

# sommaire

L'Informatique Professionnelle n° 229 décembre 2004

## DOSSIER SPÉCIAL LOGICIELS ET SERVICES

- 4 Développement**  
**Modélisons nos métiers avec UML**  
UML permet de modéliser les processus métier et les entités métier en utilisant les mêmes symboles et les mêmes outils que les équipes de développement utilisent pour modéliser leurs systèmes informatiques. C'est un avantage considérable !  
*Pascal Roques*
- 11 Programmation système**  
**Il faut revenir aux sources !**  
Le programmeur système n'a pas besoin en théorie d'étudier les sources du système d'exploitation pour lequel il conçoit des applications. Mais l'étude des sources peut être profitable.  
*Patrick Cégielski*
- 16 .NET**  
**Un premier bilan très positif**  
.NET a gagné son pari : s'imposer ! Mais Java est encore là ! .NET profite du ralentissement de l'innovation du standard Java et propose des solutions équivalentes et parfois plus économiques.  
*Eric Groise*
- 21 Architecte, ergonomiste et qualificateur**  
**Trois qualités du designer informatique**  
Le designer de produit logiciel est un hybride entre l'architecte, l'ergonomiste, et le qualificateur. C'est un généraliste, qui établit et maintient le dialogue entre les différents acteurs, aux profils et aux vécus fort différents. Il allie intuition et expérience pour innover et formuler des solutions concrètes.  
*Yves Constantinidis*
- 26 Libérer les logiciels**  
**Une idée simple, une démarche exigeante**  
Plus que la libération ponctuelle du code informatique, c'est l'investissement durable et intelligent de l'entreprise qui donne sa valeur à la libération du logiciel.  
*Antoine Pitrou*
- 30 L'informatique à la demande**  
**Rêve ou réalité**  
"Computer on demand" et "Grid computing" sont des techniques bien prometteuses. Pourtant, quelle est la véritable opportunité financière de telles évolutions face à des offres de BPO, de web services et d'ASP ?  
*Jean-Marc Lejeune*
- 35 Sociétés de services**  
**L'état d'urgence est décrété**  
Avec une baisse des prix jamais égalée, le marché des services informatiques offre aux clients des latitudes commerciales inespérées. Ce réajustement entraîne une modification de la relation client-fournisseur.  
*Pierre Jourdan*
- ...ET AUSSI...**
- EXTERNALISATION**
- 41 Réversibilité**  
**Les critères d'une réussite**  
La réversibilité est l'une des protections les plus efficaces du client vis-à-vis de ses prestataires. Mais pour réussir, il faut de la méthode et beaucoup de rigueur. Explications !  
*Sonia Boittin*
- ARCHITECTURE**
- 46 Les Web Services Transactionnels**  
**Une évolution incontournable !**  
Une plate-forme de services web intégrés libère l'utilisateur de la contrainte de gérer ses contenus. Ce modèle d'usage, basé sur la fourniture de services transactionnels, s'imposera rapidement.  
*Jean-Marc Lévy Dreyfus*

**J.M. Berlioux**

Mensuel publié par Gartner France  
Tél. 01 71 01 31 00  
Fax 01 71 01 32 32

**COMITÉ ÉDITORIAL :**

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

**DIRECTEUR****DE LA PUBLICATION :**

Johan Conix

**RÉDACTEUR EN CHEF :**

Jean-Marc Berlioux

**RÉDACTEUR EN CHEF DELEGUE :**

Jean-Michel Atzel

**SIÈGE SOCIAL :**

Gartner France  
Immeuble Triangle de l'Arche  
9-11, cours du Triangle  
92937 Paris La Défense cedex  
Tél : 01 71 01 31 00  
Fax : 01 71 01 32 32

**GESTION DES ABONNEMENTS :**

OCIFAM  
34, quai de l'Aisne  
93500 Pantin  
Tel. : 01 41 83 52 78  
Fax : 01 41 83 54 72  
Email : bguyomard@grouperf.com

