

Avis d'expert, juin 2011

Amélioration des tests: une urgence économique !

Quelques faits

Les développements logiciels coûtent de plus en plus cher. Avec une part correspondant à 30%, voire 50% du coût total de développement, les tests en sont une partie non négligeable. Malgré de nombreux efforts, la qualité des logiciels ne semble pas s'améliorer pour autant. L'actualité récente (plusieurs millions d'utilisateurs de cartes bancaires allemands pris en otage en Janvier 2010 ; le fleuron d'Airbus obligé de faire demi-tour le 20 novembre 2009 pour une panne de pilote automatique, etc.) nous le rappelle fréquemment.

Selon des études récentes [1], le niveau de qualité des logiciels livrés est encore fortement améliorable, les meilleures sociétés ne découvrant et ne corrigeant que 85% des défauts et livrant, avec le logiciel les 15% de défauts restant. Dès que l'on parle de milliers de défauts, 15% devient un nombre important.

L'augmentation sans cesse croissante de la complexité des applications et du nombre d'interactions entre les logiciels (p.ex. Cloud, SaaS, etc.) font qu'un défaut dans un autre composant peut influencer le fonctionnement de votre propre logiciel ou système. Pour vous en prémunir, il est nécessaire de faire des tests exhaustifs, ce qui est difficilement justifiable économiquement, voire impossible en termes de délais et de charge.

Dans un contexte de compétition internationale, de recherche de rentabilité et de compétitivité, il est impensable d'améliorer l'efficacité des processus de développement et de test. L'efficacité des développements logiciel a fait l'objet de nombreuses actions, (nouveaux langages, de nouveaux paradigmes et nouvelles méthodologies). Or, la qualité des logiciels ne s'est pas améliorée sensiblement, les coûts ont continué à croître, les délais à s'allonger. « Que fait la qualité, à quoi servent les testeurs ? » est une question qu'on est en droit de poser.

Professionnalisation des tests

Depuis la publication des premiers ouvrages de vulgarisation sur les tests [2] il y a plus de 30 ans, de nombreux milliards de dollars (puis d'euros) se sont envolés en fumée, reflétant frustration des utilisateurs, exaspération des testeurs et nuits blanches des développeurs. Et cela, malgré les nombreux livres de références, et outils de tests existant à ce jour.

Dans le même temps des organismes se sont penchés sur l'amélioration des processus et ont proposé des programmes d'évaluation (CMMI, SPICE, ISO9000, etc.) qui ont démontré leur intérêt économique, entre autre l'aspect d'amélioration continue des processus, seul à même de proposer une réduction des coûts à long terme.

Amélioration des processus de test

Leur efficacité des processus et techniques de test est dépendante du contexte où ils sont utilisés, et seule une amélioration continue garantit leur efficacité accrue.

L'amélioration des processus de test, pour en réduire les coûts et la durée ou d'augmenter leur efficacité est un objectif affiché par tous. Cependant, ceci ne peut se faire qu'avec un niveau de référence à partir duquel comparer les évolutions.

Plusieurs méthodes d'améliorations de processus existent, et se regroupent en deux familles : celle à modèle de référence externe où l'on compare les processus par rapport à un référentiel idéal (p.ex. TPI, TMMI, CMMI) ; ou celle à modèle de référence interne, où le référentiel est la société elle-même, et ses propre processus (p.ex. IMPROVE, PLAN, CTP, STEP, ODC).

L'amélioration des processus basé sur un référentiel externe s'appuie sur des processus établis. Il est donc simple de faire appliquer ces méthodes toutes faites. La mise en œuvre cependant peut ne pas correspondre aux attentes. Un autre point à envisager est quoi envisager comme actions d'amélioration après que le référentiel externe ait été complètement implémenté.

L'utilisation d'une méthodologie basée sur l'analyse du référentiel interne de l'entreprise est automatiquement adaptée aux besoins de celle-ci, après une évaluation (un audit) des processus de test et de développement existants, de leur rentabilité respective (en terme de coût, de génération de défauts et d'efficacité de détection). L'évaluation des processus et leur pertinence, l'identification des processus à améliorer, la sélection de ceux à améliorer en priorité, et la préconisation d'actions, nécessite une compétence pointue et une expérience étendue de la part de l'acteur. De tels niveaux de connaissance requièrent des consultants sénior expérimentés. Si leur coût journalier peut paraître élevé, les améliorations qu'ils préconisent étant adaptées aux besoins de l'entreprise, leur rentabilité à court terme est souvent fort élevée.

