

Développement : les erreurs du Build analysées en profondeur

La phase Build de développement d'une application ou d'un logiciel est importante. Elle se définit comme un processus d'assemblage qui comprend les linker, les scripts et la compilation des programmes en code machine exécutable. Or, il arrive que pendant cette période **des erreurs de compilation** surviennent et fassent perdre du temps aux programmeurs. Des chercheurs de Google et des scientifiques de l'Université des Sciences et des Technologies de Hong Kong et de celle du Nebraska se sont penchés, selon nos confrères d'*IT Pro*, sur ce sujet pour tenter d'améliorer les outils de développement, réduire les erreurs et augmenter la productivité du développeur.

De manière empirique, ils ont analysé **26,6 millions de Build** élaborés par **18 000 ingénieurs de Google** entre novembre 2012 et juillet 2013. Ces programmes ont été écrits en **Java ou C++**, les langages les plus couramment utilisés par la firme américaine. Les erreurs ont été générées soit par **le compilateur javac (pour Java) ou LLVM Clang (pour C++)**. Ils ont regroupé les messages d'erreurs des compilateurs en 5 catégories : dépendance, problème de type, syntaxe, sémantique et autres.

Pas de sensibilité à la fréquence et à l'expérience des développeurs

Les résultats sont parfois surprenants et pourfendent quelques idées reçues. Ainsi, expérience et assiduité dans le développement ne riment pas nécessairement avec un meilleur Build. Dans leur analyse, les chercheurs sont partis de différentes hypothèses. L'une d'entre elles était que **les développeurs qui créent fréquemment du code génèrent plus de taux d'échec**. Mais aucune corrélation n'a été trouvée pour justifier cette hypothèse. Autre postulat démonté, les programmeurs expérimentés auraient un faible taux d'erreur. L'analyse montre qu'il n'y a aucune relation de cause à effet.

Une dépendance génératrice d'erreurs

Pour les chercheurs, la majorité des erreurs de compilations sont liées **au facteur dépendance**. Ainsi 65% de l'ensemble des erreurs Java sont issues de ce facteur. La traduction la plus courante (43%) de ces erreurs ? L'absence de symbole. Pour C++, ce taux lié à la dépendance est plus modéré (53%), mais encore élevé avec comme souci récurrent, **les identificateurs non déclarés et les variables de classe manquantes**.

Si **C++** s'en sort un peu mieux sur la dépendance, il génère par contre **plus d'erreurs** (38,4%) que Java (28,5%). Ce décalage se retrouve aussi dans les erreurs de syntaxe, même si les chercheurs imputent cette différence à l'usage des IDE dans le développement Java, ce qui réduit ainsi les erreurs simples. Par contre, bon point pour C++ chez qui les erreurs sont plus faciles à corriger.

Ce travail, comme l'expliquent ses promoteurs, est empirique, mais a le mérite de donner des éléments de comparaison pour de futures études similaires et un peu plus poussées. Il donne aussi des pistes pour améliorer et optimiser le travail du développeur.

A lire aussi :

[Les développeurs victimes des compilateurs effaceurs de code](#)

[Salaire des développeurs : maîtriser Ruby ou Python paie !](#)