

L'approche par le produit : améliorer en douceur

Analyse du produit et amélioration du processus logiciel

Yves Constantinidis, consultant

Analyser un logiciel, par exemple dans le cadre d'un audit ou d'un diagnostic d'application, est une opération utile. Elle permet de déterminer les points forts et les points faibles du produit et de définir les axes d'amélioration. Mais cette opération peut devenir doublement profitable, car l'analyse du logiciel fournit des indicateurs précieux sur son processus de développement, qui pourra alors être remanié ponctuellement, à moindre coût. D'une pierre, deux coups !... à condition de savoir établir et interpréter le « profil produit » et traduire ces indicateurs en recommandations pratiques, à même d'améliorer le processus de développement.

Processus et produit

Dans l'industrie manufacturière, les mesures de la performance et le contrôle de la qualité se font traditionnellement sur le produit, au détriment du processus qui permet de l'obtenir¹. Cela est dû, en particulier, au fait qu'il est plus facile d'examiner et d'analyser un produit manufacturé que son processus de conception et de fabrication. Or, l'examen du produit et celui du processus sont d'une égale importance, et dès le début des années 1990, les conseils en stratégie préconisaient d'examiner et de mesurer le processus.

Mais un produit logiciel n'est pas un produit manufacturé. C'est un objet immatériel qui ne connaît ni contrainte mécanique ni usure au sens courant du terme. Il est à la fois impalpable et multiforme, invisible et omniprésent. Pour cette raison, il est très difficile de matérialiser, de visualiser le logiciel. Bien entendu, des méthodes de description d'un produit logiciel existent et sont couramment utilisées, mais elles s'adressent à un aspect du produit, et non au logiciel dans son ensemble : par exemple, un modèle de données ne tient pas compte de l'ergonomie, ni des contraintes de fiabilité ; un modèle de traitements est purement fonctionnel et ne tient que rarement compte de contraintes telles que la sécurité ou les performances (temps de traitement et encombrement de la mémoire).

Du fait de l'immatérialité du produit logiciel, les contraintes s'inversent : contrairement à ce qui se passe dans l'industrie manufacturière, il est plus facile (du moins en apparence) de décrire et d'améliorer le processus de développement du logiciel que d'agir sur le produit. En conséquence, les organisations de développement (qu'il s'agisse des DSI ou d'éditeurs indépendants) essaient d'améliorer le processus dans la perspective d'améliorer le logiciel produit, mais font rarement la démarche inverse, traditionnellement pratiquée dans l'industrie,

¹ Voir à ce sujet, les travaux des cabinets de conseil en stratégie. Les éléments cités, concernant l'industrie manufacturière, sont tirés de l'ouvrage *Perspectives on strategy* du Boston Consulting Group, 1998, qui reprend un article de 1991.

qui consiste à examiner le produit pour mieux comprendre, puis améliorer, le processus. Ils se privent de ce fait d'un moyen d'amélioration simple et efficace.

L'approche par le produit

Nous proposons ici une nouvelle approche : l'approche par le produit. Elle est axée sur un ensemble de techniques reposant sur trois constats :

- le meilleur moyen d'améliorer un logiciel donné à un coût minimal consiste à examiner le logiciel lui-même, et à agir directement sur ses caractéristiques internes et externes, en n'intervenant sur le processus de développement que de manière ponctuelle, pour ainsi dire « chirurgicale » ;
- l'examen d'un logiciel fournit des informations précieuses sur le processus de son développement ; ce processus pouvant alors être ajusté et localement optimisé,
- les informations fournies par l'analyse de plusieurs logiciels à l'intérieur d'une même organisation peuvent être capitalisées, pour contribuer efficacement à l'analyse et à l'amélioration des processus de cette organisation.

L'approche par les processus et l'approche par le produit ne sont donc pas antagonistes, mais complémentaires, et s'enrichissent mutuellement.

Dans la pratique

Un diagnostic ou un audit d'application établit un rapport qui évalue les caractéristiques fonctionnelles et non fonctionnelles du logiciel examiné. Ces caractéristiques sont quantifiées, les attributs correspondant étant classés selon la norme ISO/CEI 9126, qui permet de définir un logiciel selon six caractéristiques de base, à savoir :

- la capacité fonctionnelle (functionality) : existence d'un ensemble de fonctions et de leurs propriétés données ;
- la fiabilité (reliability) : aptitude du logiciel à maintenir son niveau de service ;

- la facilité d'utilisation (usability) : relative à l'effort nécessaire pour l'utilisation du logiciel ;
- le rendement (efficiency) : rapport entre le niveau de service d'un logiciel et la quantité de ressources utilisées ;
- la maintenabilité (maintainability) : relative à l'effort nécessaire modifier le logiciel ;
- la portabilité (portability) : aptitude du logiciel à être transféré d'un environnement à un autre.

Ces six caractéristiques se décomposent en *sous-caractéristiques* permettant de décrire finement la qualité du logiciel. On peut de cette manière visualiser la qualité du logiciel et la représenter sous forme graphique (diagramme en barres, diagramme de Kiviat, etc.). On obtient ainsi un profil, ou un faciès du logiciel, qui servira de base à la discussion et à l'action.

Amélioration du logiciel

Il est alors possible d'intervenir sur les points faibles du produit, de manière ponctuelle, sans remettre en cause (en tout cas dans un premier temps) la totalité du processus de développement. Cette action « chirurgicale » pourra se faire à un coût bien moindre que celui d'une remise à plat du processus.

Notons qu'un diagnostic d'application logicielle peut intervenir, non seulement sur un produit livré, mais également sur les produits intermédiaires (cahier des charges, spécifications, maquette, prototype...).

Ajustement du processus

D'autre part, l'analyse d'un logiciel, lors d'un audit d'application par exemple, est une bonne occasion d'obtenir des informations sur son processus de développement et d'apporter des recommandations d'amélioration. Les caractéristiques de qualité du produit sont en effet des indicateurs de l'efficacité du processus de développement. Au vu du profil du produit, un expert pourra rapidement montrer du doigt les améliorations à apporter au processus.

Cette prise de connaissance des points faibles du processus par examen du produit a un coût minime, car elle utilise les informations déjà recensées lors de l'analyse du logiciel.

Voici un exemple simple : la maturité (sous-caractéristique de la fiabilité, portant sur la fréquence des défaillances) du logiciel analysé est bonne, mais dont sa stabilité (sous-caractéristique de la maintenabilité, portant sur les effets inattendus des modifications) est faible. On peut en déduire (sans certitude, mais avec une forte présomption) que la faiblesse du processus de développement se situe au niveau de la gestion de configuration du logiciel (GCL). Le croisement de toutes les sous-caractéristiques permettra ensuite de confirmer ou infirmer cette hypothèse. Et la connaissance des autres caractéristiques donnera une idée claire des améliorations à apporter au processus de développement.

Capitalisation sur l'acquis

Enfin, les connaissances acquises lors de l'examen de plusieurs logiciels d'une même organisation peuvent être cumulées et capitalisées. Alors que l'examen d'une seule application donne des indications plutôt relatives à la *mise en œuvre* des processus, les résultats de plusieurs diagnostics d'applications, à différents stades de leur développement, donneront des indications sur la *définition* même des processus. C'est alors que l'on pourra décider, en toute connaissance des impacts sur les produits livrés, si les processus devront être entièrement revus et redéfinis.

L'approche par le produit contribue donc à une amélioration sensible, rapide, et en douceur, de la qualité du processus de développement et, par conséquent, de la maîtrise du système d'information. ▲

yconstan@easynet.fr

Bibliographie

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