

AUTOMATISATION DES TESTS

LES CLÉS POUR RÉUSSIR L'AUTOMATISATION DES TESTS

A photograph of an astronaut in a white space suit floating in space. The astronaut is wearing a helmet and has a large backpack. The background is a dark blue space with a bright light source on the left and some stars on the right.

CloudNetCare
Plateforme d'automatisation
des tests

LES SECRETS DES TESTS

Avec l'évolution des technologies, des applicatifs et des logiciels, les mises à jours sont de plus en plus fréquentes et les enjeux de plus en plus importants. Pour répondre au niveau de qualité attendu par les utilisateurs les tests de non régression sont incontournables.

Le monde des tests n'a plus vraiment de secrets pour vous. Vous êtes un expert dans la réalisation de tests manuels ou semis-automatiques mais vous commencez à toucher du doigt les limites de ces deux mécanismes ...

Vous commencez à avoir du mal à répondre aux besoins, la couverture des tests et le temps nécessaire à leur réalisation deviennent problématique et au final vos campagnes de tests de non régression ralentissent vos mises en production car deviennent plus longues que les développements, ce qui est paradoxal.

C'est que vous êtes à deux doigts de basculer dans le "monde merveilleux" des tests automatisés. Pour que cette aventure se déroule sans accroc, nous vous révélons tous les secrets de l'automatisation des tests de non régression.

SOMMAIRE

1 - Les tests logiciels pour identifier les causes racines des dysfonctionnements

2 - Quand les tests de non régressions viennent ralentir vos mises en production

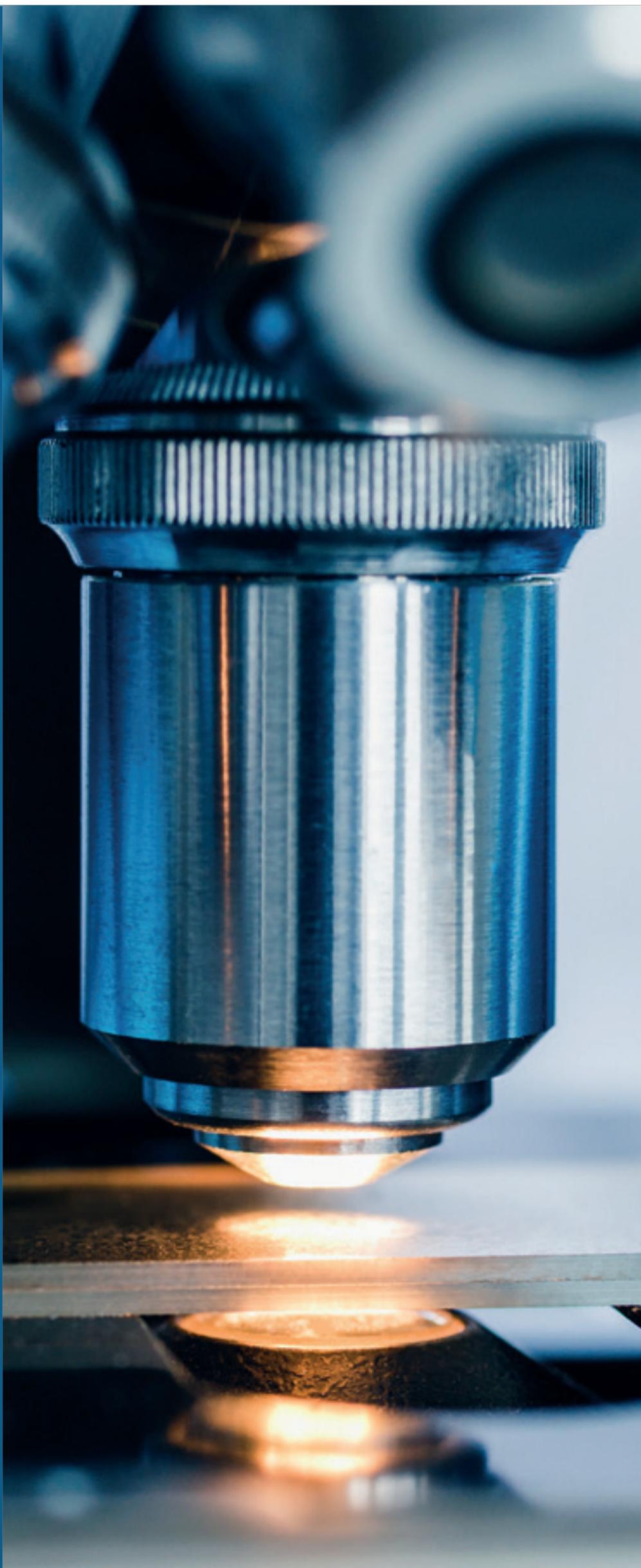
3 - L'automatisation des tests de non régression, une technologie sans limite ?

