sur les réalités. Car même préparé, analysé, valorisé, tout investissement est toujours de toute façon confronté au bon sens pécuniaire de nos dirigeants. Et l'éternelle question "combien ça coûte ?" porte déjà en elle-même la sanction du couperet financier.

Car au final, bien sûr, c'est toujours les cordons de la bourse qu'il faut délier !

Jean-Michel Atzel

VALEURS ET PERFORMANCES

LOGICIEL

Négocier et construire la valeur d'un produit

La valeur d'un logiciel n'est pas la même pour tous. Négocier par la valeur permet de rapprocher les points de vue. Utiliser un langage commun de qualité maximise les résultats.



Yves Constantinidis

Consultant à BSGL Conseil

Qui a trempé dans quelques projets informatiques, d'une certaine envergure, sait que les relations entre la maîtrise d'œuvre et la maîtrise d'ouvrage sont souvent difficiles, parfois même houleuses. Les conséquences se font ressentir sur le déroulement du projet et sur le livrable final. Les retards de livraison, un logiciel de qualité médiocre (voire totalement inutilisable) ont souvent pour origine une situation tendue, parfois conflictuelle, entre deux entités qui devraient, dans l'idéal, travailler la main dans la main.

Comment dépasser ce constat ? Comment faciliter le dialogue entre maîtrise d'œuvre et maîtrise d'ouvrage, condition nécessaire au succès d'un projet ? Avant de rechercher une solution, faisons une petite liste des difficultés de dialogue les plus fréquentes entre maître d'ouvrage (ou client) et maître d'œuvre (ou fournisseur), et balayons aussi les causes de ces difficultés.

- Le client hésite, il tergiverse, il ne sait pas ce qu'il veut. Cette attitude irrite les informaticiens, mais elle est normale de la part d'un maître d'ou-

vrage. Ce dernier ne connaît pas les possibilités offertes par une technologie. De plus, il n'a aucune idée de l'effort nécessaire au développement d'une fonction donnée. Il ne sait pas ce qui est réaliste d'exiger et ce qui ne l'est pas. Il nous est arrivé d'entendre un client nous dire qu'il n'a pas osé demander telle fonctionnalité, pensant qu'elle serait trop difficile à implémenter.

- Le client s'exprime difficilement, il omet de spécifier des besoins fondamentaux, ou au contraire il s'engouffre dans des détails inutiles. Il n'exprime pas certaines exigences, qui sont pour lui évidentes, donc sous-entendues. Or, ce qui est évident pour les uns ne l'est pas pour les autres. Ce principe est à la base des méthodes de négociation raisonnée, mais aussi de toute démarche qualité. Une autre cause de cette difficulté à s'exprimer tient au vocabulaire. Nous reviendrons sur ces deux points.

- Les objectifs et les enjeux du projet sont flous. Cela est gênant pour l'avancement du projet, mais très fréquent. Des enjeux politiques, des objectifs stratégiques peuvent être, intentionnellement ou non, passés sous silence. La solution de facilité consiste alors à "faire du flou sur du flou" et à élaborer, par exemple, un cahier des charges truffé d'ambiguïtés et de non-dits.

“

La solution de facilité consiste à "faire du flou sur du flou" et à élaborer un cahier des charges truffé d'ambiguïtés et de non-dits

”

- Les contraintes en termes de coûts et de délais ne sont pas claires. Cela arrive fréquemment dans les grandes entreprises et dans le secteur public, où une solution est développée en interne. Dans de telles structures, hautement hiérarchisées, les informaticiens sont à la fois tout puissants et taillables et corvéables à merci. Les négociations ne sont pas basées sur une grandeur mesurable (l'argent) mais sur des rapports de force.

- Les rôles ne sont pas bien définis. Le maître d'ouvrage doit exprimer le produit fini, le "quoi". C'est au maître d'œuvre d'expliquer "comment" il va faire, et le faire. Dans de nombreux cas, fournisseur et client se substituent l'un à l'autre, tout simplement parce que les rôles n'ont pas été bien définis au départ.

