

Test fonctionnel et Développement Agile

Introduction

Les méthodes agiles (Scrum ou XP par exemple) [1] connaissent depuis plusieurs années un développement important dans

le domaine logiciel car elles répondent à des enjeux fondamentaux : la priorité donnée à la valeur recherchée par le client, une réalisation incrémentale évitant l'effet tunnel, une gestion permanente des évolutions et du refactoring de

code pour garder l'application la plus simple possible, des rétrospectives régulières et une démarche d'améliorations continues.



Figure 1 – Fausse interprétation classique des méthodes agiles

Le rôle clé du test dans les démarches agiles

Le test fonctionnel constitue un point clé des projets agiles pour maintenir un bon alignement entre les besoins métiers et les développements itératifs. Les tests vont nécessairement s'organiser au sein des itérations (qui peuvent être de durée très courte, typiquement de 2 à 4 semaines), sur un mode permettant de garantir la stabilité du code et le bon alignement avec les besoins métiers. La Figure 2 (la Pyramide de Mike Cohn) montre les principaux niveaux de test mis en œuvre.

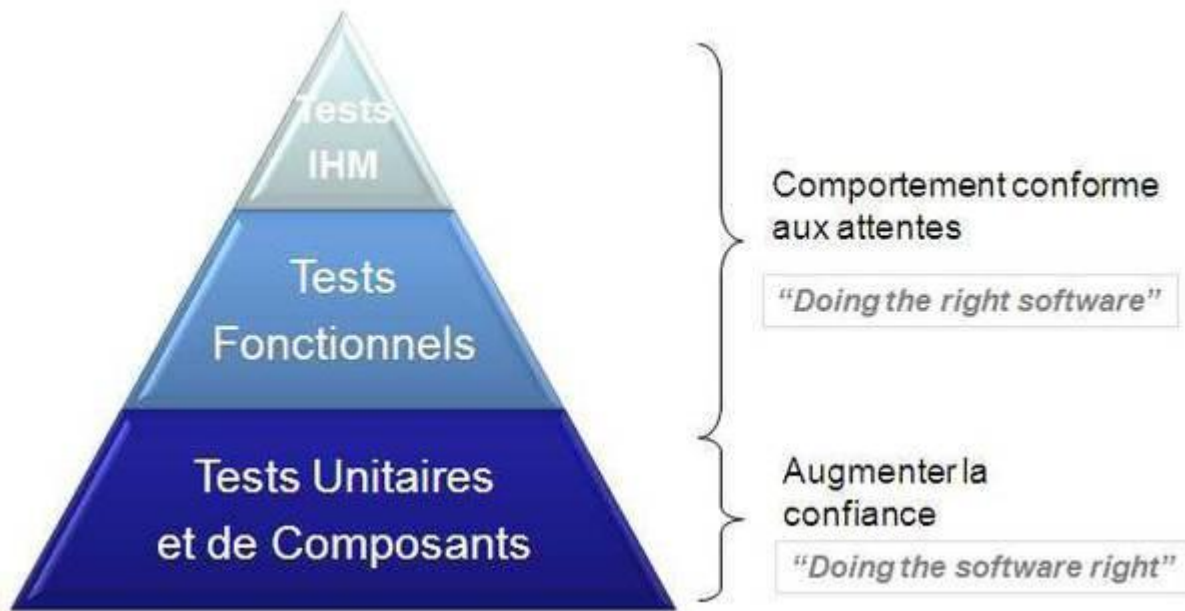


Figure 2 – Pyramide du test sur projets agiles de Mike Cohn

Les tests unitaires et de composants constituent la base de l’édifice : les approches de type TDD – Test Driven Development [2] – permettent d’assoir la qualité du code et la fréquence des évolutions sur un solide socle de tests. Il s’agit d’assurer une forte stabilité applicative (« Doing the software right »).

Le test IHM concerne la partie haute du test itératif : on s’assure de l’ergonomie et des performances de l’IHM. La forme de cette pyramide hérite d’un constat : dans un mode manuel, cela coûte très cher de faire du test IHM.

Le test fonctionnel constitue la partie centrale de la pyramide : il vise à s’assurer de la bonne adéquation des développements avec les besoins métiers devant être couverts dans l’itération, mais aussi à garantir la non régression applicative associée aux évolutions (ce qui fonctionnait doit toujours fonctionner). Il s’agit de vérifier en permanence la conformité aux besoins exprimés par le client et implémenter au sein de l’équipe agile (« Doing the right software »). Mais il s’agit aussi de maintenir la qualité globale d’un édifice construit et livré de façon incrémentale.

L’effort de test, fonctionnel en particulier, prend donc une importance encore plus grande dans les approches agiles.

