

Survivre au déplacement plus en amont des phases de test

Dans une série d'articles, Paul Gerrard, consultant et spécialiste des tests, aborde diverses questions sur ces derniers. Le déplacement plus en amont consiste avant tout à ramener la réflexion sur les tests à un stade antérieur du processus. Dans cet article, Paul évoque les raisons de ce déplacement plus en amont et explique l'évolution du rôle des tests et la manière dont cela affecte les testeurs dans un monde plus agile.

Paul Gerrard
Gerrard Consulting

Sponsorisé par



Survivre au déplacement plus en amont des phases de test

Contexte

Il y a près de cinq ans, j'ai utilisé pour la première fois l'expression « tests redistribués » pour décrire un changement survenu au niveau des activités de test. Dans le cadre de ce changement, utilisateurs, analystes, développeurs et testeurs se partagent la responsabilité des tests et collaborent plus efficacement. Ce changement implique un déplacement des activités de tests (et éventuellement des responsabilités correspondantes) vers la gauche, autrement dit à un stade plus précoce du cycle, d'où l'expression « shift-left » employée couramment en anglais, qui signifie « déplacer vers la gauche ».

En vertu du principe du « shift-left », les développeurs assument davantage la paternité et la responsabilité de leurs propres tests. Cela suppose également que les testeurs soient impliqués plus tôt, qu'ils remettent en question les définitions des besoins et qu'ils fournissent aux développeurs des exemples par le biais d'un processus de développement axé sur le comportement (*behaviour driven development*, ou BDD). Cela signifie que les utilisateurs et les analystes métier, au même titre que les développeurs, assument l'entière responsabilité des tests, et que l'entreprise pourrait se passer de testeurs et d'une équipe spécialement formée à cet effet. Après avoir passé en revue toutes les configurations, nous sommes arrivés à la conclusion qu'il n'existe pas de « vérité unique ».

Cet article décrit l'évolution du rôle des tests et les conséquences que ces changements entraînent dans un monde plus agile.

Le « shift-left » n'est pas nouveau

Depuis que je travaille dans le domaine des tests (à savoir 1992) et même probablement avant, les défenseurs des tests prêchent pour l'« exécution plus fréquente de tests plus précoces ». Le modèle W^1 , introduit par mon ami et collègue Paul Herzlich en 1993, suggérait que tous les artefacts d'un processus par étape (tant au niveau logiciel que de la documentation) peuvent (et doivent souvent) être testés.

Malgré la popularité de cette approche à une certaine époque, le nombre et la durée des étapes n'ont en fait que peu d'importance. Le principe sous-jacent était que les *sources de connaissances qui orientent la conception et le développement d'un logiciel doivent être remises en question ou testées*. Dans un projet en plusieurs étapes, cela peut impliquer des révisions formelles. Dans un projet agile, le testeur (ou le développeur, l'analyste métier ou l'utilisateur) peut suggérer des scénarios (ou des exemples) qui invitent l'auteur d'une exigence ou d'une user story à réfléchir à des exemples concrets et à en discuter avant que le code ne soit écrit.

Le déplacement vers la gauche consiste avant tout à ramener la réflexion sur les tests à un stade antérieur du processus.

Autrement dit, le déplacement vers la gauche exige simplement des testeurs qu'ils s'impliquent à un stade plus précoce et qu'ils posent des questions bizarres ? Est-ce aussi simple que cela ? Pas vraiment.

Qu'est-ce qui motive un déplacement vers la gauche ?

Plusieurs changements survenus sur le marché sont en jeu et suscitent de nouveaux comportements dans notre secteur d'activités. Certains ont démarré il y a près de cinq ans, mais d'autres sont plus récents. Mais qu'entend-on par « démarré » ? Cela signifie que les personnes vantant les mérites de nouvelles approches probantes étaient suffisamment nombreuses et convaincantes pour que d'autres les suivent dans cette démarche. Ces approches « ont amorcé un tournant » comme le décrit Geoffrey Moore², et sont devenues une alternative crédible pour les entreprises et la communauté IT dans son ensemble.

Voici les principaux changements impliqués dans le phénomène de déplacement vers la gauche :

1. L'approche BDD a permis aux développeurs, aux utilisateurs/analystes métier et aux testeurs de collaborer. Aujourd'hui, l'approche BDD gagne du terrain, car elle renforce la collaboration entre les équipes Agile.

