
SonarJ est un outil de développement pour surveiller et améliorer efficacement l'architecture et la qualité technique des projets Java de taille moyenne à grande. Au salon 'Systems' 2005, il a reçu le prix de « l'innovation la plus intéressante ».

Quels problèmes SonarJ adresse-t-il?

Durant presque tous les projets logiciels majeurs, un écart croissant se crée au cours du développement entre la structure (architecture) qui était initialement prévue et celle réalisée dans le code source. Bien que cet écart soit indésirable, il est extrêmement difficile à éviter. Il est causé par l'introduction de dépendances non voulues entre les différents éléments du logiciel. Le résultat : comprendre et tester le système devient de plus en plus difficile. Pourquoi ces dépendances non voulues sont-elles si difficiles à éviter ? La raison en est qu'elles ne sont pas normalement identifiées dans leurs premières étapes, et qu'elles n'exercent leurs effets désastreux que longtemps après – mais d'une façon très préjudiciable. Le terme utilisé est « érosion de l'architecture », et ses symptômes sont trop bien connus de la plupart des responsables informatiques : des retards se produisent pour finir le logiciel, et son coût dépasse de beaucoup le budget initial. Bien pire, la qualité n'est malgré tout pas au niveau du standard désiré, et chaque tentative pour étendre le logiciel est si chère que les développeurs commencent à suggérer qu'un « redesign » serait

préférable. Autrement dit, le logiciel est jeté au rebut et ré-écrit simplement parce qu'il s'avère que c'est moins cher que d'éliminer les défauts de la base existante – un vrai cauchemar pour celui qui est responsable des coûts encourus.

Un autre domaine de problème est de contrôler la qualité de logiciel de fournisseurs extérieurs ou de réaliser des vérifications régulières de la qualité technique d'un projet interne. Lorsque ce travail est demandé, il est souvent limité aux fonctions et aux besoins techniques, alors que la qualité technique interne est ignorée. Dans de nombreux cas, le logiciel qui est livré satisfait les formalités fonctionnelles et techniques, mais il s'avère très difficile à étendre ou à modifier, à cause des défauts de sa structure interne. Le client finit par payer le prix fort à cause d'eux, et par conséquent a fondamentalement intérêt à définir des critères de qualité technique vérifiables pour le logiciel fourni. C'était la difficulté jusqu'à présent : établir des critères mesurables pour la qualité technique et la structure interne.

Comment SonarJ a-t-il résolu ces problèmes ?

SonarJ permet aux architectes logiciels de définir la structure du logiciel qu'ils conçoivent ou qu'ils étendent avec un très haut niveau d'abstraction, et de continuer à le développer à ce niveau. Comme l'abstraction est de très haut niveau, la structure devient facile à comprendre et à communiquer. Cette définition de structure est distribuée à l'équipe de développement comme un fichier XML. SonarJ est installé sur chaque poste de développement, soit comme un plug-in Eclipse, soit comme un produit indépendant. SonarJ lit la définition d'architecture, et à partir de ce moment, vérifie que chaque changement qui est fait dans le code est conforme

avec l'architecture prévue. Le développeur, en fait, ne se rend pas compte du processus de vérification parce qu'il se déroule comme une tâche de fond avec un très haut niveau d'efficacité. Pourtant, si l'architecture spécifiée est violée, un avertissement est affiché immédiatement pour aider le développeur à éliminer le problème sur le champ. Le produit offre aussi un nombre considérable d'aides au développeur; elles rendent le logiciel et l'architecture prévue plus faciles à comprendre, familiarise plus vite les nouveaux employés avec les procédures nécessaires et améliore la productivité globale.

SonarJ est unique : il offre la combinaison de la définition d'architecture au plus haut niveau et de sa vérification directe durant le processus de développement est unique sur le marché.

Une application supplémentaire du produit est le contrôle de la qualité technique d'un logiciel. SonarJ possède de puissantes capacités d'analyse statique de code, et peut donc rendre les structures internes visibles. En évaluant certaines métriques de base calculées par SonarJ, un verdict initial sur la qualité technique d'un logiciel à auditer peut être obtenu en un temps très court (moins de 2 heures).

Le client peut désormais inclure des critères de qualité techniques vérifiables dans ses appels d'offres et dans les spécifications de ses commandes, avec comme résultat que ses fournisseurs ne pourront plus compter sur l'acceptation de systèmes logiciels avec des fonctionnalités correctes, mais défectueux quant à leur structure.

Quels sont les effets de SonarJ sur les coûts ?

Le "Software Technology Support Center" (STSC), qui a des liens étroits avec le Pentagone aux États-Unis, a étudié dans quelle mesure une structure interne, selon qu'elle est satisfaisante ou défectueuse, influence le coût de développements logiciels supplémentaires. Les résultats confirment clairement ce qui a toujours été pressenti comme vrai : les erreurs de structures s'avèrent très coûteuses.

Un système de 50 000 lignes code a été étudié; le but était de l'étendre en y ajoutant environ 3 000 lignes de code. Deux équipes sont formées : l'équipe 1 est chargée de travailler sur la base de la structure du système « à maturité », qui n'est en aucun cas

optimal, puisqu'il montre déjà les symptômes typiques d'une architecture érodée. L'équipe 2 reçoit comme base un système restructuré, après élimination des défauts de structure précédents.

Sans surprise, l'équipe 1 a eu besoin de deux fois plus de temps que l'équipe 2 pour finir sa tâche.

Encore plus alarmant : le nombre d'erreurs trouvées dans la livraison de l'équipe 1 est plus de huit fois plus grand que de celui de l'équipe 2.

Aucune autre étude de ce type n'a été publiée depuis, mais la plupart des experts dans le secteur du développement logiciel s'accorderont sûrement pour considérer que ce résultat n'est pas surprenant.

Êtes-vous intéressé ?

Nous vous proposons de vous démontrer notre technologie en l'appliquant à l'un de vos propres projets. Pour plus d'informations, des documents techniques et la possibilité d'évaluer gratuitement le produit, rendez-vous sur notre site internet: www.pcmetric.com

« J'ai été étonné de voir à quel point il a été facile et rapide pour nous d'adopter SonarJ pour gérer l'architecture et la qualité technique de la famille de logiciel du Framework Spring. Pour le projet Spring Web Flow, SonarJ nous a aidé à réduire de moitié le couplage interne de la base de code avec un très petit effort additionnel. Maintenant, nous utilisons SonarJ dans notre travail de développement quotidien et il nous aide à maintenir l'architecture et la qualité de Spring sur le haut niveau attendu par notre base globale d'utilisateurs. »

***Jurgen Hoeller – VP & Distinguished Engineer, Interface 21
(Springsource)***