**TARIFS ABONNEMENTS :**

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

**PRE-PRESSE :**

J2C COMMUNICATION  
jc-caradot@j2c-communication.fr

**IMPRIMEUR :**

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

**CRÉDIT PHOTO :**

Montage of man holding a ball  
(Comstock Images)

ISSN 0750-1080

RC 350 624 102

SARL au Capital de 162 000 €

## Un métier de bâtisseurs

Les dernières enquêtes de Gartner montre que les postes "Logiciels" et "Services externes" représentent à eux deux, en moyenne, le tiers des dépenses des départements informatiques<sup>(1)</sup>. Quant aux travaux internes portant sur les applications et la fourniture de service aux utilisateurs, ils représentent une bonne partie des coûts salariaux, qui constituent un autre tiers des dépenses.

Au total donc, définir, construire, lancer, exploiter, et faire évoluer le portefeuille de logiciels et services de l'organisation est bien une des missions centrales d'une DSI.

Dans ce domaine, on ne peut que constater une évolution progressive mais marquée vers plus de professionnalisme, de rigueur et de qualité.

Dans le monde du logiciel, le savoir-faire individuel des artisans développeurs fait place de plus en plus à une organisation structurée des travaux. L'échelle CMMI devient le standard sur lequel s'étalonnent les différentes équipes. Nombre de "software houses" indiennes s'appuient sur des niveaux CMMI impressionnants pour asseoir leur crédibilité. D'où une évolution de "l'Etat de l'art" et la nécessité pour les DSI d'accroître leur professionnalisme.

Les taux de bogues encore parfaitement acceptables il y a une dizaine d'années deviennent inadmissibles : les failles des applications ouvertes peuvent en effet livrer les systèmes d'information à la malveillance des pirates externes... et de plus en plus souvent internes.

La complexité croissante des parcs applicatifs, au moment même où les responsables métier exigent une grande flexibilité des systèmes, ne permet plus d'accepter des architectures de type "spaghetti".

Partout, le maître mot devient la rigueur de la construction. Les développeurs informatiques, qui se considéraient naguère encore un peu comme des créateurs, vont devoir respecter une discipline équivalente à celle qui règne dans la construction d'immeubles de grande hauteur...

Le service a suivi la même voie. Les conventions de service ou "Service Level Agreements" (SLA) deviennent un point de passage obligé dans les relations entre DSI et directions métier. Or, conclure et appliquer un SLA impose, la plupart du temps, de revoir l'organisation des équipes de service, de mettre en place des outils de mesure et de former les personnels.

Cette rigueur s'exprime aussi bien en interne, quand il s'agit de faire soi-même, que dans le cadre de l'externalisation. Il faut bien en effet s'imposer à soi-même la rigueur indispensable si l'on veut pouvoir l'imposer à ses fournisseurs.

En revanche, il faut aussi savoir adapter les règles du jeu à la nature des tâches à accomplir. Lorsque le besoin n'est pas bien identifié et que le demandeur n'a pas les idées claires, il ne servirait à rien de s'engager dans une démarche rigide, qui ne pourrait que générer des frustrations. Des approches plus souples, avec prototypage et réflexion en commun, seront souvent plus efficaces.

Jean-Marc Berlioux

1/ Enquête "2004 IT Spending and Staffing Survey"

ARCHITECTE, ERGONOME ET QUALITICIEN

# Trois qualités du designer informatique

Le designer de produit logiciel est un hybride entre l'architecte, l'ergonome, et le qualitatif. C'est un généraliste, qui établit et maintient le dialogue entre les différents acteurs, aux profils et aux vécus fort différents. Il allie intuition et expérience pour innover et formuler des solutions concrètes.