Les approches du test fonctionnel haut niveau dans les projets agiles

Les bonnes pratiques en matière de test fonctionnel au sein des projets agiles (cf [3] & [4]) s’articulent autour de deux pratiques essentielles:

- La mise en œuvre de tests exploratoires manuels couvrant les fonctionnalités nouvelles produites lors de l’itération;
- La production d’un référentiel de tests automatisés couvrant de façon systématique les fonctionnalités définies et assurant en permanence la vérification de conformité aux besoins métier;

Le test exploratoire [5] consiste en un ensemble de techniques, de test manuel, qui vise à investiguer le logiciel sur la base des fonctionnalités attendues et de différentes heuristiques de test (valeurs aux limites, messages d’erreurs,

tour guidé et adaptation du parcours de test aux résultats obtenus, ...). Les limites du test exploratoire proviennent de son coût et le destine prioritairement aux fonctionnalités nouvelles.

La production du référentiel de test est une nécessité pour incrémentalement assurer une couverture systématique des exigences fonctionnelles ; Ce référentiel de test vise l'expression de tests d'acceptation basés sur les règles métiers et les comportements attendus, tant au niveau nominal (les cas passants) qu'au niveau des cas d'erreurs (dits non-passants). En pratique, l'automatisation est obligatoire et rentable, car il s'agit de pouvoir exécuter ces tests à chaque itération (donc très souvent), voire en continu si ils sont intégrés au build applicatif. Du coup, deux défis **majeurs du test fonctionnel automatisés** se trouvent exacerbés : comment faciliter l'expression à haut niveau des tests, le plus proche possible du métier ? Comment faciliter la maintenance des scripts automatisés ? Les solutions déployées aujourd'hui sur les projets agiles sont fondées sur une approche à base de mots clés et sur un découplage entre l'expression des scénarios métiers et l'implémentation des mots clés à des fins d'automatisation. C'est l'approche retenue par des environnements tels que FitNesse [6] et les approches de génération automatique de tests [4].

L'apport de la génération de tests pour gérer la production et la maintenance des tests fonctionnels

L'idée principale pour permettre de maîtriser la production et la maintenance des tests fonctionnels est de séparer la partie métier (les scénarios de tests à partir de mots clés) et la partie technique (l'automatisation des mots clés).

La génération de tests s'appuie sur un modèle comportemental pour la production des tests automatisés. Ce modèle est mis à jour en permanence en fonction des besoins métier, et permet la production des cas fonctionnels et les scripts associés via un générateur de tests.

Le modèle comportemental, véritable spécification exécutable de haut niveau de l'application testée, peut être défini au travers de notation textuelle comme dans l'outil Spec Explorer de Microsoft [7], mais aussi s'appuyer sur une modélisation graphique de type UML, permettant la représentation des processus et règles métier (voir l'article de IT-Expert de Novembre 2009 pour plus de détail - <http://www.it-expertise.com/ite-article/255/Normes-Standards-Qualite/model-based-testing-mbttests>).

Les avantages de ces approches pour le test fonctionnel dans le cadre de projets agiles sont :

- La séparation entre les aspects métiers et les aspects techniques des tests automatisés.
- La maintenance des bases de test accélérée par la factorisation du métier dans le modèle de test.
- L'augmentation de couverture permise par la génération automatique de tests, et ainsi le gain en sécurisation sur les applications;
- Le fort aspect incrémental ; les modèles de test sont développés et raffinés en phase avec l'évolution des développements incrémentaux.

L'organisation des équipes agiles conduit à la mise en place de rôles dédiés aux tests, qui développent et maintiennent le modèle de test en lien avec le client et les développeurs qui eux implémentent les mots d'actions pour l'automatisation.

Conclusion

En synthèse, l'évolution massive des processus de développements logiciels vers l'agilité pose des nouvelles questions aux pratiques de test ; Il s'agit de fait de tester plus souvent (en phase avec les itérations) et mieux (plus de couverture).

Le test fonctionnel joue un rôle essentiel au sein des méthodologies agiles, et la maîtrise de la production des tests automatisés (et de leur maintenance) constitue un enjeu essentiel. La génération de tests à partir d'un modèle de test apporte une solution efficace pour faciliter la production de tests, augmenter la couverture fonctionnelle et accélérer l'automatisation, en complémentaire avec l'approche de test manuel exploratoire.

Bruno Legeard

Références

- [1] Messenger Rota V, Gestion de projet vers les méthodes agiles, Editions Eyrolles, 2009, ISBN 978-2212125184.
- [2] Kent Beck, Test Driven Development: By Example, Addison-Wesley, 2002, ISBN 0-321-14653-0
- [3] Lisa Crispin, Janet Gregory, Agile Testing: A practical guide for testers and agile teams. Addison-Wesley, 2009, ISBN 978-0321534460
- [4] Bruno Legeard, Fabrice Bouquet, Natacha Pickaert, Industrialiser le test fonctionnel – Des exigences métier au référentiel de tests automatisés, Dunod, 2009, ISBN 978-2100515332.
- [5] James A. Whittaker, Exploratory Software Testing, Addison-Wesley, 2010, ISBN 978-0-321-63641-6.
- [6] FitNesse - <http://fitnesse.org/>
- [7] Spec Explorer - <http://msdn.microsoft.com/en-us/devlabs/ee692301.aspx>