

Le test en charge à la vitesse Agile

Un livre blanc Neotys

Ce livre blanc décrit quelques-uns des défis du test en charge et de performance dans un environnement Agile, et fournit l'essentiel des meilleures pratiques dans ce domaine, comme prioriser les objectifs de performance et automatiser les tests avec des plate-formes d'Intégration Continue. Les responsables et les membres des équipes de test et toutes les personnes concernées par le test dans des équipes de développement Agile ou « orientées Agile » tireront profit de ces recommandations.

Sommaire

Introduction.....	3
Les avantages du test en charge et de performance continu.....	3
Éviter l'identification tardive de problèmes de performance.....	3
Corriger plus tôt pour limiter le coût des problèmes.....	3
Créer de nouvelles fonctionnalités, pas de nouveaux problèmes de performance ...	4
Les défis du test en charge dans un environnement Agile.....	5
Des cycles de développement plus courts exigent plus de tests en moins de temps .	5
Un code opérationnel n'est pas toujours performant.....	5
Fournir des résultats de test utiles aux développeurs, rapidement.....	6
Le passage entre Dev et Ops peut sembler risqué.....	6
Les meilleures pratiques.....	6
Inscrivez les SLA de l'application au tableau de tâches.....	6
Travaillez avec les développeurs pour anticiper les modifications.....	7
Intégrez les tests en charge à votre serveur de build.....	7
Intégration Continue + <i>builds</i> quotidiens + test en charge de fin de sprint.....	7
Comment choisir une solution Agile de test en charge.....	8
Conception simple de tests avec l'automatisation.....	8
Fonctions collaboratives.....	9
Intégration avec les serveurs d'Intégration Continue.....	9
Des rapports facile à produire et à interpréter.....	10
Des tests réalistes.....	10
Support des technologies les plus récentes.....	11
Conclusion.....	11

Introduction

L'agilité se vit au quotidien. Vous n'êtes peut-être pas « 100% Agile » et vous ne faites peut-être pas d'Intégration Continue, ni même envisagez le DevOps, mais la pression est bien réelle et de plus en plus forte pour profiter des avantages qu'apportent les méthodes de développement Agile ou « orientées Agile » comme la qualité et la vitesse.

Quand vous commencez à devenir plus Agile, vos développeurs se mettent à coder rapidement pour essayer de développer un maximum de récits utilisateurs (user stories) ou de tâches avant la fin du sprint, et les testeurs doivent tout faire pour garder ce rythme. De plus, les testeurs au sein d'équipes Agiles sont souvent responsables, en plus des tests en charge et de performance, des tests automatisés, des tests unitaires et des tests de régression. Dans ce contexte, il faut faire face à la cadence de développement tout en répondant aux fortes attentes de qualité.

Les avantages du développement Agile sont reconnus (notamment des délais raccourcis de mise sur le marché, une meilleure adaptabilité, un feedback continu, etc.). Ceux du test en charge et de performance le sont aussi (notamment la capacité à définir la charge maximale supportée par une application en production, savoir quand ajouter un serveur, quand reparamétrer le réseau, où optimiser le code, etc.). Ce qui est moins connu, c'est le fait que la combinaison des deux pratiques peut apporter des synergies qui dépassent la somme des deux bénéfiques (c'est-à-dire $2+2=5$).