2. Le concept de livraison continue³ existe déjà depuis 5 à 10 ans. Son origine remonte à l'époque où de grandes entreprises présentes sur la toile ont massivement automatisé la création de versions et adopté des approches d'automatisation de la mise en production. Désormais, la plupart des organisations présentes sur Internet ont adopté ce concept.
3. La livraison continue a systématisé et accéléré le processus de mise en production grâce à l'automatisation. Elle a par ailleurs fait ressortir les retards au niveau des déploiements et des changements d'infrastructure qui, jusque-là, étaient occultés par la lenteur des processus de création, de test et de mise en production. DevOps est un changement culturel, un état d'esprit en vertu duquel les développeurs collaborent bien plus étroitement avec les autres équipes, notamment la production IT. À l'heure actuelle, de nouveaux outils apparaissent presque chaque jour, tandis que les fournisseurs font la promotion de DevOps comme « la nouvelle révolution ». Cela crée une situation très médiatisée et dynamique.
4. Le SMAC, acronyme anglais qui représente les 4 leviers de l'entreprise numérique (à savoir réseaux sociaux, mobilité, analyse et Cloud), témoigne d'une évolution dans la manière dont les organisations gèrent leur activité et les changements au niveau des systèmes dans la sphère mobile. L'expérimentation dans le domaine des affaires, qui se produit lorsque les systèmes de production évoluent, est supervisée en détail. Les énormes volumes de données (« Big Data ») capturées sont traités et les analyses dont ils font l'objet servent de base pour la prise de décisions métier.

De fréquentes expérimentations effectuées sur des systèmes de production permettent d'innover « à la vitesse du marché ». L'expérimentation se trouve au cœur de ce qui semble être le mouvement majeur des années 2010, à savoir la « transformation numérique ». Cette transformation numérique, ou « digitale » est au centre de toutes les attentions (et de tous les budgets) en ce moment. Les marketeurs promettent aux clients un meilleur accès, plus rapide, sur de nouveaux canaux, notamment mobiles.

Mon article intitulé « Digital Transformation, Testing and Automation »⁴ décrit la révolution numérique et propose des réponses. Il peut donc être intéressant de le lire.

Quel est l'implication d'un déplacement vers la gauche pour les testeurs ?

En vertu du déplacement vers la gauche, chaque fois qu'il est possible de fournir des commentaires susceptibles d'aider l'équipe à comprendre et améliorer les objectifs, la définition des besoins, la conception ou l'implémentation, ces commentaires doivent indiscutablement être transmis. Ce comportement est comme une seconde nature pour de nombreux testeurs, mais pas tous. Les utilisateurs, les analystes métier, les développeurs et toute l'équipe doivent être disposés à fournir des commentaires et à les accepter. Cette transition se heurtera probablement à quelques résistances, mais l'objectif final est d'exécuter un meilleur projet, mieux informé, un point c'est tout.

Quel est le rôle du testeur dans le cadre du déplacement vers la gauche ? Ce comportement peut se résumer en trois mots : « impliquez-vous tôt », le plus tôt possible. Participez aux discussions, échangez vos idées et vos définitions des besoins, et communiquez pour chaque étape qui contribue à la valeur du produit fini. En clair, le testeur met au défi les sources de connaissances, qu'il s'agisse d'intervenants, d'utilisateurs, de développeurs, de documents ou de l'opinion la plus répandue.

L'approche la plus courante consiste à « mettre au défi via des exemples ». À toutes les étapes, ces exemples peuvent être considérés comme des tests. Ils peuvent être rapidement oubliés sitôt après avoir été utilisés, ou bien codifiés en tests automatisés ou en vérifications manuelles. Ces exemples peuvent être utilisés de façon tactique pour pointer du doigt les failles dans la pensée des gens ou être fournis aux développeurs qui peuvent s'en servir comme points de départ pour des tests. Ils peuvent également servir de support de coaching pour aider les utilisateurs ou les développeurs à concevoir de meilleurs tests.

Les projets logiciels ont été qualifiés de processus d'acquisition de connaissances⁴. Ces connaissances sont collectées tout au long du projet et évoluent généralement avec le temps. L'objectif du déplacement vers la gauche consiste à garantir ces connaissances au moyen de tests proches de la source et d'en vérifier la fiabilité avant qu'elles ne soient figées dans le code.