La qualité du logiciel a deux faces : l'une, austère, est celle de la stricte conformité aux attentes minimales des utilisateurs et aux normes en vigueur. L'autre, rayonnante, est liée au caractère innovant du logiciel. L'une est mesurée par le "coût de la non qualité", l'autre par les bénéfices induits par un produit attrayant, confortable, agréable. Un "bon" logiciel possède ces deux faces. Il concilie rigueur et créativité. Mais un bon logiciel est une chose rare. La plupart des logiciels n'apportent que très peu d'innovations par rapport aux logiciels concurrents, qu'ils se contentent le plus souvent de copier, voire même de singer. Et beaucoup trop de logiciels sont encore loin de satisfaire les exigences minimales des utilisateurs, qu'il s'agisse de fonctionnalité, de fiabilité ou de performances techniques.

Peut-on améliorer sensiblement la qualité du logiciel, sur ces deux faces

à la fois ? Améliorer les processus de développement est certes nécessaire, mais ne joue que sur la face "conformiste" de la qualité. A l'inverse, innover sans gérer l'évolution des exigences, sans respecter les normes,

“

La qualité du logiciel est mesurée par le coût de la non qualité et par les bénéfices induits

”

aboutit aux échecs que l'on sait. Que faut-il changer pour arriver à des logiciels véritablement satisfaisants et innovants ? L'industrie manufacturière, plus ancienne que l'industrie du logiciel, pourrait nous donner des pistes intéressantes. Mais, auparavant, examinons de plus près les constituants de la qualité du logiciel.

## Trois niveaux de vision

La norme ISO 9126, relative à la qualité du produit logiciel et conçue à l'origine pour évaluer la qualité du logiciel, peut être utilisée pour construire la qualité. Selon cette norme, la qualité du logiciel peut être perçue sur trois niveaux (voir figure 1 page suivante).

- La qualité interne (internal quality) correspond au niveau structurel d'analyse, et peut être évaluée par examen du code source, de l'architecture, des spécifications.
- Le niveau externe (external quality) correspond au niveau comportemental, visible et mesurable en particulier lors de tests.
- La qualité de fonctionnement (quality in use) correspond au point de vue de l'utilisateur. Elle se décompose en quatre caractéristiques : l'ef-



Yves Constantinidis  
Consultant

ficacité (relative à l'atteinte des objectifs de l'utilisateur) ; la productivité (économie de temps, d'efforts) la sécurité (risques sur l'utilisateur, le logiciel, l'environnement) ; et la satisfaction de l'utilisateur.

Ces trois niveaux de qualité du produit sont empilés et interdépendants : la qualité de fonctionnement dépend de la qualité externe, qui dépend de la qualité interne, qui elle-même dépend de la qualité du processus.

A chaque niveau, la qualité est traditionnellement perçue et manipulée par des acteurs dont les métiers sont très différents. La qualité interne est l'affaire de l'architecte, du concepteur, du développeur. La qualité externe est "visible" du chef de projet et des équipes de validation. La qualité de fonctionnement est définie et mesurée par le maître d'ouvrage, le marketing, le consultant métier, l'ergonome, le responsable sécurité. Quant à la qualité du processus, elle est depuis longtemps sous la responsabilité des qualitatifs.

Les dépendances entre niveaux de perception de la qualité ne sont pas

exclusives. C'est d'ailleurs pour cela que des mesures doivent être effectuées à la fois sur le processus, le produit (final et en cours de développement) et la satisfaction de l'utilisateur. Imaginer qu'un bon processus aboutisse nécessairement à un bon produit conduit à des effets pervers que l'on commence à bien connaître : "notre processus est certifié, donc nos produits sont bons".

“  
**La qualité est traditionnellement perçue et manipulée par des acteurs dont les métiers sont très différents**  
 ”

Erreur grossière, car d'une part, un processus peut être excellent sans jamais avoir été certifié (voire formalisé) et, d'autre part, aucun processus n'est parfait, qu'il soit certifié ou non. Le développement du logiciel (au sens large du terme, incluant les phases d'architecture et de conception) est une activité créa-

tive, par essence non entièrement formalisable.