Les avantages du test en charge et de performance continu

Éviter l'identification tardive de problèmes de performance

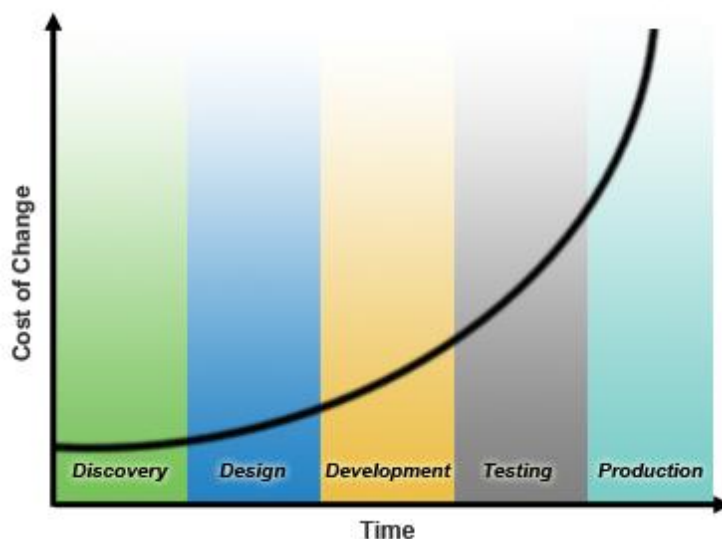
Quand les tests en charge et de performance sont relayés en fin de cycle de développement, il ne reste souvent que peu, voire pas, de temps aux développeurs pour apporter les modifications nécessaires. Ceci peut amener les équipes à différer les dates de sortie de version et à reporter certaines fonctionnalités attendues par les clients. Par ailleurs, si les problèmes de performance identifiés sont mineurs, les équipes peuvent décider de mettre l'application en production, tout en acceptant les risques encourus. Mais si les problèmes de performance sont plus critiques, ils peuvent entraîner des bouleversements d'architecture qui peuvent prendre des semaines ou des mois à mettre en œuvre.

Corriger plus tôt pour limiter le coût des problèmes

Associer des tests en charge et de performance aux processus de tests d'Intégration Continue permet d'identifier les problèmes de performance bien avant que le coût de résolution de ces derniers n'explose. Les développeurs peuvent ainsi détecter immédiatement que la nouvelle fonctionnalité du dernier *build* induit un franchissement des seuils de qualité de service (SLA) prévus. Ils peuvent alors régler le problème sur-le-champ à un coût mesuré. C'est particulièrement vrai pour les équipes Agile, pour lesquelles la découverte d'un problème de performance après

Le test en charge à la vitesse Agile

plusieurs semaines signifie que plusieurs *builds* sont concernés, ce qui rend l'identification de son origine cauchemardesque.



Créer de nouvelles fonctionnalités, pas de nouveaux problèmes de performance

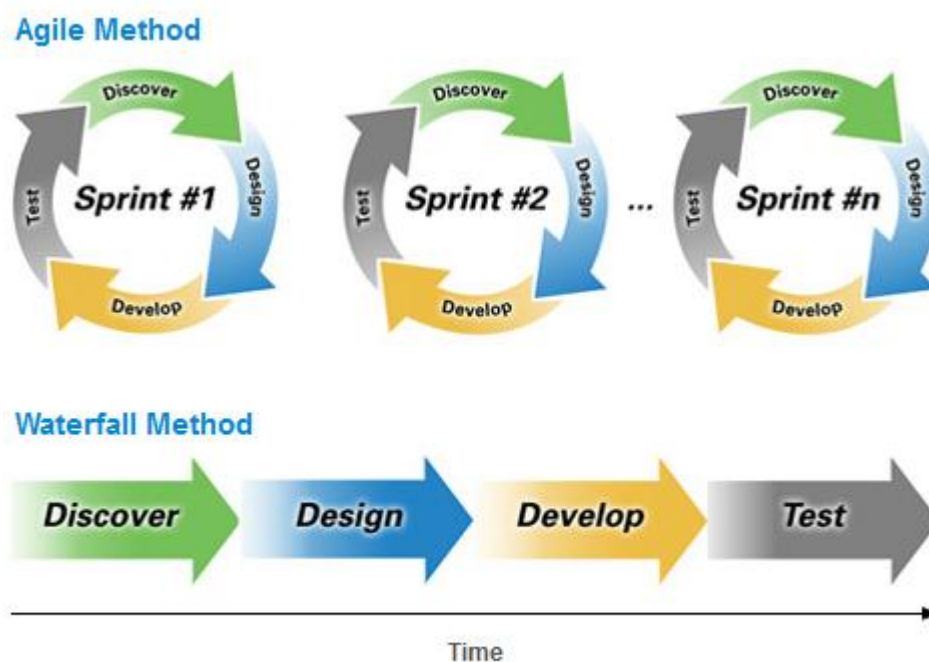
Dans certaines équipes Agile, les évolutions sont extrêmement rapides. Il se peut qu'en seulement quelques minutes, un nouveau module ou une nouvelle fonctionnalité soit contrôlée dans l'outil de gestion des sources, qu'un *build* soit exécuté dans l'Intégration Continue, que tous les tests automatisés soient lancés, et que l'application soit déployée. Mais si ce code n'est pas optimisé pour gérer le nombre d'utilisateurs simultanés atteint en période de pics de charge, il peut provoquer le crash du système entier. Intégrer des tests en charge dans le processus avant la mise en production de ces modifications garantit à vos utilisateurs l'accès aux fonctionnalités qu'ils attendent, sans risque d'expérience utilisateur décevante. C'est, pour votre entreprise, une économie qui se chiffre en milliers, voire en centaine de milliers d'euros sur des revenus qui peuvent être perdus si certains de vos utilisateurs se tournent vers la concurrence, ou que votre image de marque est ternie par des problèmes rencontrés avec votre application.