4 - Les pré requis avant de se lancer dans l'automatisation

5 - Les 4 moments clés pour automatiser

6 - Le nombre de personne pour mener à bien des TNR

7 - Comment se déroule la mise en place de tests automatisés chez CloudNetCare ?



Identifier les causes racines des dysfonctionnements

Les tests de non régression pour traiter les causes et non les symptômes.



À quoi servent les tests ? La question peut paraître étrange et pourtant elle a tout son sens. Bien souvent lors de la mise en place des tests logiciels, qu'ils soient fonctionnels, unitaires ou de non régression les équipes oublient le rôle premier de l'exécution de ces tests.

Les tests sont utiles pour réduire les risques d'occurrence de dysfonctionnements dans l'environnement opérationnel, pour mesurer la qualité des logiciels, pour atteindre des normes industrielles spécifiques ...

Ils sont incontournables et on ne cesse de le répéter : il faut tester, tester et encore tester. Seule certitude, les tests vont faire apparaître les dysfonctionnements mais est-ce bien leur VRAI rôle ?

Le Vrai rôle des tests logiciels et d'identifier les causes racines des dysfonctionnements

Si l'on s'arrête de tester, d'identifier et de corriger, on est dans un schéma du type "j'ai mal à la tête, je prends du paracétamol et je n'ai plus mal". Ce n'est pas pour autant que le lendemain, dans une semaine, dans un mois je ne me retrouverai pas dans cette même

situation car, en agissant ainsi, j'ai occulté la cause de mon raisonnement : pourquoi ai-je des maux de têtes ?

Pour les tests, c'est exactement la même chose. Dans bien trop de cas, on teste, on identifie un dysfonctionnement et on le corrige sans pour autant remonter jusqu'à l'origine de ce dernier.

Pourquoi ce dysfonctionnement à ce stade du projet ? Et pourtant, c'est ça le vrai rôle des tests : identifier les origines des dysfonctionnements.

Pour remonter à l'origine, on commence par catégoriser le dysfonctionnement rencontré :

- Erreurs
- Défauts
- Défaillances

Nb : pour comprendre la différence de ces nuances, voir notre article sur les typologies de dysfonctionnement.

L'objectif est de déterminer où le dysfonctionnement a été rencontré : lors des tests unitaires ? Lors des tests de non régression ? Ou lors de la supervision applicative ?

Il sera, bien entendu, plus facile de remonter à l'origine du dysfonctionnement si ce dernier apparaît au début du processus de développement.

Qu'est ce que les causes racines ?

Les causes racines, bien que nombreuses, sont souvent les mêmes.

Dans la majorité des cas on retrouve :

Une incompréhension : manque de spécification, faute de frappe, défaut de vérification ...

Un manque de temps : que des tests unitaires, couverture limitée ...

Un budget limité : tests manuels insuffisants, automatisation d'une seule partie des tests, environnement restreint ...

La difficulté est de fixer des priorités d'actions et d'être conscient des limites des tests effectués. En remontant systématiquement à la racine des dysfonctionnements vous devriez diminuer le pourcentage de dysfonctionnement de projet en projet.

Le graal, l'objectif mythique étant de conduire des développements logiciels sans rencontrer le moindre dysfonctionnement.

Les tests de non régression sont inhérents à la réussite d'un projet de développement.

Les tests sont inhérents à la réussite d'un projet de développement de logiciel ou d'applications. On ne peut se contenter de tester à la fin du processus de développement.

Il faut absolument intégrer les tests tout au long de la fabrication du logiciel.

Et, surtout, à chaque dysfonctionnement il faut prendre le temps nécessaire pour remonter à la cause racine pour que l'erreur, le défaut ou la défaillance rencontrée n'apparaisse plus sur ce projet, ni dans un autre !

L'automatisation pour accélérer les mises en production

Les tests : l'allié de la qualité et non l'ennemie de la productivité.

Dans un schéma "classique" vous développez une application, une fonction, vous effectuez les tests fonctionnels et de non régression nécessaires et vous procédez à la mise en production de votre évolution. Ce schéma va pouvoir s'appliquer si le rythme des évolutions n'est pas trop soutenu et surtout si vos campagnes de tests sont suffisamment industrialisées. Car si ce n'est pas le cas vos tests de non régression viendront ralentir vos mises en production.