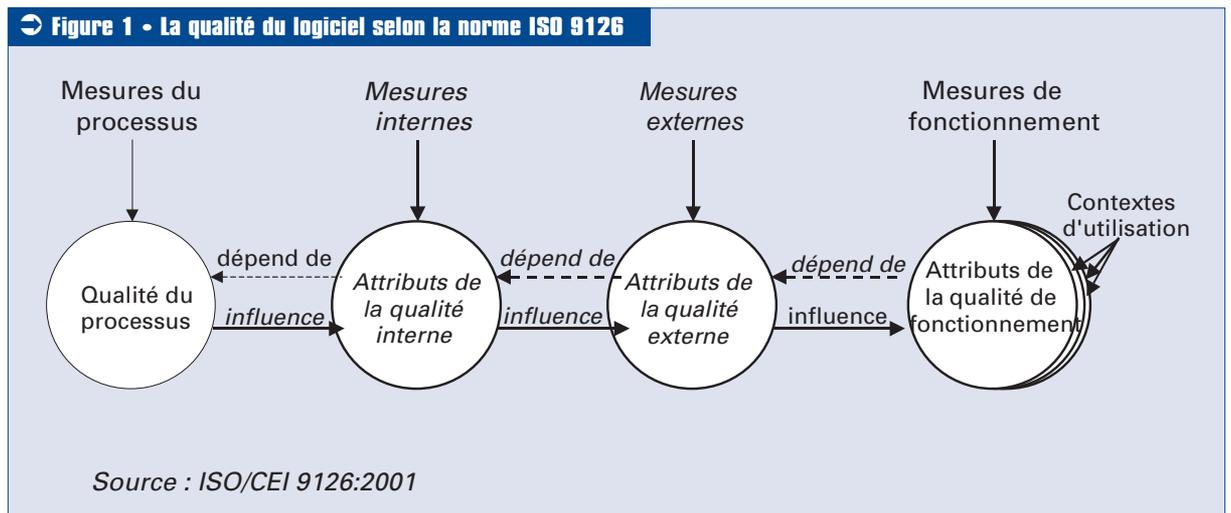
### Les caractéristiques de la qualité

Selon ISO 9126, les qualités interne et externe du logiciel sont définies au moyen de six caractéristiques et vingt-sept sous-caractéristiques.

Rappelons ces six caractéristiques :

1. capacité fonctionnelle (le logiciel remplit les fonctions exigées de lui) ;
2. fiabilité (le logiciel a peu de défaillances et récupère sans dommages suite à une défaillance) ;
3. maintenabilité (le logiciel est facile à analyser et facile à adapter, à modifier, à corriger) ;
4. portabilité (il est adaptable au contexte, aux environnements, aux plates-formes) ;
5. facilité d'utilisation (ergonomie) ;
6. rendement (il est performant en temps et économe en ressources).

Ces six caractéristiques, bien qu'indépendantes, s'influencent mutuellement et peuvent se trouver antagonistes. Ainsi, la portabilité et la maintenabilité peuvent avoir une influence négative sur le rendement ;



la fiabilité peut influencer la facilité d'utilisation ; etc.

Un logiciel sera de qualité s'il satisfait les exigences du client, eu égard à ces six caractéristiques. Cependant, chercher à maximiser toutes les caractéristiques à la fois se révèle en général irréaliste en termes de coût ou de délais. La qualité du logiciel ne vient pas de la perfection, mais de l'équilibre entre ses différents attributs.

La priorité à donner à chaque caractéristique lors de l'achat ou de la conception d'un logiciel dépend bien entendu de nombreux paramètres : type d'application, client, contexte d'utilisation, profil et culture des utilisateurs, contraintes techniques.

### Une vision globale de la qualité

Pour obtenir un produit à haute valeur ajoutée, il faut donc travailler sur toutes les caractéristiques à la fois, en cherchant un équilibre, en arbitrant entre les priorités, en tenant compte des contraintes de délai et de coût. Il ne s'agit pas tant de "faire bien du premier coup" comme on le dit souvent, que de "faire bien" dans six directions, en fonction des objectifs à atteindre et des priorités fixées en amont du développement. Pour y

arriver, plusieurs conditions doivent être réunies.