Les défis du test en charge dans un environnement Agile

De la même manière que les synergies Agile / tests en charge peuvent offrir des avantages incontestables, cette organisation peut aussi constituer pour vos équipes des défis d'une nature complètement nouvelle.

Des cycles de développement plus courts exigent plus de tests en moins de temps

Les tests en charge et de performance sont souvent réalisés à la fin du cycle de développement. Avec la méthode Agile, les cycles de développement sont plus courts et le test en charge et de performance peut être repoussé jusqu'au dernier jour d'un sprint, ou même n'être réalisé qu'un sprint sur deux. Cela se traduit souvent par un code qui n'est pas testé correctement, ou par des user stories qui sont reportées sur la prochaine version, lorsqu'elles ont été testées. Idéalement, la solution serait de tester plus tôt dans le cycle de développement ; mais c'est souvent difficile à réaliser compte tenu du manque de ressources et d'outils nécessaires pour y parvenir.



Multiple sprints (Agile Method) versus one development cycle (Waterfall Method)

Un code opérationnel n'est pas toujours performant

Dans les équipes agiles, les développeurs s'appliquent à fournir du code "opérationnel", mais un code est-il vraiment opérationnel s'il ne tient pas en situation de charge ? Peut-on valider une user story ou une tâche si le code associé fait que l'application s'écroule au-delà de 100 utilisateurs

Le test en charge à la vitesse Agile

simultanés ? Que se passera-t-il avec 1 000 ou 100 000 utilisateurs ? La pression pour sortir du code est très forte, mais le coût d'une application qui crashe en production aussi.

Fournir des résultats de test utiles aux développeurs, rapidement

Les développeurs agiles doivent connaître tous les détails des raisons pour lesquelles leur code génère des problèmes de performance : ils doivent savoir à partir de quand les problèmes ont commencé, et quelle user story est concernée. C'est un effort considérable pour les développeurs de devoir corriger un code sur lequel ils ont travaillé il y a plusieurs semaines ou plusieurs mois. C'est aussi autant de temps qu'ils ne passeront pas à travailler sur de nouvelles fonctionnalités. Identifier les problèmes de performance tôt dans le cycle de développement et en informer les développeurs permet de limiter les coûts.

Le passage entre Dev et Ops peut sembler risqué

Même si les concepts de DevOps et d'Intégration Continue sont assez récents, la crainte des équipes Opérations de voir une application crasher en production a toujours existé. Automatiser une partie des tests dans le processus d'intégration continue permet de limiter cette peur. Mais sans test de performance approprié, le risque reste bien réel. Les équipes Opérations connaissent bien l'impact énorme sur leur activité de l'interruption de service d'une application.

Les meilleures pratiques

Les meilleures pratiques décrites dans ce chapitre peuvent vous aider à tirer profit des avantages du test en charge dans un environnement Agile, et à surmonter le défi de sa mise œuvre.