Il y a fort à parier que les premiers tests que vous avez mis en place remontent à la création de votre logiciel ou de votre application. Après des mois, voire des années de développement et de tests exécutés manuellement, vous avez mis en production le fruit de votre travail !

Et, bien souvent, l'on constate que malgré tous les tests réalisés il subsiste encore des dysfonctionnements après la mise en production. Et, dans les faits, c'est presque inévitable. Pour s'en rendre compte, il faut prendre un peu de recul sur ce qu'il vient de se passer au cours des mois de développements et de tests.

Au démarrage du projet, vous avez fixé une date "butoir" de fin de projet qui correspond, bien souvent, à la mise en ligne du logiciel ou de l'application. À la suite de cela, les premiers développements commencent, les premiers tests aussi et les premières modifications également. La date étant encore suffisamment lointaine, vous prenez le temps nécessaire à la correction de toutes les erreurs de fonctionnement et continuez en ce sens : une phase de développement, une phase de tests, une phase de correctifs, une phase de tests. Une phase de développement ...

Le manque de temps est l'ennemi juré des tests de non régression.

Mais, au fur et à mesure de l'avancement du projet et la date "butoir" approchant (voire la nouvelle date butoir), les développements s'accroissent, les tests aussi, les correctifs ... tant est si bien que, à la fin, pour pouvoir mettre en production dans les délais impartis il aura fallu gagner du temps.



Les tests sont bien trop souvent perçus comme une contrainte.

Et dans 99% des cas, le temps a été gagné sur les tests :

- Soit en ne testant qu'une partie des dernières fonctions développées
- Soit en n'effectuant pas les tests après l'application des correctifs
- Soit en n'effectuant pas les tests de l'ensemble des fonctions développées

Ce processus itératif va se reproduire pour chaque nouvelle version que vous allez réaliser.

Inévitablement vous vous exposez à découvrir des régressions au fur et à mesure de l'utilisation de l'appli sur son environnement de production.

Quand le temps devient l'ennemi des tests.

Le problème c'est que pour conduire des tests qui couvrent l'ensemble du périmètre

fonctionnel cela prend du temps, beaucoup, beaucoup de temps !

Pour mener à bien des tests il faut :

Le minimum requis : tester la fonction sur un environnement "neutre" avec les bons jeux de données (voir notre article sur l'importance des jeux de données)

Le bon élève : ne pas se contenter de tester « la nouvelle fonctionnalité » mais bien toute la couverture de tests pour vérifier s'il n'y a pas d'« effet de bord » et de régression d'autres parties de votre logiciel.

Le zéro défaut : tester sur les différents environnements de production : navigateurs (et différentes versions) ? Systèmes d'exploitation (et différentes versions) ? Tester la fonction sur les bons terminaux : ordinateur, tablette, smartphone(s)...



Les tests de non régression sont incontournables et il est impératif de tester son applicatif avant sa mise en production.

Sans toutes ces étapes et ces différentes configurations vous ne pourrez pas être sûr que la version mise en production est stable.

Souvent, on occulte une grande partie de ces configurations en partant du principe que si cela marche pour une, il y a de grande chance que cela marche pour l'autre... et bien après 30 ans de retour d'expérience, on vous livre un secret : ce n'est jamais le cas !

Si vous ne pouvez pas vous permettre de prendre des risques au moment de la mise en production, vous prendrez le temps qu'il faut pour exécuter les tests.

Se faisant il devient malheureusement inévitable, en fonction de votre patrimoine de tests, que votre phase de tests de non régression viennent ralentir vos mises en production et vous vous retrouverez avec des

développements en "file d'attente".

Il y a fort à parier que l'on ne vienne vous reprocher des délais de mise en ligne trop long qui plus est si cela a un impact direct sur le business de votre entreprise.

Mais le problème reste le même. Si vous faites l'impasse sur la qualité des livrables, les utilisateurs s'éloigneront d'une version boguée. Vous l'aurez compris, dans tous les cas c'est votre faute.

Les tests de non régression sont souvent négligés pour respecter les délais de mise en production.

Il faudra vous poser les bonnes questions pour pouvoir gagner du temps sans négliger la qualité des tests. La seule solution : l'automatisation de tout ou partie de vos tests de non régression.

Les tests de régression sont souvent les négligés des projets de développement. Et pour cause, le temps nécessaire à leur bonne conduite est souvent une difficulté que beaucoup contourne car, sans cela, les tests ralentissent les mises en production.

