Les erreurs de logiciels, le plus cher feu d'artifice de tous les temps

En Europe, grâce aux logiciels de tests nous pourrions économiser plus de 100 milliards d'euros par an

de Klaus Lambertz, Verifysoft Technology GmbH

„A 300 mètres veuillez tourner" nous indique le GPS sur l'autoroute, Windows devient bleu ou lorsque le logiciel de votre téléphone se bloque quand vous prenez une photo. Les causes de ces problèmes sont le plus souvent des erreurs de logiciel.
Dans les années 1940, pour la première fois, les problèmes dans un relais d'ordinateur dérangeaient. Un insecte (en anglais bug) a provoqué un court-circuit dans un ordinateur. Depuis le développeur nomme le problème informatique de bug. Aujourd'hui, les insectes ne jouent plus un rôle dans les problèmes informatiques. Les erreurs de programmations sont nées des mains de l'homme et ne sont pas toujours sans danger. En effet les bugs peuvent paralyser les réseaux de téléphones, être la cause de l'explosion d'une fusée, ou bien coûter la vie à des être humains.

En France, au début de l'année, un bug informatique provenant de la BNP Paribas a eu des conséquences plus que fâcheuses, en effet des dizaines de milliers de comptes clients ont été débités plusieurs fois par erreur.
La fusée Ariane 5 a explosé 40s après son décollage dû à un bug informatique. En effet les ingénieurs européens avaient introduit le logiciel d'Ariane 4 dans la nouvelle fusée qui était bien différente. Ce logiciel n'était pas capable de calculer les données d'Ariane 5 étant donné sa plus forte puissance : avec environ 500 millions de dollars de dommage matériel, ça a été le feu d'artifice le plus coûteux de tous les temps.
Au cours de l'année 2000, dans le domaine de la médecine, un programme de mesure de radiation a donné des valeurs incorrectes ce qui a coûté la vie à huit patients et environ 20 personnes ont été grièvement blessées.

Tous ces cas d'erreurs logicielles ne sont malheureusement pas isolés, on peut en répertorier des milliers. Les programmes employés dans l'industrie ne répondent pas souvent aux exigences. Les conséquences de ses maladresses ont un poids énorme sur l'entreprise. En effet il y a un manque de productivité, les produits défectueux doivent être retournés ce qui peut nuire à la réputation d'une entreprise.

Les logiciels de voitures sont aussi complexes que les logiciels de la fusée Saturn V

On trouve aujourd´hui des logiciels dans tous les domaines. La complexité des programmes est mouvante, en effet elle double environ tous les 18 mois.

Avec en moyenne 50 appareils calculateurs comme l'ABS et EPS, une voiture de classe moyenne est aujourd'hui équipée avec autant de logiciel que la fusée Saturn V qui a emmené les premières personnes sur la lune dans les années 60. La valeur de la voiture est composée aujourd'hui de logiciel et d'électronique à hauteur de 40%. Cette tendance tend à évoluer.

Plus la complexité des logiciels grandit, plus le nombre d'erreurs de programmation est en hausse. Un logiciel qui a environ 2 bugs dans 1000 lignes de code est considéré comme un bon logiciel. Le programme informatique de la navette spatiale de la NASA a fait l'objet d'un examen particulier. On estime qu'il y a tout de même 300 erreurs qui subsistent dans les trois millions de lignes de codes et l'une d'entre elles peut suffire pour déclencher une catastrophe.

Selon Intel, chaque processeur Pentium a environ 80 à 90 bugs. Dans le cas d'un téléphone avec une moyenne de 200 000 lignes de code, on peut compter près de 600 erreurs.
Un programme comme Windows XP a environ 40 Millions de lignes de code ce qui représente, imprimé, sur une page A4 une pile de 60 mètres de feuilles. Selon les statistiques environ 800 000 bugs se sont glissés dans ce logiciel.

150 Milliards d'euros de dommages par an en Europe

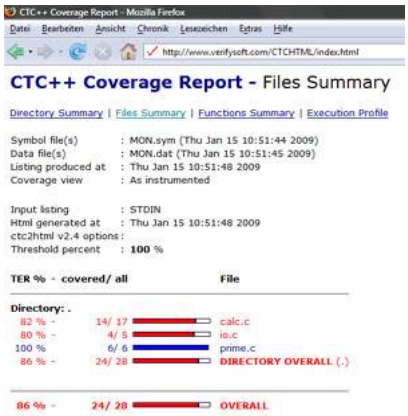
Les Hatton, professeur de l'université Kingston de Londres estime qu'en Europe les pertes liées aux erreurs de programmation coûtent jusqu'à 150 milliards d'euros.
A travers des tests corrects les dommages pourraient nettement diminuer. Il est plus facile et peu coûteux de résoudre des bugs lors de la programmation du logiciel, par contre lorsque le processus de développement a été effectué, les frais de correction des bugs sont bien plus coûteux. En effet lorsque le logiciel est fini, l'erreur coûte au moins 30 fois plus cher.

Dans le développement de logiciel dans les domaines critiques tels que l'aéronautique, l'automobile, le secteur médical, on essaie de contrôler la qualité à travers des certifications. En effet certaines normes internationales décrivent le processus de développement et le processus de test d'un logiciel.