Inscrivez les SLA de l'application au tableau de tâches

Chaque application a son propre niveau de qualité de service (SLA) minimum. Mais les équipes Agiles sont souvent plus concentrées sur la conception de modules et de fonctionnalités que sur l'optimisation de la performance applicative. Les récits utilisateurs sont généralement rédigés d'un point de vue fonctionnel (par exemple « En tant qu'utilisateur, je peux cliquer sur le bouton « Voir mon panier » pour afficher la page « Mon panier »), sans préciser les critères de performance de l'application (par exemple « En tant qu'utilisateur, je peux cliquer sur le bouton « Voir mon panier » pour afficher la page « Mon panier » en moins d'une seconde quand le site est consulté par 1 000 utilisateurs »). Ce n'est peut-être pas la meilleure façon d'écrire un récit utilisateur, mais elle illustre le point suivant : La mesure de performance doit être inscrite clairement sur le tableau de tâches pour que l'équipe y prête vraiment attention.

Le test en charge à la vitesse Agile

Une façon pour être sûr de voir la performance inscrite sur ce tableau, consiste à se servir de vos SLA comme tests d'acceptation pour chaque scénario. Un scénario ne peut pas être considéré comme « validé » tant que des modifications empêchent la conformité aux SLA (il sera alors peut-être nécessaire de définir de nouveaux SLA pour les nouvelles fonctionnalités et les nouvelles applications). Cette approche est efficace quand les modifications apportées au récit utilisateur ne concernent qu'une partie réduite du code (par exemple une nouvelle fonction de recherche doit retourner des résultats en moins de 2 secondes), auquel cas les éventuels problèmes de performance sont limités à une petite portion de l'application.

Pour les SLA généralisés à *toute* l'application (par exemple toutes les pages doivent être chargées en moins d'une seconde), des tests seront ajoutés à une liste étendue de contraintes (qui peuvent inclure des tests fonctionnels) qui sont testées pour chaque récit utilisateur afin de déterminer si le scénario est conforme aux critères minimum et peut être déclaré « validé » car répondant à toutes les contraintes.

Travaillez avec les développeurs pour anticiper les modifications

Un des avantages du test en environnement Agile est d'être informé des mises à jour des tâches de développement lors des réunions quotidiennes. Afin de tirer le meilleur parti de ce niveau de collaboration, les testeurs doivent constamment s'interroger sur la manière dont les récits utilisateurs en cours de développement devront être testés. De nouveaux tests en charge seront-ils nécessaires ? Le nouveau code va-t-il induire des modifications dans les scénarios des tests actuels ? Puis-je m'en tirer avec une simple adaptation des scénarios de tests actuels sans risque pour la suite ? La plupart du temps, il s'agit de modifications mineures, et les testeurs peuvent garder une longueur d'avance s'ils sont constamment et fortement impliqués.

Intégrez les tests en charge à votre serveur de build

Même si vous n'avez pas encore complètement fait le pari de l'Agilité, vous avez sans doute un serveur de *build* qui déclenche des tests automatisés, des tests unitaires, des « smoke tests », des tests de régression, etc. De la même manière que les objectifs de performance doivent être ajoutés au tableau de tâches, chaque *build* doit être accompagné de tests de performance. Cela peut être aussi simple que de définir le lancement automatique sur le serveur de *build* pour démarrer le test. Mais en fonction de la sophistication de l'intégration, cela peut aussi inclure l'affichage de résultats de test dans l'outil de *build*. Idéalement, la personne qui lance le *build* doit pouvoir consulter immédiatement les résultats des tests et connaître les modifications apportées au *build* pour les corriger en cas de problème de performance.

Intégration Continue + *builds* quotidiens + test en charge de fin de sprint

Le test en charge à la vitesse Agile

Les différences entre les *builds* d'Intégration Continue, les *builds* quotidiens, et les *builds* générés en fin de sprint peuvent être énormes. Dans le premier cas Il s'agit des modifications relatives à un « commit » sur un serveur de contrôle de version, dans le second cas ce sont toutes les modifications publiées dans une journée et dans le troisième cas ce sont toutes les modifications exécutées lors d'un sprint. Dans cette optique, il faut ajuster vos tests en charge selon le type de *build* exécuté.