Quelques raisons d'avoir confiance

Les difficultés de dialogue sont donc nombreuses, et influent négativement sur la qualité du logiciel. Cependant, il est possible de les aplanir ou de les contourner. En prélude à notre méthode de "négociation par la valeur", examinons les efforts que la maîtrise d'ouvrage est prête à faire pour contribuer à la qualité du logiciel.

- La maîtrise d'ouvrage est prête à faire un effort, pour comprendre les aspects techniques, s'ils sont présentés avec rigueur et simplicité, et si le maître d'œuvre fait un effort de pédagogie. Ce n'est pas parce qu'il est banquier, assureur ou médecin, en tout cas pas informaticien, que le client doit être laissé dans le noir, ou considéré comme techniquement passif. L'expérience montre que ce dernier est tout à fait disposé à examiner, par exemple, l'architecture d'une solution, du moins les aspects le concernant. C'est alors aux informaticiens de faire un effort de pédagogie, de dégager l'essentiel des concepts, de présenter des schémas clairs, dépouillés des détails inutiles, d'utiliser un vocabulaire adéquat, un langage directement compréhensible par le client, et un vocabulaire lié à sa problématique.

- La maîtrise d'ouvrage accepte la collaboration, les séances communes de "remue-ménages".

Très souvent, le client est réticent quant à la participation des utilisateurs finaux. Comment impliquer ces derniers dans le projet tout en les empêchant de le vampiriser ? Comment tenir compte de leur point de vue sans qu'ils viennent parasiter les spécifications ? Ce sont là des craintes légitimes du client, qui a peur de perdre le contrôle de la situation, et ce, d'autant plus que les rapports avec le maître d'œuvre sont tendus, le vocabulaire mal maîtrisé, l'architecture obscure. Une des solutions consiste à dire au client : "bâtissons ensemble l'architecture globale de la solution. Soyons clairs sur le noyau dur, les exigences qui doivent rester intangibles, et fermes sur les choix de base. Faisons intervenir les utilisateurs par la suite, afin que ceux-ci apportent leur contribution sur les points qui relèvent de leur travail au quotidien". De cette manière, on utilise au mieux le savoir-faire du terrain sans mettre en cause la stabilité de l'édifice.

- La maîtrise d'ouvrage s'intéresse à l'avancement des projets et est à l'affût d'indicateurs clairs, précis, synthétiques. L'erreur fréquente consiste à maintenir le maître d'ouvrage dans le "black-out" des coûts, des charges et des délais. Ou, au contraire, à le noyer sous des chiffres et des plannings n'ayant aucune signification concrète pour lui. Un tableau mensuel d'avancement du projet ainsi qu'une page de rapport sont, en général, suffisants. Les jalons choisis doivent être significatifs pour le client. Ils doivent correspondre à des étapes bien visibles et vérifiables par lui, faute de quoi il ne pourra rien en tirer.

Négocier par la valeur

Si maître d'œuvre et maître d'ouvrage sont prêts à faire ces efforts, c'est un bon début pour une négociation raisonnée des exigences. Pour que cette situation ne se dégrade pas au cours de l'avancement du projet, le principe à appliquer est simple, mais souvent très difficile à mettre en œuvre : il faut passer de la "logique de l'honneur" à la logique du contrat. Les contrats existent, le cahier des charges et le plan qualité d'un projet en sont deux exemples. En nous appuyant sur ces contrats, appliquons les principes de la négociation raisonnée.

“
Les informaticiens sont à la fois tout puissants et taillables et corvéables à merci

”

“
C'est aux informaticiens de faire un effort de pédagogie

”

“

Soyons hard avec le soft

”

• S'attaquer au problème et non aux personnes. Devant un logiciel défaillant, inadapté ou inutilisable beaucoup de clients s'en prennent au fournisseur, le mettent à la porte, ou au contraire, après négociation, se contentent de quelques modifications "cosmétiques" et concèdent que "tout finit par s'arranger". Ce faisant, ils se trompent de cible. Il ne s'agit pas d'être dur ou tendre avec le fournisseur, il s'agit d'obtenir satisfaction sur un certain nombre de points précisés dans un cahier des charges. Ainsi, "plutôt que de vouloir changer le maître d'œuvre, soyons hard avec le soft" !