Le déplacement vers la gauche pousse plus loin la philosophie centrée sur les tests. Les méthodes Agile ont toujours favorisé la collaboration et les commentaires rapides, et le déplacement vers la gauche pourrait être perçu simplement comme l'approche du commentaire rapide par excellence.

Si vous l'adoptez, le déplacement vers la gauche a un effet radical sur la manière dont vous travaillez en tant que testeur.

En tant que testeur, comment dois-je appliquer le déplacement vers la gauche ?

Il semblerait que le déplacement vers la gauche ne soit pas qu'une mode. Quelles sont les conséquences sur ceux qui travaillent au sein d'une équipe chargée de tester les systèmes ? Si vous faites partie d'une équipe agile, cela a-t-il une quelconque importance ? Que devriez-vous faire ?

Cela fait déjà quelques temps que nous vantons les mérites d'un déplacement vers la gauche en tant que stratégie de test clé au sein des projets agiles. Dans un contexte d'agilité, une stratégie de test peut être perçue comme une série d'« interventions agiles ». Dans tous les projets apparaissent des opportunités pour collecter et présenter des commentaires. Le testeur doit se concentrer sur ces moments cruciaux et se tenir prêt à apporter sa contribution dans ces moments-là.

Lors d'un récent webinar⁶, j'ai présenté la pensée à l'origine de cette approche. Pour illustrer les moments où ces interventions sont susceptibles de se produire, j'utilise une étude de cas client. Dans vos propres projets, vous devez identifier vos « moments cruciaux » et identifier les choix que vous et votre équipe pouvez faire. Par exemple, devez-vous écrire des tests unitaires pour des développeurs, leur donner des exemples pour les aider à démarrer, ou les guider pour qu'ils améliorent leurs aptitudes en matière d'écriture de tests ?

Selon la réponse, votre rôle sera différent. Il se peut que les testeurs ne soient pas simplement en train de réfléchir, mais plutôt qu'ils procèdent à un déplacement vers la gauche et se mettent au service des développeurs. Cela n'est probablement pas la meilleure issue pour vous ou votre projet. Nous suggérons que vous identifiiez les moments cruciaux, que vous apportiez votre contribution et que vous négociiez avec votre équipe. Vous êtes capable de servir de conseiller et de leadership pour les tests, au lieu de vous contenter d'en assumer la responsabilité. Si vous adoptez cette approche, vous serez bien plus en mesure de démontrer votre valeur à l'équipe (qui n'aura plus besoin d'autant de testeurs).

Je suis un responsable de tests. Que dois-je faire ?

Si vous êtes déjà un responsable de tests, il vous sera peut-être plus difficile de justifier votre rôle si la direction table sur une réduction des coûts de test en introduisant un déplacement vers la gauche. Si votre entreprise s'oriente dans cette direction, vous avez probablement une décision à prendre concernant votre carrière à plus long terme. Où voulez-vous être dans cinq ans ? Dans six mois ? Nous avons identifié cinq options susceptibles de vous intéresser.

1. **Fournir aux entreprises des compétences en matière de test** : en remontant la chaîne vers les parties prenantes, votre rôle pourrait être de prodiguer des conseils aux responsables métier qui souhaitent prendre le contrôle de leurs projets informatiques. En tant qu'agent indépendant, vous comprenez les préoccupations de l'entreprise et en tenez compte dans les projets. Vous conseillez et chouchoutez les responsables de projets, supervisez leurs performances et réalisations, interprétez les résultats et conseillez vos parties prenantes.
2. **Gérer les connaissances en matière de définition des besoins** : dans ce rôle, vous prenez le contrôle des connaissances requises pour définir et créer des systèmes. Grâce à vos précieuses compétences, vous êtes à même de garantir la clarté et la précision des besoins et des exemples qui illustrent les fonctionnalités utilisées. Vous aidez les équipes métier et les développeurs à déterminer à quel moment la définition des besoins est suffisamment fiable pour que les logiciels puissent être créés et testés raisonnablement. Vous gérez les définitions des besoins, ainsi que les glossaires et dictionnaires concernant l'utilisation des concepts métier et des éléments de données. Vous fournissez un service d'analyse de l'impact métier.
3. **Devenir TestMaster** : vous endossez une fonction de garant auprès des équipes, des projets et des parties prenantes. Il s'agit d'un rôle similaire au premier susmentionné, mais pour des environnements plus agiles. Vous êtes un spécialiste des tests, qui veille à la bonne réalisation des projets agiles. Vous travaillez en étroite collaboration avec des clients sur site et propriétaires de produit. Vous aidez les équipes de projets à identifier les risques et à y réagir, vous les coachez et les guidez, et vous contribuez à la gestion des activités de test et, à l'occasion, à la réalisation de tests.