La meilleure pratique dans ce cas consiste à commencer petit, et en interne. Pour les *builds* d'Intégration Continue exécutés à chaque fois qu'un développeur publie une modification, les tests doivent être lancés immédiatement pour fournir au développeur des informations sur l'impact de ses modifications sur le système. Envisagez d'exécuter un test de performance simple sur les scénarios les plus courants avec une charge normale, générée par vos propres injecteurs de charges. Pour les *builds* quotidiens, intensifiez ce test pour inclure plusieurs scénarios plus complexes et augmenter la charge pour atteindre un volume correspondant à vos périodes de pointe, et vérifiez qu'aucun problème de performance n'est passé au travers de vos tests d'Intégration Continue. À la fin du sprint, il faut faire le test ultime : Envisagez alors de générer la charge depuis le Cloud pour analyser le comportement de l'application lorsque les utilisateurs se connectent à travers le pare-feu. Vérifiez que chaque SLA de la liste des contraintes est satisfaisant et que chaque récit utilisateur écrit pendant le sprint peut être marqué comme « validé ».

Comment choisir une solution Agile de test en charge

Alors que la plupart des solutions de test en charge peuvent vous permettre de le faire dans un environnement Agile, relativement peu d'entre elles vous permettent d'appliquer les meilleures pratiques décrites dans ce document, et de suivre le rythme Agile de vos équipes de développement.

Lors de l'évaluation d'une solution de test en charge Agile, vous devrez vous poser les questions suivantes :

1. Les tests peuvent-ils être conçus rapidement par un testeur ou par un développeur (et pas seulement par des experts du test de performance) ?
2. Mes équipes pourront-elles partager les scénarios et les résultats de test ?
3. Dans quelle mesure l'outil s'intègre-t-il avec les serveurs d'Intégration Continue pour automatiser le processus de test ?
4. Chaque membre de l'équipe pourra-t-il créer et interpréter les comptes-rendus des tests afin d'en déduire des décisions ?
5. Les conditions de test pourront-elles reproduire fidèlement ce qui se passe en production ?
6. La solution supporte-t-elle les technologies utilisées aujourd'hui pour concevoir l'application et celles que nous prévoyons d'utiliser à l'avenir ?

Conception simple de tests avec l'automatisation

Le test en charge à la vitesse Agile

Il est rare que les équipes Agiles aient un ingénieur de test spécialisé qui intervienne à la demande pour concevoir un nouveau test à chaque fois qu'une modification dans une application génère des erreurs. Vous disposez peut-être d'un ou deux testeurs dans vos équipes, mais ce sont rarement des experts de la performance. Dans certains cas, ce sont vos développeurs qui écrivent les tests eux même. Dans l'un ou l'autre de ces scénarios, ces collaborateurs vont être sous la pression des délais, et un outil de test compliqué à utiliser et à paramétrer est inenvisageable.

Lors de la conception et de l'exécution de tests de performance, plusieurs éléments clés aident à améliorer la productivité des tests, notamment quand il s'agit de :

- Lancer facilement l'enregistrement d'un profil d'utilisateur virtuel (si possible en un seul clic),
- Définir des comportements avancés (avec des structures comme des conditions et des boucles) dans une interface graphique,
- Gérer automatiquement des paramètres dynamiques. Cela inclut un ensemble de règles de corrélation pour les frameworks de serveur les plus courants. Idéalement, la solution devra détecter et gérer dynamiquement les paramètres personnalisés spécifiques à votre application,
- Partager des portions communes de scénarios, comme les transactions de connexion ou de déconnexion, entre différents profils d'utilisateur virtuel.

Cette liste de fonctionnalités qui aident les collaborateurs à travailler plus efficacement n'est pas exhaustive ; Il faut plutôt la considérer comme une base de réflexion des caractéristiques attendues pour une solution de test en charge efficace.

Fonctions collaboratives

Le travail en équipe Agile repose entièrement sur la collaboration, et vous aurez probablement envie de partager, avec les membres de l'équipe, des éléments comme les profils d'utilisateur virtuel, les groupes, les configurations de monitoring, les politiques de charge, et les résultats de test. Les solutions de test en charge à examiner doivent permettre à différents collaborateurs de participer à la création et à l'analyse des scénarios de test et aux testeurs de partager facilement les résultats avec les développeurs pour qu'ils identifient les problèmes et les corrigent.