- La qualité d'un produit logiciel doit être pensée en amont, dès la phase de définition des besoins.
- Elle doit être obtenue par construction. Le produit final doit être "visualisé" dès le départ (et non simplement décrit sur le plan fonctionnel). Cela n'empêchera pas, au contraire, de modifier cette vision en cours de développement si de nouvelles idées surgissent.

“ Pour être de qualité, le produit logiciel doit être pensé de manière globale

”

- La construction de la qualité est une démarche continue. Elle est obtenue en maintenant la pression des exigences de qualité tout au long du cycle de vie du logiciel.
- C'est une démarche créative. Elle est spécifique au produit à développer. La construction de la qualité ne peut donc être entièrement formalisée.
- C'est une démarche participative, à laquelle doivent travailler tous les acteurs du projet (maître d'ouvrage,

maître d'œuvre, architecte...).

- La qualité perçue, qui correspond en grande partie à la qualité de fonctionnement, et qui dépend largement des exigences implicites des utilisateurs, doit également être pensée très en amont.

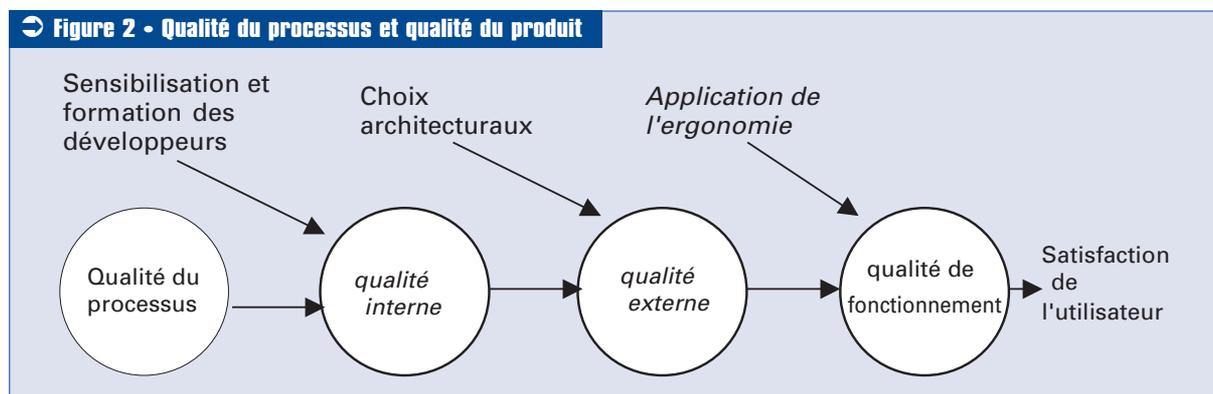
En résumé, pour être de qualité, le produit logiciel doit être pensé de manière globale, intégrant à la fois les aspects structurels et les aspects comportementaux, la vision externe et la vision interne. Autrement dit, l'architecte, l'ergonome, le qualitatif, l'expert métier et le marketing doivent pouvoir partager une vision commune et dialoguer de manière fluide. Le lien entre qualité du processus et qualité du produit peut maintenant être complété pour tenir compte de ces facteurs (voir figure 2).

### Le besoin d'un designer

Actuellement, cette vision globale de la qualité est souvent insuffisante, voire absente. Les développeurs sont insuffisamment sensibles à l'ergonomie du logiciel ; les ergonomes ne voient pas les contraintes techniques, et les qualitatifs ne sont pas en contact avec l'utilisateur.