• Elargir l'analyse des besoins. La plupart des cahiers des charges actuellement produits sont des cahiers des charges fonctionnels. Or, une exigence d'un utilisateur n'est pas uniquement fonctionnelle. Il est donc indispensable qu'un cahier des charges tienne compte non seulement des fonctionnalités requises, mais de tout ce que l'utilisateur et le maître d'ouvrage sont en droit d'exiger : fiabilité, sécurité, performances, etc. D'habitude, de telles caractéristiques sont spécifiées dans un document distinct du cahier des charges, le contrat de service. Il est cependant possible de regrouper les exigences fonctionnelles et non fonctionnelles dans un seul document, un cahier des charges unique. Cela permet d'uniformiser le processus de recueil et de formalisation des exigences, et d'appuyer le fait que la qualité d'un logiciel est un tout, qu'un logiciel peut être inacceptable malgré une couverture fonctionnelle suffisante.

• Changer de vocabulaire. La satisfaction des exigences fonctionnelles et non fonctionnelles déterminent la valeur du logiciel. Le terme de "valeur" semble ici plus approprié que le terme de "qualité", traditionnellement utilisé, car la qualité évoque quelque chose de net, de précis, de rigide, alors que la valeur est par définition négociable. Elle dépend du client, de la conjoncture, et de la date de livraison. Un logiciel excellent peut perdre toute valeur en l'espace de quelques mois. La valeur ne dépend pas que du fournisseur. Elle dépend aussi de l'utilisateur qui est

fait du produit. Elle dépend de l'aptitude des utilisateurs à se l'approprier, et donc de la manière dont ils ont été informés et formés. Parler de valeur, c'est déjà responsabiliser le maître d'ouvrage, l'amener à adopter une attitude proactive, à penser par objectif, à tenir compte du facteur coût et du facteur temps. En d'autres termes à tenir son rôle, c'est-à-dire : maîtriser l'ouvrage.

La perspective de l'utilisateur

Parler de valeur, grandeur fluctuante, nous mène rapidement à nous poser la question suivante : de la valeur pour qui ? Un logiciel n'a de valeur que si l'on s'en sert. Si les enjeux et la stratégie sont effectivement fixés par le maître d'ouvrage, la valeur réelle d'un logiciel ne dépend, en réalité, que de l'utilisateur.

Or, malgré toutes les recommandations méthodologiques, les utilisateurs sont insuffisamment impliqués dans les projets informatiques, en particulier lors de la phase d'expression des besoins. Les maîtrises d'ouvrage négligent de faire participer les utilisateurs, ou les écartent intentionnellement. Il est vrai que ces recueils méthodologiques sont muets sur la manière de trouver des utilisateurs véritablement représentatifs, de les sélectionner, de les interroger. Seule l'expérience et l'intuition peuvent nous guider.

Les difficultés d'expression des besoins, une fois les "utilisateurs représentatifs" trouvés, sont nombreuses. Citons quelques difficultés récurrentes, et quelques moyens de les résoudre.

• Besoins mal exprimés : un utilisateur ne parle pas le même langage que la personne qui va l'interroger. Il aura tendance à utiliser son jargon professionnel. Il est nécessaire pour cela d'élaborer un langage commun.

• Besoins non exprimés : l'utilisateur aura tendance à passer sous silence des besoins qui lui paraissent évidents, mais qui ne le sont pas nécessairement pour la personne qui va les recueillir.

“

Parler de valeur, c'est déjà responsabiliser le maître d'ouvrage

”

- Besoins manquants ou incomplets : il est rare qu'un seul utilisateur, lors d'un seul entretien, exprime tous ses besoins. D'où la nécessité d'associer ponctuellement plusieurs utilisateurs de même profil à la définition des besoins, et de faire valider les entretiens.