Comme pour une assurance, ne pas en avoir n'a aucune conséquence jusqu'au jour où on en a besoin ! Gardez à l'esprit que selon la complexité des tests à mener ils représentent 35% des coûts de développement d'un logiciel et quasiment autant du temps total. Le seul moyen pour faire diminuer ces deux paramètres est d'automatiser vos tests.

L'automatisation : sans limites ?

Peut-on tout tester avec 100% d'efficacité en automatisant ?



L'automatisation des tests est une avancée technologique majeure qui vous offrira bien des avantages. Nous venons de le voir c'est un gain de temps indéniable et la fin des erreurs probables sur vos tests. Mais la transparence étant l'une de nos valeurs, nous allons nous attaquer un sujet qui peut être fâcheux pour certains : les limites de cette technologie.

Des tests toujours efficaces ?

Il existe toujours une marge d'erreur, et ce, quel que soit le sujet ou l'action que vous allez aborder. Mais dans les tests automatisés, il n'existe que 2 faits pouvant créer des problèmes :

La rigueur du cahier de recette. En effet, plus ce dernier sera précis, moins il y aura d'erreurs lors de la couverture des tests. C'est pour cela que nous vous le répétons : ne sous-estimez jamais la rédaction de votre cahier de recettes, et prenez le temps qu'il faut pour le mettre en place ! D'ailleurs, la mise en place du cahier de

recette prend environ 30% du temps du projet. Ainsi si votre projet dure 100 jours, la conception dudit cahier doit être d'environ 30 jours.

L'équipe de programmeurs. Comme dit l'expression, « l'erreur est humaine », c'est donc lorsque l'équipe humaine implémente les tests aux automates qu'il risque d'y avoir des erreurs, notamment dues à une mauvaise rédaction du cahier des charges, voire à un manque de concentration. Mais une fois que les automates sont programmés, ils effectueront la seule et même tâche sans jamais faillir ni diverger.

Pour résumer, s'il y a des erreurs dans l'automatisation des tests, les équipes humaines sont les seules responsables. Les automates, eux, ne font qu'exécuter inlassablement les ordres dictés par l'équipe. Du moins, pour le moment ...

Y-a t'il des limites dans la mise en place des tests ?

D'une manière générale, plus de 80% des tests sont automatisables. Le faible taux restant prend en compte des technologies beaucoup trop complexes pour le moment, comme par exemple le contrôle de connexion facial de l'iPhone X.

La mise en place de l'automatisation de ce genre de tests coûtera bien plus cher que de le faire manuellement, c'est pour cela que chez CloudNetCare, nous automatisons les tests uniquement lorsqu'il n'y a pas de marge d'erreurs, et pas de coûts inutiles pour vous.

Bien sûr, au regard de la croissance constante et fulgurante des avancées technologiques de nos jours, il y a fort à parier que d'ici quelques temps, 100% des tests seront automatisables. Comme promis, nous avons été d'une totale transparence concernant l'automatisation des tests. Et comme vous l'aurez compris, même en s'attardant sur les petits défauts qu'elle peut avoir, cette technologie reste tout de même très avantageuse pour n'importe qui.



30%

du temps du projet consacré au cahier de
recette.



35%

représentation des tests dans les coûts de
développement.

Les pré-requis de l'automatisation des tests de non régression.

Convaincu ? Êtes-vous vraiment prêt à vous lancer dans l'automatisation ?



Un scénario automatisable

Élémentaire mon cher Watson ! Et bien pas tant que ça. Tous les scénarios ne sont pas automatisables car toutes les fonctions end user ne peuvent pas être testées automatiquement.

Pour exemple, sur les nouveaux terminaux, la reconnaissance faciale ne peut pas être testée par une tâche programmée. Soyez donc certain que vos scénarios intègrent uniquement des séquences techniquement réalisables.

Un logiciel de tests pertinent

Par chance, le panel des logiciels d'automatisation des tests reste relativement restreint et vous n'aurez pas à parcourir des catalogues de solution pour dénicher l'élite !

Les logiciels viables se comptent sur les doigts d'une main. On ne serait que vous conseiller notre solution CloudNetCare mais on doute de manquer un poil d'objectivité. Quoi qu'il en soit, il y a quelques questions qu'il faut vous poser pour orienter votre choix :

- Que dois-je tester ? Souhaitez-vous mettre en place des tests côté back office et base de données ou alors des tests end-user ? Vous trouverez des logiciels spécifiques à ces typologies de tests.