Evaluation de la rentabilité.

L'évaluation de la rentabilité des processus doit étudier, outre les coûts de réalisation et le nombre de défauts identifiés, le nombre de défauts qui sortent du processus sans avoir été détectés. Il faut donc prendre en compte les coûts des défauts (et tous les coûts associés à leur correction), lesquels augmentent en cas de détection tardive.

Combien de sociétés évaluent actuellement les coûts de leurs défauts selon le moment où ils sont découverts ? Pourtant ces informations sont disponibles, il suffit de les mesurer.

Combien de responsables de développement sont en mesure d'identifier l'origine d'un défaut, de l'associer à un processus de développement, et de proposer des actions d'amélioration de façon à éviter une répétition de ce défaut dans le futur ? Combien sont en mesure de fournir une évaluation statistique liant les défauts identifiés aux processus de développement qui sont à l'origine de ces défauts ?

Des axes d'amélioration

Plusieurs ouvrages de référence[3] mettent clairement en avant l'efficacité des processus de test statiques (revues, inspections, analyse de code) et le besoin d'améliorer la qualité des exigences, comme moyens d'atteindre rapidement un niveau de qualité adéquat des développements. Une étude récente[4] portant sur plus de 12000 sociétés a démontré qu'aucune technique de test individuelle ne permettait de trouver plus de 30% des défauts d'une application. Il est donc recommandé de mettre en œuvre plusieurs techniques complémentaires, pour assurer une efficacité de détection autour de 85-90%.

Choix de processus à améliorer

La sélection des processus à améliorer doit se faire selon leur rentabilité. Celle-ci s'évalue d'une part en fonction des améliorations immédiates fournies (défauts supplémentaires détectés), d'autre part en fonction des améliorations induites dans les autres processus. Exemple : un éditeur connu a réussi, par l'application d'inspections formelles (analyse statique) à trouver 50% de défauts de plus que lors des releases précédent, réduisant la durée totale des tests de 15% et la charge de test dynamique de 75%.

Synthèse

La détection des défauts et leur correction sont les activités les plus chères du cycle de développement de logiciels. Toute amélioration de ces processus se traduira directement par des réductions de coût, par des développements de qualité et une diminution du nombre de défauts détectés par les utilisateurs.

Pour les améliorer, il est important de mesurer les activités de tests, et aussi toutes les autres activités liées au cycle de vie [5] des logiciels, depuis les exigences jusqu'à la maintenance, au support hotline et au SAV, car l'amélioration des processus de test influence aussi ces autres processus. Une mesure en terme d'impact et en terme de volumétrie, aide à déterminer les processus à améliorer en priorité.

Une recherche des origines des défauts (Analyse des Causes Racine) permet d'identifier les processus défaillants à l'origine des défauts. Des méthodes d'amélioration des processus, intégrant les aspects systématiques et statistiques permettent de garantir un taux de retour sur investissement supérieur aux meilleures sociétés cotées en bourse.

Parce que les tests représentent entre 30 à 50% des coûts de développement, leur amélioration est une urgence économique justifiant un regard appuyé de la part de la direction de nos entreprises. Une amélioration itérative de la rentabilité des processus de test permettra de montrer la maturité de notre profession, notre professionnalisme et notre valeur ajoutée.

Bernard Homès

bhomes@tesscogroup.com

[1] Estimating Software Costs, Capers Jones, 2008, McGraw Hill, ISBN: 978-0-07-148300-1

[2] The Art of Software Testing, Glenford Myers, 1979, John Wiley & Sons, ISBN 0471-04328-1

[3] Software Estimation, S.McConnell, 2006, Microsoft Press, ISBN 978-0735605350

Metrics and Models in Software Quality Engineering (2nd ed.), S. Kan, 2007, Addison Wesley, ISBN 0201729156

Software Inspection, T.Gilb & D.Graham, 1993, Addison Wesley, ISBN 0201631814

Software Reliability Engineering (2nd Edition), J.D.Musa, 2004, Author House, ISBN 9781418493875

[4] Estimating Software Costs, C.Jones, 2007, Mc Graw Hill, ISBN 978-007148300-1

[5] plus vaste que le cycle de développement, car un logiciel en maintenance reste une source de coûts de test et de correction.