La conséquence de cette absence de vision globale se reflète dans beau-

Figure 2 - Qualité du processus et qualité du produit



coup de logiciels, qu'il s'agisse de produits sur étagère ou d'applications spécifiques. La démarche habituelle consiste à partir d'un cahier des charges purement fonctionnel, auquel on annexe des contraintes techniques, et, dans le meilleur des cas, une charte d'ergonomie. Mais qui pense, qui visualise le logiciel à venir ? Le concepteur a trop souvent une vision très technique du produit. Le représentant des utilisateurs a souvent du mal à affirmer son point de vue. La maîtrise d'ouvrage (ou le marketing, dans le cas d'un produit sur étagère) manque souvent d'outils pour exprimer son besoin, sa vision. Résultat : le dialogue s'établit difficilement.

Alors, trop souvent, on se contente de copier les produits du concurrent, ou de réécrire l'application précédente plutôt que d'innover. Le produit final est un produit banal. Même s'il arrive en temps voulu, même s'il respecte formellement les exigences exprimées (ce qui est encore trop rarement le cas), il lui manquera justement la qualité de fonctionnement qui en fera un outil puissant, facile, sûr et agréable à utiliser. Inversement, on trouve sur le marché des produits innovants, originaux, mais qui manquent manifestement de sérieux : peu fiables, peu performants, et ne respectant pas les règles élémentaires de l'ergonomie.

Comment concilier la qualité-rigueur avec la qualité-innovation ? Ce qui s'est opéré dans l'industrie manufacturière peut nous apporter des idées nouvelles. Pour sortir de la médiocrité habituelle, pour véritablement innover, tout en continuant à industrialiser l'élaboration du pro-

duit logiciel, nous devons mettre à jour un nouveau métier : le designer de produit logiciel. Un hybride entre l'architecte, l'ergonome, et le qualitatif. C'est un généraliste, qui établit et maintient le dialogue entre les différents acteurs, aux profils et aux vécus fort différents. Il allie intuition et expérience pour innover et formuler des solutions concrètes.

“

**Le designer est un mélange d'architecte, d'ergonome et de qualitatif**

”

C'est bien d'un designer que l'on a besoin. Non pas au sens de "concepteur d'application", mais au sens actuellement utilisé dans l'industrie manufacturière. Une personne possédant une double compétence, mais surtout une personne alliant deux qualités : rigueur et créativité.

### Un généraliste

Les designers de produit logiciel n'existent-ils pas déjà ? Dans une grande mesure, oui. On les trouve souvent sous l'appellation "consultant", "assistant à la maîtrise d'ouvrage" ou "chef de produit". Cependant, il ne s'agit pas d'un métier en tant que tel. Très souvent, un consultant devient designer d'application, non pas sur la demande explicite de son client, mais parce qu'il en perçoit la nécessité en cours de mission. Ainsi, il nous est arrivé, lors d'une mission d'élaboration de cahier des charges et de suivi de projet, d'apporter des conseils en ergonomie.

Cependant, dans une grande entreprise ou sur un grand projet, pour le développement d'un produit important, un designer attiré pourrait être utile, et même indispensable pour assurer le succès du projet, voire la compétitivité du produit. Quel que soit son background, le designer est un mélange d'architecte, d'ergonome et de qualitatif. Il doit connaître les méthodes de développement, les nouvelles technologies, avoir un certain sens esthétique et des expériences variées dans le logiciel.

Voyons dans quelles situations un tel designer attiré pourrait intervenir...

### Ergonomie et qualité

Ergonomie et démarche qualité sont deux démarches complémentaires, visant à l'amélioration de la satisfaction de l'utilisateur. Mais actuellement, ces deux démarches sont totalement disjointes. Le qualitatif travaille sur le processus de développement, et l'ergonome sur l'interaction entre l'homme et le produit. Un designer travaille, avec intuition et créativité, sur la qualité perçue, et la démarche qualité devient alors bien plus qu'un ensemble de règles et de contrôles. Le designer ne se contente pas d'examiner la conformité entre le produit et le cahier des charges, il va au-delà, en participant à l'élaboration du cahier des charges, en imaginant des solutions qui sortent des sentiers battus. Il a un bon background technique, mais il sait prendre du recul par rapport à la technique pour mieux innover.