- Ai-je intégré toute la problématique liée à la plate-forme de tests automatisés que vous allez mettre en œuvre ? Qui pour le configurer ? Qui pour le maintenir à jour ? Qui pour l'exploiter ? ...

- Quel modèle économique ? Souhaitez-vous

acheter une licence, vous tourner vers une solution "gratuite" et open source ou encore faire appel aux compétences d'un prestataire.

Les compétences nécessaires à disposition

Si vous avez déjà mis en place des tests semi-automatiques vous avez touché du doigt la pluralité des compétences nécessaires pour mettre en œuvre des tâches automatisées. Pour le logiciel sélectionné il faudra le configurer, développer et programmer les tâches, vérifier leur bonne exécution, le maintenir à jour, corriger les régressions du logiciel de tests semi-automatisés ...

Du côté de l'infrastructure il faudra acquérir le matériel nécessaire et implémenter les machines qui vont reproduire les tests, les maintenir à jour, corriger les régressions ...

Pensez à faire la liste des compétences à mobiliser, vérifiez la disponibilité de vos équipes si vous souhaitez conduire le projet en interne. Ou alors cela vous permettra d'identifier à quelles compétences faire appel en externe.

Ce pré-requis est fondamental. La majorité des projets d'automatisation ne voient jamais le jour car, finalement, il n'y avait pas les compétences nécessaires en interne ou chez le prestataire retenu.

L'équipement correspondant

Pour exécution les tests automatisés, il faut des machines bien évidemment, mais pour les reproduire il faut les environnements correspondant à vos besoins. Listez les

équipements nécessaires, le nombre de version à tester et vous aurez le nombre de système à maintenir. À cela il conviendra de suivre l'évolution technologique et donc de renouveler le parc en fonction du comportement des utilisateurs : nouveaux smartphone, nouvelles versions de système d'exploitation, ...

Si vous ne deviez retenir qu'une seule chose de ce point c'est de tout mettre à plat avant de vous lancer dans l'automatisation de vos tests.

Si vous avez besoin de déployer une solution rapidement avec des résultats fiables il faudra externaliser vos tests car la mise en place d'une solution pérenne en interne va vous demander du temps, beaucoup de temps ... et des compétences, beaucoup de compétences.

Mais si vous disposez de tous ces pré-requis, allez-y, le monde de l'automatisation vous tend les bras !

Les 4 moments clés où l'automatisation des tests devient pertinente.

Vous êtes prêts ? Mais ce n'est pas pour autant que l'automatisation soit la solution la plus pertinente à ce stade de votre applicatif.

Bon nombre de tests peuvent être mis en place et conduits manuellement. Il n'est pas nécessaire d'automatiser les tâches si le ou les tests restent épisodiques. Cela va de soit pour plusieurs types de tests : montée en charge et non régression par exemple. Les tests pour assurer la fiabilité de vos applicatifs rentrent toujours sous la contrainte du ROI. Que vous coûte réellement une série de tests générée manuellement ? Quelle est la marge d'erreur que vous êtes prêts à tolérer ? Autant de questions auxquelles il n'est pas évident d'apporter une réponse claire et précise. Alors, voici les 4 moments clés où l'automatisation des tests devient pertinente.

1 - La fréquence.

Élémentaire ! La fréquence est un critère qui va naturellement vous amener à vous poser la question de l'automatisation. Pour mettre en place et conduire un test de non régression une fois par an, il y a fort à parier que dans 9 cas sur 10 une réalisation manuelle du test répondra à vos besoins. Toutefois, il se peut que les caractéristiques de la campagne de tests impose une automatisation même ponctuelle.

À contrario, lorsque la fréquence de vos tests est de plus en plus rapprochée, l'automatisation peut devenir une vraie alternative, voire s'imposer naturellement par exemple en développement Agile.

2 - Quelle couverture de test ?

Pour tenir compte des différentes évolutions : fonctionnelles, technologiques, des navigateurs utilisés par les internautes ou des devices par les mobilnauts, la couverture nécessaire des tests est de plus en plus large. On s'explique : réaliser une campagne de tests de non régression avant une mise en production, c'est capital.

Mais la réaliser sur un seul environnement peut s'avérer insuffisant. Aujourd'hui, des tests de non régression pour un site internet doivent être effectués a minima sur 2 à 3 navigateurs : Chrome (68% des utilisateurs), Firefox (11%), IE (7%) – Source StatCounter et sur 2 à 3 machines et si vous avez une apps mobile ou un accès en responsive design sur plusieurs mobiles, tablettes.

