

Le Monde INFORMATIQUE

N° 1070 - 13/05/2005

Pierre Pezziardi
Directeur Technique
OCTO Technology

TRIBUNE

Par Pierre Pezziardi,
cofondateur et directeur
technique d'Octo Technology,
cabinet de conseil
en architecture de systèmes
d'information. A ce titre,
il dirige la recherche
et l'ensemble des publications
d'Octo (dont un récent livre
blanc sur la gouvernance
des données). Agé de 33 ans,
il est diplômé de l'Ecole
centrale de Lyon.



MARC GUILLEMINOT

Codeur, pas maçon

Architecture, urbanisme, ouvriers du code... La métaphore du BTP dans le monde informatique n'a produit que des clichés fallacieux ou propres à délocaliser des populations entières d'"ouvriers". Or, on ne construit pas des logiciels comme on construit des maisons. Dans le BTP, tous les acteurs d'un projet sont unis par leur compréhension uniforme de la finalité de ce qu'ils bâtissent : vivre au chaud, optionnellement à la lumière ou protégé du bruit.

Dans le logiciel, ce cœur n'est pas forcément partagé, et bouge énormément. Vous imaginiez des ouvriers travaillant sur des plans univoques ? Observez plutôt des écrivains guidés par le scénario général d'un metteur en scène (l'architecte). Vous pensiez rentrer dans votre appartement au départ des peintres ? Retournez plutôt le sablier de la construction et patientez pendant qu'on teste votre duplex pour une durée équivalente.

Mais le mythe du cahier des charges à la vie dure, l'écrit semble tellement "vrai" et univoque. Or, il n'a ni l'une ni l'autre de ces propriétés, les fiascos récurrents de notre industrie nous le rappellent. Si la conception générale est un art complexe qu'il convient d'aborder de manière itérative, la conception détaillée est clairement un art vain. Car seul ce qui permet de minimiser l'effort futur de modification du code produit de la valeur. Les méthodes inspirées de CMM (1) alimentent une bureaucratie en documents à la valeur ajoutée très volatile. Une batterie de tests opérationnels et du code commenté constituent une assurance bien supérieure à celle que fournit le papier.

La surface de nos logiciels ne cesse de croître, et les fonctionnalités qu'ils proposent s'imbriquent copieusement. Et c'est bien ainsi, un logiciel qui ne verrait pas toutes ses parties fortement couplées à un cœur (2) ne

serait qu'un tas de code informe, donc faiblement maintenable. Ces fonctionnalités doivent être retestées, avec un contexte toujours plus riche. Sur ces phases aval, nous avons peu investi. Je ne parle pas de logiciel miracle, mais d'investissement réel de toutes les forces opérationnelles, c'est-à-dire de développement piloté par les tests. Du code réutilisable pour tester le code représente le principal gisement de productivité dans la maintenance, qui se matérialisera par la réduction sensible des phases d'homologation, synonyme de capacité de production en cycles courts.

Hier, l'informaticien était majoritairement "monovalent", tout à son affaire informatique et sa complexité. Aujourd'hui, les bons ingénieurs sont largement bivalents, ouverts au métier : ce sont les effets naturels de nos gains de productivité. Or, les organisations n'ont pas évolué. Elles ont même amplifié les cloisonnements et la bureaucratie autour des logiciels. Ces opérationnels accomplis sont malheureux et improductifs dans des structures où la ligne de spécification précède la ligne de code. La noblesse des métiers opérationnels de l'informatique n'a pas inspiré les gestionnaires du SI, embués par une vision héritée du BTP. Ils les imaginent encore à l'usine... donc déjà très loin.

1) Capability Maturity Model : certification du processus logiciel très orienté ISO privilégiant la traçabilité par la documentation externe.

2) Le cœur constitue la structure la plus reproduite dans le logiciel, par exemple une grappe d'objets fondatrice, une implantation systématique de traitements (sécurité, collecte d'informations, validation, trace, diffusion).

Réagissez à cette tribune
lmitotechnologies@ldg.fr