Intégration avec les serveurs d'Intégration Continue

Vous devez considérer une solution de test en charge qui s'intègre avec les serveurs d'Intégration Continue de sorte que vous puissiez automatiser le déclenchement d'un test de performance à chaque *build* et voir rapidement le résultat de ce test et des autres tests lancés pendant le *build* pour fournir des informations au développement. La capacité à voir l'évolution des performances sur plusieurs *builds* est fondamentale pour le test de régression. Les testeurs doivent être en mesure d'identifier précisément le *build* et les modifications de code au moment où la performance a commencé à se dégrader.

Le test en charge à la vitesse Agile

Automatiser l'interaction entre le système d'Intégration Continue et les tests en charge est essentiel pour garantir que la performance devienne un objectif régulier, à chaque itération. Si les tests en charge sont effectués manuellement dans les équipes Agile, il est tentant de les reporter sur le sprint suivant.

Des rapports facile à produire et à interpréter

Les membres de votre équipe n'étant pas tous des spécialistes du test de performance, vous devez chercher des solutions de test en charge qui produisent des rapports faciles à créer et à comprendre pour pouvoir prendre des décisions. Les développeurs ont besoin de rapports qu'ils peuvent interpréter pour apporter les modifications nécessaires au code, et que les équipes opérationnelles peuvent utiliser pour adapter les paramètres de l'infrastructure. Et les managers doivent pouvoir analyser ces rapports pour avoir une vision globale des indicateurs de performance. Des fonctions comme les statistiques par drag-and-drop et les filtres permettent de réduire considérablement le temps nécessaire à la production de ces rapports. Les rapports changent constamment, aussi est-il particulièrement utile que l'outil mémorise vos préférences et supporte différents formats, notamment PDF, Word, HTML et XML pour l'intégration avec d'autres systèmes.

Des tests réalistes

La plupart des applications web et mobiles actuelles sont conçues pour être utilisées par différents utilisateurs, dans des lieux variés, sur différents terminaux, avec des navigateurs multiples et des conditions de réseau variées. Une bonne solution de test en charge doit permettre de :

- **Simuler des conditions de réseau réelles.** Lors du choix d'une solution de test en charge, considérez un outil qui permet d'émuler le réseau. Elle doit pouvoir brider la bande passante et simuler la latence et la perte de paquets pour établir des conditions réalistes de téléchargement de contenu applicatif pour les utilisateurs virtuels. Cette fonction est particulièrement importante lors du test d'applications mobiles, car les terminaux mobiles bénéficient généralement de moins de bande passante que les ordinateurs portable ou fixes et sont considérablement impactés par les changements de latence et la perte de paquets, notamment quand le signal est faible.
- **Supporter différents terminaux et navigateurs.** De même, considérez une solution capable d'enregistrer à partir de n'importe quel navigateur ou de n'importe quel terminal, et d'en simuler le comportement au moment des tests en charge. Il est important de simuler les terminaux pour avoir les bonnes réponses du serveur et le nombre de connexions simultanées avec des temps de réponse et une charge de serveur réalistes. En outre, les navigateurs modernes ont la capacité de paralléliser les requêtes HTTP pendant qu'ils chargent les ressources statiques d'une page web. Ces requêtes parallèles exigent plus de connexions avec le serveur et peuvent augmenter les temps de réponse. Les solutions de test en charge qui ne parallélisent pas les requêtes sont inefficaces pour produire des tests de performance d'applications web vraiment réalistes.

Le test en charge à la vitesse Agile

- **Tester à l'extérieur du pare-feu.** L'écrasante majorité des applications web et mobiles sont consultées par des utilisateurs à l'extérieur du pare-feu. Il est absolument essentiel de réaliser des tests internes dans un environnement Agile mais pour comprendre vraiment l'effet de la situation géographique de vos utilisateurs sur la performance de l'application, vous devez envisager une solution qui est également capable de générer de la charge depuis des serveurs Cloud à travers le monde. Vous pouvez aussi considérer une solution qui génère de la charge depuis différents fournisseurs Cloud - pour réduire le risque - et qui permette la gestion des tirs cloud depuis une console de gestion unique pour tous les différents fournisseurs.

Support des technologies les plus récentes

Pour tester des applications conçues avec des technologies comme Adobe Flex, Microsoft Silverlight, RTMP, AJAX Push, HTML5 ou WebSocket, il vous faut un outil de test en charge qui les supporte nativement. Sans ça, il vous sera très difficile, voire impossible, de tester efficacement la performance de vos applications.