On résume pour une campagne a minima d'une application disponible sur mobile et web, soit la majeure partie aujourd'hui, on a :

Soit, pour une configuration simple 5 environnements à tester donc autant de scénarii à exécuter, maintenir, faire évoluer .. et pléthore de résultats à analyser, répertorier, cartographier, à communiquer aux équipes de développement, ... Sans parler des KPIs à établir pour obtenir un retour sur cette MEP et améliorer la prochaine.

Si on combine la « fréquence X couverture fonctionnelle X environnement » on se rend compte qu'il est impératif de mettre en place une automatisation.

3 - Quels scénarii ?

La typologie du scénario est critique pour déterminer s'il est pertinent d'automatiser ses tests. Dans le cas d'un seul scénario où vous allez tester les points critiques du parcours utilisateur : valider que l'utilisateur peut mettre un produit dans son panier, s'enregistrer et payer, des tests manuels pourraient suffire. Mais ce cas n'arrive quasiment jamais car alors votre couverture de tests se restreint à une seule fonction, certes stratégique !

Encore que, comme expliqué ci-dessus, il faut corréliser cela avec la fréquence et l'environnement à tester.

Pour exemple un scénario plus précis et intégrant l'ensemble du parcours utilisateur de votre site internet ou votre application mobile :

- Chargement de la page d'accueil
- Chercher un produit dans la barre de recherche
- Cliquer sur la fiche produit
- Cliquer sur l'image
- Ajouter le produit au comparateur
- Aller dans le menu catégorie
- Aller et cliquer sur la sous-catégorie A
- Cliquer sur une fiche produit
- L'ajouter au comparateur
- Accéder au comparateur
- Sélectionner un des deux produits
- L'ajouter au panier
- Se connecter / Ou créer un compte
- Choisir son option de transport
- Payer
- Accuser réception du paiement

Le nombre d'actions par scénario peut vite augmenter, il est cependant impératif de segmenter au mieux les parcours pour les établir par fonction.

4 - Le nombre de profils ?

Le dernier des éléments clés est le nombre de profils à tester. Les parcours utilisateurs sur votre site ou votre applicatif varie-t-il en fonction des différents droits d'accès, ou selon des personnalisations par profil (connecté, non connecté, client récurrent, client VIP, Admin ...). Ces profils sont autant de coefficients multiplicateurs au nombre d'actions à reproduire. Dans notre exemple ci-dessus, admettons que 3 profils utilisateurs à tester :

On résume la formule magique : fréquence x nbr environnements x actions par scénario x profils = nbr d'actions à reproduire.

À raison d'une minute en moyenne par action : prise de connaissance de l'action, saisie de la valeur, action, étude du résultat, saisie du résultat et commentaire, vous avez devant vous une journée de tests à effectuer pour une seule phase de tests.

Bien entendu avant la livraison de la nouvelle MEP il sera nécessaire d'en réaliser plusieurs. Sachant que dans notre exemple nous n'avons que 4 scénarios. Nous vous laissons faire le calcul pour votre contexte. Attention le nombre de jours/hommes nécessaires va vous effrayer !

À cela s'ajoute que la récurrence des actions manuelles va augmenter le pourcentage d'erreur. À ce stade, pas de secret, l'automatisation des tests est la seule alternative pour assurer la fiabilité de vos applicatifs !



Comment évaluer le nombre de personnes pour mener à bien des tests de non régression ?



Les différents tests nécessaires au bon fonctionnement de vos applicatifs sont incontournables. Mais, pour les mener à bien, ces tests peuvent vite devenir chronophages. Pour minimiser les risques d'erreur, il est souvent pertinent d'impliquer plusieurs collaborateurs (pour savoir qui impliquer lors de vos tests de performance nous vous invitons à lire notre article [ici](#)). Mais entre l'effectif trop réduit et l'équipe en surnombre, quel est le juste milieu ?

I. Évaluer le contexte

La première étape consiste à définir un contexte aux différents tests :

Quels sont les enjeux ? Est-ce que les tests sont relatifs à une fonctionnalité cruciale (bouton d'achat, processus d'identification) ou secondaire. Est-ce qu'un dysfonctionnement entraîne l'arrêt complet du parcours utilisateur ?

Quels sont les délais ? En combien de temps les tests doivent-ils être réalisés ? La connaissance des délais vous permettra de planifier les différentes actions à conduire.

Quel coût ? Il est important de budgéter sa phase de tests pour juger de sa pertinence. On n'utilisera pas les mêmes moyens pour un test à faible valeur ajoutée que pour un test sur une fonctionnalité critique.