Le « logiciel banane » est vu comme un problème

Malheureusement, dans les domaines moins critiques, le principe de „logiciel banane“ prédomine encore. Le logiciel est vendu malgré les tests insuffisants, il est pour ainsi dire encore « vert » et mûrit alors chez le client. Les entreprises pensent qu'en évitant les tests, elles feront des économies car d'après la devise « le client nous dira déjà ce qui ne va pas », donc ce n'est pas la peine de tester tant que rien n'est arrivé. Réagir comme ceci peut leur coûter très cher. Beaucoup de ces sociétés de développement peu professionnelles ont déjà disparu du marché - et cela continuera encore.

Des outils efficaces simplifient le test de logiciels



Les outils de tests, la vérification du logiciel est simplifiée et plus efficace. En effet, à travers les outils de tests on peut détecter dès le début du processus de développement des erreurs et éviter des problèmes lorsque le logiciel sera fini. Les résultats sont de meilleure qualité et les coûts sont réduits. Les tests de logiciels ne sont donc pas un facteur de coût pour l'entreprise mais sont à moyen et long terme une contribution à leur réduction.

Les outils d'analyse de la complexité du code montrent la façon dont le logiciel est programmé. Dans un logiciel certaines parties développées peuvent être compliquées, ces parties trop complexes ont potentiellement plus d'erreurs et elles sont également plus difficiles à tester. Si les parties de logiciel trop complexes doivent être réutilisées pour un autre produit, des problèmes peuvent se présenter. Les outils d'analyse de la complexité du logiciel possèdent aussi un index de maintenabilité qui nous montre quand il est préférable de réécrire la partie du logiciel trop complexe. Ils nous indiquent également le nombre minimal de cas de test qui est nécessaire pour vérifier le code. Lorsque l'on sous traite le développement de logiciel, l'outil peut servir à montrer le temps nécessaire à l'exécution pour écrire le projet.

Grâce à **l'outil de test unitaire**, il est possible de tester des parties séparément dès le début du processus de développement même si le projet n'est pas fini. En effet les parties manquantes sont simulées par l'outil. Si des erreurs sont trouvées, celles ci sont uniquement dans la partie de code qui a été testée. De cette façon la localisation de la source d'erreur est simplifiée. Chaque partie du logiciel sera alors testée, elles seront déjà « propres » avant que le logiciel ne soit installé. Aujourd'hui encore dans beaucoup de domaines, le logiciel n'est testé qu'à la fin : comme un fabricant d'automobiles qui installe toutes ses pièces détachées, par exemple le démarreur, sans test préalable du moteur et remarque seulement à l'essai que le véhicule ne fonctionne pas. Dans la fabrication de voiture c'est impensable mais malheureusement dans certaines société de développement de logiciel cela existe encore.

La fonction de **l'outil de couverture de test** (appelé aussi couverture de code) est de montrer quelles sont les parties du code qui sont déjà testées et celles qui doivent encore l'être. Il ne s'agit donc pas d'un test à proprement parlé mais d'une analyse de la qualité du test. En ce qui concerne les domaines critiques, les logiciels doivent être vérifiés à 100% et le test doit être documenté.

L'outil pour **l'analyse statique** a pour objectif d'examiner le logiciel sans que le code soit exécuté. Ainsi il n'est pas nécessaire que toutes les parties du logiciel soient faites. Avec l'analyse statique, les défauts peuvent être trouvés plus tôt dans le processus de développement. Les outils d'analyse scannent les codes du programme. L'écriture de test n'est pas nécessaire pour faire ça. Après l'analyse automatique, l'outil de test statique affiche les défauts dans le code. Il y a aussi des analyses statiques qui montrent si on a bien respecté les règles de programmation qui servent à éviter des erreurs.

Lors d'un **test fonctionnel**, on examine le logiciel avec les yeux d'un utilisateur. L'outil analyse si le logiciel se comporte conformément aux exigences du client. Ce type de test fait des analyses de « **boîtes noires** », en effet ce n'est pas le code de programmation du logiciel qui est analysé mais seulement son comportement. Depuis certaines années, les générateurs de cas de test automatiques sont disponibles pour cette sorte de test. Les exigences des utilisateurs du logiciel sont décrites dans un modèle. Le générateur de test l'analyse et génère automatiquement les vérifications nécessaires.

Klaus LAMBERTZ est co-fondateur de Verifysoft Technology http://www.verifysoft.com/fr.html basée à Offenburg en Allemagne, près de la frontière française. Cette société a pour objectif de fournir des solutions et des conseils dans le domaine des tests et d'analyse des logiciels.
Avant de créer Verifysoft Technology il a occupé les postes d'ingénieur commercial grands comptes et de manager de l'équipe des ventes pour différentes sociétés dans le domaine des tests de logiciels en France et en Allemagne. *Contact: lambertz@verifysoft.com*

Verifysoft Technology GmbH
Technologiepark
In der Spöck 10
77656 Offenburg (Allemagne)
Tel. +49 781 63 92 027
Tel. France 03 68 33 58 84
www.verifysoft.com