

Jean-Raymond Abrial, consultant indépendant et inventeur de la méthode formelle B

« Le zéro défaut n'existe pas mais on peut tout de même s'en approcher »

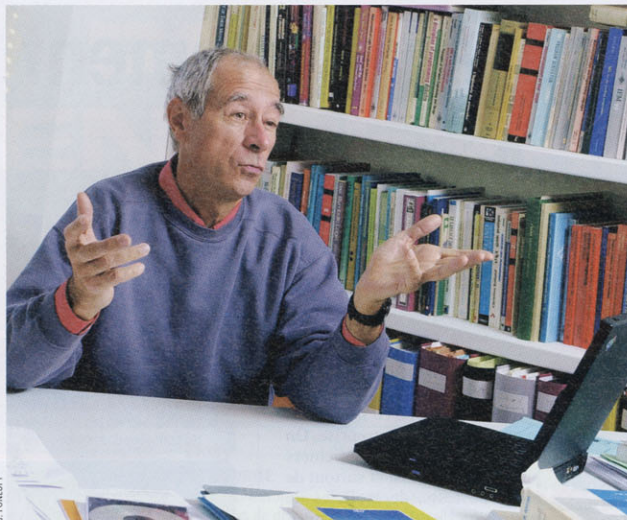
Les méthodes formelles sont en mesure de garantir des logiciels plus fiables. Pourtant, elles ont encore du mal à percer.

Bien que permettant d'accroître la fiabilité des logiciels, les méthodes formelles sont encore peu utilisées. Comment l'expliquez-vous ?

Il est possible que certains éditeurs n'aient pas intérêt à ce que leurs logiciels fonctionnent correctement. Je pense, en effet, que c'est ce qui se passe pour certains logiciels grand public. Entre autres, parce que beaucoup d'utilisateurs aiment bien bricoler. Et quand quelque chose marche bien, cela n'est pas très intéressant pour eux. Les dysfonctionnements d'un logiciel peuvent donc être voulus, car considérés comme un plus pour le produit. Mais je n'évoque ici que le bon côté des choses. Je ne parle pas des professionnels qui le font exprès pour que le client paye ensuite la maintenance. Utiliser les méthodes formelles avec cette idée en tête ne présente donc aucun intérêt. Car le logiciel tournerait correctement. Par contre, s'en servir pour garantir qu'un train sans conducteur, ou un satellite, remplit sa mission peut avoir un sens. Les enjeux ne sont ici, à l'évidence, plus du tout les mêmes.

Les méthodes formelles impliquent de s'attarder sur les spécifications au détriment des tests. C'est une remise en cause du cycle de développement...

Dans d'autres disciplines de l'ingénieur, comme la construction aéronautique ou le génie civil, il existe l'équivalent des méthodes formelles. C'est la notion de bureau d'étude. On y réalise un dessin – le modèle du futur objet – sur lequel on raisonne selon des théories. Ce sont les techniques propres à l'ingénieur. L'idée de raisonner longtemps avant sa finalisation fait partie de l'ingénierie moderne. Utiliser les méthodes formelles revient à faire du bureau d'étude pour les logiciels: le dessin d'un programme est un objet formel que l'on ne peut pas exécuter, mais sur lequel on peut raisonner. La plupart des programmeurs prétendent qu'il suffit d'exécuter un logiciel et de faire des tests pour s'assurer qu'il fonctionne. Comme si, pour faire un avion, vous fabriquiez quelque chose qui ressemble à un fer à repasser avec des ailes, et que vous le



G. FOUQUIER

testiez. Cela s'est pratiqué dans l'histoire des technologies, mais c'est une approche très primitive. Quand une technologie prend de l'ampleur, on se met à raisonner sur un dessin ou un modèle.

Quelles difficultés présente l'utilisation des méthodes formelles ?