4. **Devenir DevOpsMaster** : vous gérez le flux stratégique d'informations depuis et vers les processus DevOps (processus de création, de test et de déploiement automatisés). Ce flux d'informations est crucial. Vous pouvez éventuellement définir et superviser les processus appliqués pour gérer les flux qui permettent de contrôler les changements, les tests et la livraison.
5. **Gérer des équipes externalisées/hors site** : dans ce rôle, vous renoncez à constituer une équipe de test sur site et gérez le transfert des tâches vers des prestataires externes ou hors site. Vous vous posez en expert du flux d'informations et gérez les communications avec l'équipe de test externe, supervisez les performances et garantisiez les résultats.

Si vous n'avez pas encore opéré de déplacement vers la gauche, examinez les éléments (externes et internes) qui entourent votre équipe et réfléchissez à la manière dont votre rôle pourrait évoluer. Votre rôle finira par changer, mais vous pouvez choisir l'orientation que vous voulez lui donner. Je vous souhaite de faire les bons choix.

À propos de l'auteur

Paul Gerrard est consultant, enseignant, auteur, webmaster, développeur, testeur, conférencier, entraîneur d'aviron et éditeur. Il a assumé de nombreuses missions de conseils concernant tous les aspects des phases de test et d'assurance qualité logiciels, et s'est spécialisé dans l'assurance qualité des tests. Il a organisé des présentations majeures et des didacticiels lors de nombreuses conférences sur le thème des tests, en Europe, aux États-Unis, en Australie et en Afrique du Sud, et a été récompensé à plusieurs reprises.

Diplômé des universités d'Oxford et Imperial College de Londres, Paul a été lauréat en 2010 du prix d'excellence Eurostar European Testing, et en 2013 du prix The European Software Testing Awards (TESTA) Lifetime Achievement.

En 2002, Paul a coécrit l'ouvrage « Risk-Based E-Business Testing » avec Neil Thompson. En 2009, il a écrit « The Tester's Pocketbook ». En 2011, il a coécrit « The Business Story Pocketbook » avec Susan Windsor et a rédigé « Lean Python » en 2014.

En 2014, Paul a été également président de programme à la conférence EuroSTAR de Dublin.

Il est directeur de Gerrard Consulting Limited, directeur de TestOpera Limited et hôte du Test Management Forum.

Courriel : paul@gerrardconsulting.com

Twitter : [@paul_gerrard](https://twitter.com/paul_gerrard)

Site Internet : gerrardconsulting.com

Pour plus d'informations, rendez-vous sur la page **Développement et test** de CA Technologies.



Restez connecté à CA Technologies sur ca.com/fr



CA Technologies (NASDAQ : CA) fournit les logiciels qui aident les entreprises à opérer leur transformation numérique. Dans tous les secteurs, les modèles économiques des entreprises sont redéfinis par les applications. Partout, une application sert d'interface entre une entreprise et un utilisateur. CA Technologies aide ces entreprises à saisir les opportunités créées par cette révolution numérique et à naviguer dans « l'Économie des applications ». Grâce à ses logiciels pour planifier, développer, gérer la performance et la sécurité des applications, CA Technologies aide ainsi ces entreprises à devenir plus productives, à offrir une meilleure qualité d'expérience à leurs utilisateurs, et leur ouvre de nouveaux relais de croissance et de compétitivité sur tous les environnements : mobile, Cloud, distribué ou mainframe. Pour en savoir plus, rendez-vous sur ca.com/fr.

Références

1. « The W-Model », <http://blog.gerrardconsulting.com/?q=node/531>
2. « Crossing the Chasm » et autres ouvrages de Geoffrey A Moore, <http://www.chasminstitute.com/>
3. Définition de la livraison continue, Martin Fowler, <http://martinfowler.com/bliki/ContinuousDelivery.html>
4. « Digital Transformation, Testing and Automation », blog de Paul Gerrard, <http://blog.gerrardconsulting.com/?q=node/660>
5. « The Laws of Software Process », Philip G Armour.
6. Webinaire : « Agile Test Strategy », Paul Gerrard, <http://blog.gerrardconsulting.com/?q=node/627>