Ceci peut être particulièrement problématique avec des équipes Agile car elles souhaitent ajouter constamment de nouvelles fonctionnalités pour lesquelles les développeurs utilisent les toutes dernières technologies à leur disposition. Vos développeurs peuvent expérimenter Google SPDY et l'instant suivant ils ajoutent du streaming vidéo; vous devez être certain que votre solution de test en charge pourra contrôler l'impact de ces modifications sur la performance de l'application.

Conclusion

Le développement Agile et l'Intégration Continue augmentent la productivité des équipes et la qualité des applications. Ajouter des tests en charge et de performance à ces processus de développement nécessite une planification rigoureuse pour garantir que la performance devienne une priorité à chaque itération. Le but du test en charge « à la vitesse Agile » est d'apporter le maximum de valeur aux utilisateurs d'une application avec des fonctionnalités innovantes, tout en garantissant sa performance en situation de charge réaliste.

Pour être sûr de tirer le maximum de l'association des méthodes Agiles et du test en charge, vous devez :

- Vérifier que les SLA de performance sont sur le tableau de tâches ou sur la liste des contraintes pour garantir que le code associé à une tâche fonctionne parfaitement avant de marquer cette tâche comme « validée »,
- Coopérer avec les développeurs pour anticiper l'impact des modifications de code dans les scénarios de tests,

Le test en charge à la vitesse Agile

- Automatiser le lancement des tests de performance à chaque nouveau *build* et suivre l'évolution de cette performance entre chaque *build*,
- Augmenter progressivement la complexité et la charge des tests en fonction de la taille du *build* pour réduire les temps de génération des *builds* (Intégration Continue => smoke test, *build* quotidien => test en charge complet, fin de sprint => test aux limites avec des injecteurs de charge Cloud)

Lors du choix d'une solution de test en charge pour le développement Agile, vérifiez que l'outil peut s'adapter aux besoins de votre équipe et de votre application. La solution de test doit pouvoir :

- Permettre aux testeurs débutants et experts de concevoir et de modifier facilement des scénarios de test,
- Permettre à tous les membres de l'équipe de partager des scénarios et des résultats de test,
- S'intégrer parfaitement avec les serveurs d'Intégration Continue pour démarrer des tests et afficher l'évolution des résultats sur différents *builds* pour en contrôler les régressions,
- Générer des rapports que toute l'équipe peut comprendre pour pouvoir prendre des décisions,
- Exécuter des tests réalistes qui tiennent compte des terminaux, des navigateurs, des conditions de réseau et de la situation géographique des utilisateurs,
- Supporter les protocoles et les différentes technologies que vous utilisez pour concevoir vos applications actuelles et à venir.



A propos de Neotys | www.neotys.fr

Neotys, depuis 2005 et dans plus de 60 pays, aide plus de 1 500 clients à garantir la performance, la qualité et la fiabilité de leurs applications web et mobiles. NeoLoad, la solution de test de charge et de performance de Neotys est agile et facile à utiliser, sans aucun compromis sur le support des dernières technologies web 2.0 et mobiles. La plateforme Cloud de Neotys permet d'utiliser simplement toute la puissance du cloud pour générer des niveaux charge extrêmes. Neotys c'est aussi une équipe de consultants experts en performance qui vous accompagnent à tous les stades de votre projet pour assurer le succès de vos applications. Neotys est une société française qui investit 30% de son CA dans son centre de Recherche et Développement situé à Gémenos, près de Marseille.

Pour plus d'informations sur Neotys et NeoLoad, visitez : www.neotys.fr ou contactez sales@neotys.com

Contact pour plus d'informations

USA : Téléphone : +1 781 899 7200 | **EMEA** : Téléphone : +33 442 180 830

Email : sales@neotys.com | **En savoir plus** : www.neotys.fr

Neotys et NeoLoad sont des marques déposées de Neotys SAS aux Etats-Unis et dans d'autres pays. Toutes les autres marques commerciales appartiennent à leurs propriétaires respectifs.

Copyright © 2014 Neotys. Tous droits réservés. Reproduction interdite de tout ou partie de ce document sans autorisation expresse.