- Besoins conflictuels : deux utilisateurs de profils différents (et aussi, très souvent, deux utilisateurs d'un même profil) auront des besoins différents, et parfois contradictoires. La personne chargée de recueillir les besoins devra jouer le rôle d'arbitre ou de médiateur, en cherchant à connaître les enjeux, en classant les besoins par priorités, sachant que les priorités ne sont pas les mêmes pour tous.

- Besoins redondants : deux utilisateurs peuvent exprimer le même besoin, mais sous des formes différentes. En particulier, la tentation est grande, pour un utilisateur, d'exprimer un besoin sous forme de solution. C'est souvent en croisant les

propositions de solution que l'on pourra remonter au véritable besoin.

- Besoins superflus : un utilisateur ne connaît pas en général le coût de la mise en œuvre de ses exigences. Très souvent, il demandera des fonctionnalités dont il n'aura qu'un usage très limité, et dont le coût sera disproportionné par rapport au bénéfice généré. Les besoins "bruts" devront donc être raffinés avant d'être intégrés dans le cahier des charges.

Il est clair que le chef de projet maîtrise d'ouvrage, même (et surtout) dans le cas où il aurait délégué à un consultant la phase d'expression des besoins, doit reprendre le contrôle de la situation et jouer un rôle d'arbitre.

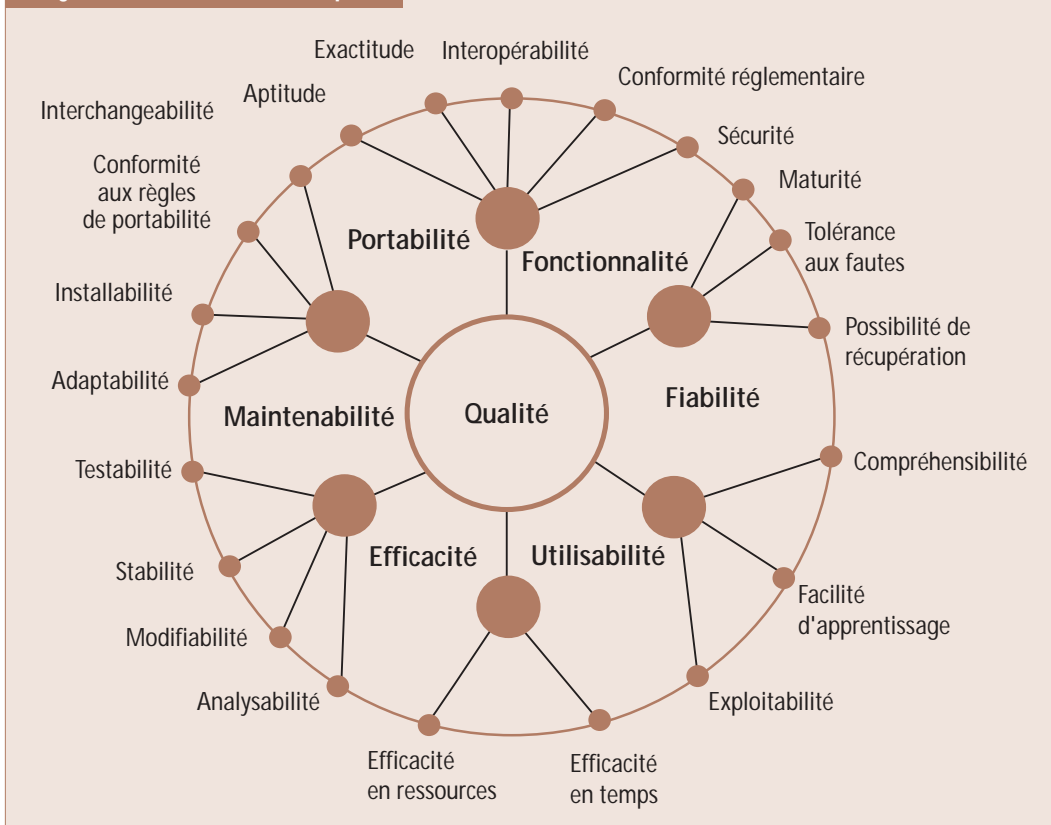
Elaborer un langage commun

Osons une métaphore musicale : les acteurs d'un projet jouent chacun d'un instrument différent. Les

