

## Introduction au génie logiciel

La baisse continue des prix du matériel informatique causée par les innovations technologiques, rend chaque jour plus important le rôle joué par l'informatique dans les sociétés industrialisées. Tous les domaines de l'activité humaine sont touchés par la progression de l'informatique.

Un système informatique ne se réduit pourtant pas à un ensemble matériel (hardware) mais il est également composé du logiciel correspondant (software) dont le coût n'a pas suivi la baisse de prix du matériel. On compte qu'en moyenne, la part de budget alloué aux logiciels dans un système informatique dépasse les 80%.

Le coût de développement d'un logiciel n'est, en effet, pas proportionnel à sa taille car les problèmes rencontrés lors de la mise au point de gros logiciels n'ont rien de comparable avec le développement de petits programmes. Un parallèle peut être dressé avec la comparaison entre la construction d'un pont devant surplomber un estuaire et celle d'une passerelle au-dessus d'un ruisseau: il ne viendrait pas à l'esprit de l'ingénieur en charge du premier problème de reprendre les plans tracés pour la passerelle en se contentant d'en agrandir les dimensions.

La complexité des petits programmes est telle qu'une seule personne peut la maîtriser et traiter toute seule le problème de bout en bout. De plus, elle est à même de mémoriser toutes les décisions prises lors de la conception et donc d'apporter par la suite toute modification au logiciel terminé.

Il en va tout autrement en ce qui concerne les gros logiciels dont la complexité est telle qu'il n'est plus possible à une seule personne d'en assurer le développement ni d'en connaître toutes les finesses. Le développement devra donc être organisé et coordonné; les personnes en charge du développement devront documenter leur travail de manière exhaustive et standardisée, et les différentes tâches du développement devront être soigneusement synchronisées. En un mot: le projet doit être bien géré.

Une discipline appelée génie logiciel (software engineering) s'occupe de cette activité de la science et de l'industrie informatique. En gros, le génie logiciel propose aux informaticiens de nouvelles techniques et de nouvelles méthodologies spécialement destinées au développement de gros logiciels.

Si le terme de génie logiciel date du début des années '70, cette branche de l'informatique suscite ces dernières années un intérêt croissant. L'explication est avant tout économique: un logiciel bien conçu et dont le développement a été rationnellement planifié coûte indéniablement moins cher. Pour s'en convaincre, voyons comment se répartissent les coûts dans le développement d'un gros logiciel.

Le développement d'une grosse application informatique peut se décomposer en 4 phases:

- l'analyse des besoins (requirements analysis): les besoins, buts et contraintes du clients sont répertoriés et exprimés dans un langage compréhensible tant par l'informaticien que par le client;

- le choix d'une solution (system design): sur base des besoins, on structure le problème en sous-problèmes et on associe à chaque sous-problème une solution informatique pour le résoudre;
- 
- la réalisation pratique (implementation): on concrétise les solutions choisies lors de la phase précédente en produisant un ensemble de programmes et en testant chacun de ceux-ci individuellement;
- 
- intégration et test (system testing): on intègre les différents modules réalisés lors de la phase précédente et on teste l'ensemble ainsi formé.

Ces 4 phases se suivent, ou plutôt s'enchaînent dans cet ordre. Cependant, certains "retours en arrière" sont nécessaires : lorsque les tests de la phase d'intégration mettent en évidence une erreur, il faut retourner à une étape précédente. Plus l'étape à laquelle il faut retourner est éloignée, plus la correction coûtera cher.

Le coût relatif des différentes phases de la réalisation d'un gros logiciel varie d'une application à l'autre. En moyenne, pour une application de type scientifique ou commercial, les coûts se répartissent comme suit:

|   |     |
|---|-----|
| Analyse des besoins et choix d'une solution ..... | 45% |
| Réalisation pratique.....                         | 25% |
| Intégration et tests .....                        | 30% |

L'ensemble des 4 phases présentées ci-dessus ne représentent cependant qu'entre 1/3 et 1/5 du coût total d'un gros logiciel ! En effet, lorsqu'une erreur passe entre les mailles du filet, c'est-à-dire lorsqu'elle n'est pas détectée lors des tests, elle apparaît chez le client lors de l'utilisation du logiciel et sa correction est extrêmement lourde surtout si les programmes ne sont pas suffisamment documentés. Une mauvaise documentation rendra également fastidieuse toute modification demandée par l'utilisateur. Tous ces travaux effectués au logiciel après sa livraison chez le client forme ce que l'on appelle la maintenance.

Si l'on veut réduire le coût d'un gros logiciel nous voyons donc qu'un gestionnaire de projet a tout intérêt à accorder un maximum d'attention aux premières phases de la réalisation d'un gros logiciel afin de diminuer les frais de maintenance :

- une bonne analyse des besoins peut éviter que le client ne demande des modifications après la réalisation du logiciel;
- un choix de solution judicieux évitera l'apparition d'erreurs au moment de l'intégration des différents modules;
- une bonne réalisation pratique diminuera le risque d'erreur dans le programme;
- des tests efficaces limiteront au maximum le nombre d'erreurs dans le logiciel livré au client.

Ces mesures visant à une réduction des coûts peuvent sembler simples. Cependant leur mise en pratique n'est pas évidente et c'est là qu'interviennent les techniques du génie logiciel.

REFERENCE: "Software Engineering - 2nd éd.", Ian Sommerville, Addison-Wesley Publishing Company, 1985.

Philippe Dubois

Article paru dans « l'escargot », la revue de l'asbl en juillet 1989 (Escargot 89/3).