Quelle est la durée des tests ? Certains tests peuvent demander plusieurs heures pour être menés à bien en fonction des différents cas à étudier. Il est indispensable de confronter la durée des tests avec les délais disponibles (voir n°2). Le ratio trouvé permettra ou obligera d'impliquer X nombre d'utilisateurs.

Quels environnements ? Un des éléments fondamentaux des tests réside sur les environnements sur lesquels les tests se déroulent. Pour une application Web il est nécessaire de tester sur, a minima, quatre environnements différents correspondant aux quatre navigateurs les plus utilisés en France à savoir : Chrome, Safari, Firefox et Edge. Pour des tests conformes et reproduisant la réalité on ne peut demander aux testeurs de « jongler » sur une même machine entre ces quatre navigateurs, au risque de perturber les résultats obtenus.

Une fois le contexte déterminé, vous allez pouvoir commencer à réfléchir au nombre de personnes à impliquer, le matériel nécessaire, pour mener à bien vos différents tests. Il peut arriver que les enjeux des tests soient marginaux et qu'il devienne presque obsolète de mettre en place des tests rigoureux et qu'un simple contrôle manuel suffise. Si ce n'est pas le cas, vous allez devoir déterminer le nombre de personnes à impliquer et c'est là que le plus dur commence.

II. Critères d'évaluation

En fonction du contexte déterminé ci-dessus, vous allez pouvoir identifier le nombre pertinent de personnes à impliquer selon les critères ci-dessous :

Quel est le budget ? Sans budget, pas de test. C'est ce dernier qui, dans un premier temps, va permettre de savoir si vous avez le luxe d'impliquer plusieurs personnes pour mener à bien vos différents tests.

Quels sont les différents profils ? À chaque utilisateur un profil différent. Il est souvent

pertinent d'impliquer des utilisateurs avec des profils complémentaires pour être sûr de reproduire toutes les typologies d'action. Vous éliminez considérablement les risques d'oubli. Ratio délai / durée : comme évoqué ci-dessus, selon les contraintes de temps vous n'aurez peut-être pas le choix que d'impliquer plusieurs collaborateurs dans un souci de productivité. Il n'est jamais bon de procéder de la sorte mais on ne sait que trop qu'il est fréquent de procéder à des tests en "urgence" avant une mise en production.

Automatisable ? Il est fréquent que tout ou partie de vos tests soient automatisable. Vous gagnez ainsi en temps, en fiabilité voir même en budget selon les cas. Si vous anticipez vos tests vous allez pouvoir demander une étude du coût de l'automatisation. Vous pourrez ainsi comparer le budget global (temps passé, coût des ressources humaines VS coût de l'automatisation). Pour une étude complète et rapide de votre besoin nous sommes à votre disposition.

Ces différents critères vont vous permettre d'estimer le nombre de personnes à impliquer pour vos phases de tests. Par la suite vous n'aurez plus qu'à planifier les différentes séquences en fonction de la disponibilité de chacun. Au-delà de 3 à 4 personnes impliquées et/ou d'un délai trop important pour tester la solution il devient intéressant de se tourner vers l'automatisation d'une partie de votre recette. Toutefois, l'automatisation ça s'organise, s'anticipe et s'étudie. Il est donc primordial d'anticiper cette phase d'analyse en commençant la réflexion en même temps que le développement de votre nouvelle fonctionnalité.

Pour mener des tests efficaces il faut y consacrer du temps, mobiliser ses équipes et tout cela coûte de l'argent qu'il faut prévoir dans les budgets de développement. C'est le ratio coût/délai/efficacité/durée qui vous permettra de déterminer le nombre de personnes à impliquer ou que vous pouvez impliquer.

Comment se déroule la mise en place de tests automatisés chez CloudNetCare ?

Simple, rapide et efficace.

ETAPE 1

La découverte de VOS tests

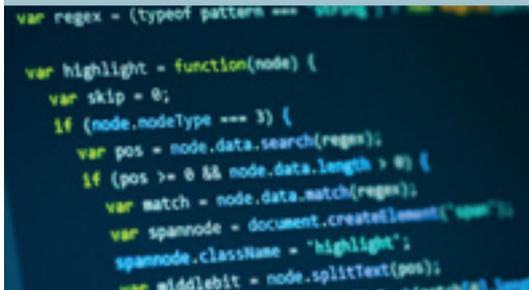


Après la prise de contact, nous analysons votre besoin. Ce dernier nous permet de mettre en place une démonstration complète du test choisi composé de deux ou trois scénarios que vous aurez défini. À la suite de cette démonstration, vous aurez des résultats concrets quant à l'efficacité de nos tests.