“
La personne chargée
de recueillir les besoins
devra jouer le rôle
d'arbitre

”

Figure 1 • le vocabulaire de la qualité



“
Un utilisateur ne
connaît pas en général
le coût de ses exi-
gences

”

“

Les langages de modélisation ont pour ambition de rapprocher l'informaticien et son client

”

savoir-faire et les techniques n'ont rien de commun. Pourtant, ils doivent tous jouer le même morceau. Pour que chacun s'y retrouve, il est donc indispensable d'inventer un langage commun. Certes, les sons du piano et celui de la flûte ne sont pas les mêmes ; cependant, flûtiste et pianiste utilisent un langage universel. Ajoutons, pour poursuivre la métaphore, que ce langage commun est d'autant plus nécessaire que ces deux instrumentistes peuvent venir de milieux très différents et parler des langues différentes.

Ce vocabulaire universel existe-t-il ? En ce qui concerne les exigences purement fonctionnelles, la réponse n'est pas tranchée. Certes, les langages de modélisation (Merise, UML ou autres) ont pour ambition de rapprocher l'informaticien et son client. Ils ont leur utilité, mais leur maniement n'est pas si simple, et dans les cas complexes, ils montrent leurs limites.

En ce qui concerne les aspects "non fonctionnels" (au sens courant du terme) un vocabulaire commun existe. Il est précisément et formellement défini par une norme internationale : la norme ISO 9126 sur l'évaluation de la qualité du logiciel. La qualité est définie à l'aide de six caractéristiques se décomposant en une vingtaine de sous-caractéristiques. Le vocabulaire utilisé dans cette norme n'est pas toujours très parlant pour un novice, mais les définitions sont claires, lapidaires, et compréhensibles par tous (voir figure 1).

La décomposition de la qualité du logiciel en un nombre restreint d'attributs présente un avantage certain : elle force à une vision globale du logiciel, pouvant être saisie par un non informaticien, alors qu'une décomposition plus fine ou plus spécifique morcelle la vision et attise les combats entre experts.

La mise en œuvre sur le terrain

A l'origine, cette norme ISO 9126 a pour objectif l'évaluation de la qualité. Il est clair cependant que les critères qui permettent d'évaluer la qualité du logiciel permettent aussi de la construire.

Les caractéristiques de qualité peuvent alors devenir une base pour la négociation entre la maîtrise d'œuvre et la maîtrise d'ouvrage. La définition de ces différentes caractéristiques apporte un vocabulaire commun, stable et indépendant du domaine d'application.

Qu'il s'agisse de sélectionner un progiciel ou de développer un spécifique, c'est sur ce schéma que nous appuyons notre dialogue. Nous commençons par présenter les six caractéristiques au maître d'ouvrage, en lui demandant de nous indiquer celles qu'il considère comme prioritaires. Rapidement, le client prend conscience de la nécessité d'établir un équilibre entre la satisfaction des caractéristiques fonctionnelles et non fonctionnelles. Nous descendons progressivement dans le détail. Et le maître d'œuvre reçoit un cahier des charges sans ambiguïté. Cet ensemble de caractéristiques servira de fil conducteur tout au long du projet, jusqu'à la vérification en service régulier.

Insistons sur un point important : ces caractéristiques concernent essentiellement l'ouvrage, c'est-à-dire le produit logiciel. Il ne s'agit pas là d'un jargon informatique, mais d'un vocabulaire que tout maître d'ouvrage pourra manier avec facilité, pourvu que ce vocabulaire lui soit présenté clairement. Il fait appel à des concepts informatiques, mais ces concepts peuvent être assimilés sans difficulté par un non-informaticien. Notre expérience prouve que cet effort d'assimilation en vue d'acquérir un vocabulaire commun entre les informaticiens et leurs clients est un investissement extrêmement rentable.

Yves Constantinidis
yconstan@easynet.fr

➔ Bibliographie

Yves Constantinidis, *Le logiciel à valeur ajoutée*, Hermès, 2001.