### Ergonomie, architecture et développement

L'ergonomie concerne le dialogue entre l'homme et le produit logiciel.

Mais ce dialogue ne se limite pas à ce qu'il est convenu d'appeler "l'interface graphique". Le dialogue touche au comportement du logiciel et, au-delà, à son architecture.

Inversement, les choix initiaux des technologies à mettre en œuvre prennent en compte les contraintes de sécurité et de fiabilité, mais rarement l'impact sur l'ergonomie. Cet impact est visible lorsque l'offre technologique est en phase de transition et que l'offre en nouvelles technologies est foisonnante. Par exemple, si le choix d'utiliser ou non du javascript ou des applets Java est fait par un architecte, il y a peu de chances que ce dernier étudie les impacts de ce choix sur l'ergonomie et la qualité du produit final.

De même, la qualité du produit final dépend du choix des composants de base et de la construction globale du système, c'est-à-dire de l'architecture. Ces choix dépassent les attributions du qualitatif, qui traditionnellement ne s'occupe que du processus de développement.

Un designer n'est pas un architecte, mais il sait dialoguer avec les architectes, comprendre leurs préoccupations en termes de performances ou de sécurité, et éventuellement influencer sur leur choix. Il a suffisamment de background technique pour influencer sur les choix faits par les équipes de développement et leur apporter la bonne parole en termes d'ergonomie.

Ceci est d'autant plus utile que les normes et chartes d'ergonomie ne peuvent pas être appliquées par les développeurs comme des recettes de cuisine.

### Esthétique, efficacité, élégance

Le succès d'une application, comme le succès commercial d'un produit sur étagère, est lié à l'esthétique de ce produit. L'esthétique est d'ailleurs elle-même un indicateur de la qualité. Mais elle est difficilement formalisable et chacun de nous a vu des logiciels beaux, agréables à utiliser, et d'autres qui ne le sont pas. Le plaisir de l'utilisateur (et la productivité qui en résulte) n'est pas qu'une question de look. Il dépend aussi de la fonctionnalité, de la fiabilité, de l'architecture, et de la cohérence de l'ensemble de ces caractéristiques.

“

L'esthétique est un indicateur de la qualité

”

Les logiciels actuels ne souffrent pas tant d'insuffisances que d'excès : trop de fonctions, trop de fenêtres, trop de boutons, de clics, de cases à cocher. Il en résulte des produits lourds, coûteux, peu maintenables et stressants pour l'utilisateur. Une des actions du designer est de simplifier, d'alléger, de limiter à l'utile. Il devrait être le garant d'un logiciel à la fois beau, efficace et économique. En d'autres termes, d'un logiciel élégant.

### Quel avenir pour le designer ?

Verra-t-on un jour le métier de designer (quel que soit le terme employé) enseigné à l'université ? Espérons-le ! En attendant, on peut déjà esquisser quelques idées pour améliorer la

qualité globale des logiciels :

- les consultants intervenant en assistance à maîtrise d'ouvrage devraient lire quelques bons ouvrages sur l'ergonomie du logiciel, et les appliquer chaque fois que nécessaire ;
- les maîtrises d'ouvrages des entreprises et administrations qui font appel à un consultant devraient s'assurer que ce dernier n'est ni un pur technicien, ni un pur expert métier, mais qu'il a le souci de la qualité globale du produit (performances, qualité formelle et qualité perçue) ;
- les écoles d'informatique devraient inclure l'enseignement des notions élémentaires d'ergonomie du logiciel parmi les matières obligatoires.

Yves CONSTANTINIDIS  
yconstan@easynet.fr

Revue d'auteurs, L'Informatique Professionnelle accueille des opinions qui n'engagent pas la rédaction.

#### ↳ Bibliographie

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

#### ↳ VOS SITES WEB NOUS INTERESSENT !

Vos sites Web préférés nous intéressent.

Merci de nous communiquer leur URL avec quelques lignes de présentation à l'adresse :

[jma.com@free.fr](mailto:jma.com@free.fr)