Si après la découverte des résultats vous montrez toujours un intérêt à poursuivre notre collaboration, nous passons à l'étape 2.

ETAPE 2

POC dans VOTRE contexte



L'étape 2 consiste à conduire les tests déterminés à l'étape 1 sur une à trois semaines. On le rappelle, vous n'avez pas à verser le moindre euro et vous pouvez arrêter l'expérience dès que vous le voulez.

À l'issue de cette seconde étape, vous vous verrez proposer un bilan complet, et si vous êtes toujours aussi intéressé, nous passerons à l'étape 3, qui est la phase pilote.

ETAPE 3

Phase Pilote de l'automatisation



Cette étape s'étale entre 1 et 3 mois, et il s'agit de la réalisation des tests qui correspondent à vos besoins sous un SER (sous ensemble représentatif). Le coût est déterminé de manière très réduite et ne prend en compte que le coût engendré par la phase pilote.

Ensuite, c'est à vous de nous faire part de l'abonnement que vous souhaitez concernant vos tests. À savoir que chaque prestation que nous vous proposons est réalisée sur-mesure, vos demandes « improbables » ne sont pas un frein pour nous.

Automatisation par ci, automatisation par là ! Elle devient vite incontournable dans le monde des tests. Toutefois, il n'est pas évident de savoir par où commencer. Pour vous y aider, voici les étapes qui sont, en pratique, à mettre en place.

À noter : cet article à une valeur « générale », c'est-à-dire que les étapes précises pour automatiser les tests de montée en charge et les tests de non-régression sont différentes.

I. Pourquoi externaliser ?

Avant de se lancer dans l'automatisation de ses tests il faut mener toute une réflexion. Pourquoi ? Quels gains ? Et par qui ? L'équipe en charge de vos tests propose-t-elle un service de qualité, et adapté à vos besoins ? Si vous disposez des compétences en interne pour programmer des logiciels de tests "open source" il peut être pertinent d'évaluer le temps nécessaire à la mise en place des tests avant d'externaliser tout ou partie de la mission.

L'objectif est de choisir votre partenaire en fonction de la typologie de vos tests. Sur des tests "standards" il existe des solutions clés

en main permettant une mise en place rapide. Pour des tests plus techniques, analysant des comportements utilisateurs ou techniques plus poussés ou sur-mesure, il devient intéressant de se tourner vers des éditeurs de logiciels dédiés (comme nous).

En tant qu'éditeur, nous avons la possibilité de réaliser des tests sur mesure qui reproduisent à l'identique de la réalité le comportement de vos internautes sur votre site ou logiciel, ce en fonction de vos besoins.

Ça y est : après vous être renseigné de part et d'autre, vous avez enfin décidé de vous lancer dans l'externalisation de l'automatisation de vos tests. Que ce soit pour de la non-régression ou de la montée en charge, les procédures sont similaires sur de nombreux points.

II. En pratique, comment ça se passe ?

Le prérequis chez CloudNetCare est que, tout au long du processus de mise en place de l'automatisation des tests et même pendant les premiers essais, vous avez la possibilité de mettre fin à notre collaboration sans aucune raison. Chez nous on ne signe pas pour voir

mais pour poursuivre.

Pour prouver notre transparence et notre engagement, nous mettons toujours en place des démonstrations de tests à échelle réduite sur VOTRE contexte afin que vous puissiez voir par vous-même l'efficacité de nos services. En résumé : aucune raison de ne pas essayer.

Externaliser l'automatisation de ses tests n'est pas un long parcours du combattant. En choisissant le bon partenaire, vous avez même la possibilité de vous engager uniquement après avoir vu l'efficacité de la solution. Nous sommes éditeur de notre logiciel de tests CloudNetCare et pouvons avec la même plateforme réaliser vos tests de montée en charge, vos tests de non régression, vos tests en continue ce avec un langage de scripting open source (Selenium) et une plateforme ouverte via des API.

Les scénarios vous appartiennent et vous pouvez les utiliser à votre convenance si vous décidez, malheureusement pour nous, d'arrêter un jour la collaboration.



Merci.

L'univers des tests nous passionne et nous tenions à vous remercier pour avoir téléchargé ce document. Nous espérons avoir répondu à la majorité de vos questions.

Pour des informations complémentaires

Christian Sayegh

CloudNetCare

<https://www.cloudnetcare.fr>

+33 (0)1 84 20 44 13

+33 (0)6 58 26 16 27