



Productivité et tests automatisés sont-ils compatibles ?

L'automatisation des tests engendre nécessairement un coût sur un projet, et se trouve de ce fait susceptible de ralentir le développeur. Un tel examen superficiel conduit à la conclusion que l'automatisation des tests a tendance à diminuer la productivité.



■ Dominique Méra - *Objet Direct - Consultant - Directeur de projets*
 ■ Illustrations par Ludivine Alligier

Ce raisonnement ne tient pas compte du gain obtenu au moment du passage des tests, puisqu'un test automatique ne coûte presque rien à passer. La question fondamentale est donc de déterminer si le gain compense l'investissement initial. Sur ce point les avis sont partagés.

A priori, un test donné a de bonnes

chances d'être exécuté un nombre de fois non négligeable. Prenons l'exemple d'un écran : le développeur mettra deux heures à le mettre en place. Ensuite, il convient d'effectuer un premier test pour vérifier qu'il fonctionne. Que ce test soit manuel ou automatique, il est fort probable que le développeur identifie à travers lui une erreur dans le code écrit. Cette erreur rectifiée, le développeur effectuera de nouveau le même test. Ce cycle sera répété tant que le test ne passera pas avec succès. Le nombre de cycles est en général corrélié avec l'expérience du développeur, et varie de 1 à 10, voire plus. Ultérieurement, une seconde passe de tests pour ce même écran sera exécutée juste avant la livraison. Or la

conduit à une nouvelle livraison. En toute rigueur, l'ensemble des tests devrait être exécuté de nouveau préalablement à la re-livraison, pour garantir l'absence de régressions.

En pratique, le coût de cette nouvelle phase de tests globale étant réductible, seul un sous-ensemble de ces tests est effectivement exécuté, dans le meilleur des cas.

Il apparaît alors un risque sur la qualité du produit. Bien entendu, le cycle de tests utilisateurs / corrections / relivraison se produit en pratique plusieurs fois et est susceptible de retarder la mise en production (ou la distribution) de la version définitive de logiciel. Il n'est pas rare qu'un même écran soit testé 10 ou 20 fois dans le processus. Sans compter les tests qui seront nécessaires pour les versions successives pendant la durée de vie de l'application.

Nos recommandations

- La totalité des tests ne peut pas être automatisée. La règle des 80/20 s'applique : 80% des tests peuvent être automatisés avec 20% de l'effort. Il convient donc de ne pas automatiser les tests qui seraient onéreux à l'être.
- Construire un nombre de jeux de données restreint, de telle sorte que plusieurs tests puissent s'appuyer sur le même jeu de données (éventuellement modifié légèrement dans la phase de préparation du test pour les besoins propres de ce test).
- Privilégier les tests écrits dans un langage par rapport à des outils d'enregistrement : ces derniers outils ne peuvent être mis en œuvre que lorsque le code est suffisamment abouti, et ne sont donc pas disponibles pour les tests unitaires du développeur.
- Dans la mesure du possible, utiliser un test fonctionnel pour faire un test unitaire ou d'intégration, pour des raisons évidentes d'économie.
- Porter l'effort sur les fonctions les plus importantes du logiciel, et les plus visibles des utilisateurs. Automatiser les tests sur une fonction qui sera en pratique utilisée par une petite poignée d'utilisateurs une ou deux fois pas an est probablement peu rentable.
- Concevoir le test de manière à ce qu'il soit peu sensible aux évolutions ultérieures du code adjacent au code testé. Par exemple, dans un écran, si le test clique sur un bouton identifié à partir de son emplacement sur l'écran, la moindre modification sur cet écran, comme l'ajout d'un bouton, risque de nécessiter une mise à niveau du test qui pourtant ne concerne pas le bouton ajouté.

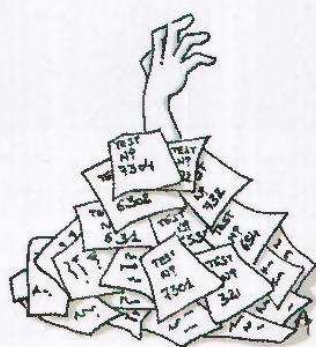


Figure 2 : Lorsque les tests ont été automatisés, la batterie complète peut être exécutée avec un effort modeste.

phase de tests conduite par les utilisateurs suite à cette livraison amène la plupart du temps à la détection d'un certain nombre de demandes de changement (bugs ou évolutions).

La prise en compte de ces demandes

Optimiser les coûts

Le coût initial de l'automatisation des tests peut très bien s'avérer inférieur au coût global des tests exécutés manuellement. Il s'agit d'une certaine manière d'un investissement : le temps passé sur l'automatisation, qui représente un coût immédiat, devra rester inférieur au temps global des tests manuels. L'évaluation de rentabilité n'est possible que pour un projet donné, en tenant compte du contexte. L'expérience montre que dans la majorité des projets, l'automatisation des tests permet de fournir un logiciel de qualité supérieure à un coût inférieur et dans un délai plus court. ■