Il est difficile de vendre une nouvelle technologie à des personnes qui n'en ont pas besoin. Un industriel qui n'a pas rencontré de gros problèmes au cours de la mise en œuvre de son processus de développement est généralement peu enclin à le remettre en question. Les méthodes formelles ne se vendent bien qu'à des personnes qui ont souffert. Leur intégration dans une pratique industrielle présente deux grosses difficultés. Elles nécessitent de revoir le processus de développement pour y intégrer les éléments caractéristiques de l'approche formelle (affinement et preuve) et elles renversent le profil de financement des projets: au lieu d'avoir des dépenses qui culminent au moment des tests, un investissement non négligeable a lieu beaucoup plus en amont, dès les spécifications et la conception, voire dès l'étude système. Pour un décideur, c'est un choix difficile à prendre. Par contre, la maîtrise technique d'une méthode formelle ne pose pas de difficultés particulières.

Dans quels types de développements logiciels les méthodes formelles prennent-elles le plus facilement ?

Il y en a beaucoup. Par exemple, pour la conception des systèmes distribués. La complexité de ces derniers réside dans le fait que la notion de maître n'existe pas. Dans un protocole de routage, un nœud du réseau ne prend pas de décision pour les autres. L'information progresse et, à chaque nœud, redétermine son chemin en fonction de la configuration locale du réseau. Il devient alors extrêmement difficile de faire des tests puisque les conditions d'exécution ne sont jamais répétitives. De fait, les méthodes formelles sont déjà en partie utilisées pour l'élaboration de protocoles de transmission, de routage ou cryptographiques. Elles sont également de plus en plus appliquées dans un domaine plus vaste que le développement logiciel proprement dit. En l'occurrence, pour l'étude des systèmes complexes. Là aussi, il faut fabriquer un objet dont il est, à l'évidence, nécessaire de pratiquer des raisonnements très rigoureux sur le modèle.

Quels sont les pays les plus avancés dans l'utilisation des méthodes formelles ?

L'Europe est plutôt en avance dans le domaine. En particulier la France, l'Angleterre et l'Allema-

► « Certains éditeurs n'ont pas intérêt à ce que leurs logiciels fonctionnent correctement »

► « Raisonner sur un modèle coûte beaucoup moins cher »

► « L'Europe est plutôt en avance dans l'utilisation des méthodes formelles »

gne. Les Américains, quant à eux, ont beaucoup d'argent et de bonnes méthodes de travail. Les gens sont extrêmement responsables dans les projets. Tout cela compense en partie. Ce sont des technologues. Sur le plan de la sociologie professionnelle, ils accordent beaucoup d'importance à la valeur du technicien. D'ailleurs, dans les grandes compagnies, il est fréquent que de très hauts responsables soient encore des techniciens. Il existe une vraie reconnaissance de la valeur technique des individus, considérée comme quelque chose d'irremplaçable. Cela leur permet de construire des systèmes remarquables.

Les méthodes formelles sont-elles vraiment la solution miracle ?

Le zéro défaut n'existe pas. Il peut toujours se passer quelque chose. Si on place un ordinateur dans un tunnel avec plein de bruits électroniques, la valeur d'un bit peut changer spontanément. Ceux qui prétendent pouvoir atteindre le zéro défaut ne sont pas de vrais ingénieurs. Mais, on peut s'en approcher toujours plus. Un ingénieur raisonne constamment en termes de probabilité. Et c'est en connaissance de cause qu'il accepte le risque, aussi faible soit-il.

Propos recueillis par Jean-Marie Portal

LE CONTEXTE

De nos jours, il est devenu courant, de parler de qualité logicielle, en entreprise. Paradoxalement, l'utilisation des méthodes formelles, qui garantissent une plus grande fiabilité des programmes, a du mal à se généraliser. Jean-Raymond Abrial, l'inventeur de la méthode B, donne son point de vue sur la situation.

NOTRE ANALYSE

« Les méthodes formelles si nécessaire, mais pas nécessairement. » C'est, en gros, le raisonnement tenu actuellement par les entreprises. Car cette technologie, lorsqu'elle est connue, dérange a priori quelque peu. Remettant en cause les processus habituels de développement, elle tend aussi à reléguer au placard les programmeurs. En somme, une petite révolution. Si les utilisateurs continuent à se contenter des produits qui leur sont délivrés, la situation mettra du temps à évoluer.