



PERFORMANCE ADVISORY COUNCIL

TEST DE PERFORMANCE : LES 20 MEILLEURES APPROCHES 2018

2ème édition **20 experts de 10 pays** **24 heures de live**

35 tasses de café **800 inscrits** **950 autocollants**

En juin dernier, Neotys a réuni 20 experts du monde entier pour partager leur expérience et leurs connaissances concernant les techniques, les défis et les meilleures pratiques associées aux tests de charge pour les entreprises modernes. La réunion a été publiée en ligne en tant qu'événement virtuel.

Les participants ont été initiés à de nouvelles pratiques innovantes dans le domaine des tests de performance. Chaque présentation a fourni une mine d'informations avec des idées et des techniques visant à améliorer les tests de performance.

Ce qui suit est un résumé des présentations faites lors de l'événement. Nous sommes heureux de partager ces informations dans le but de créer une communauté de praticiens du test plus bien informée et plus efficace.

Permettre le partage d'histoires d'utilisateurs et l'apprentissage à l'échelle internationale est essentiel pour nous.



Stijn Schepers

Les tests de performance ne sont pas un jeu de moyennes.

[Stijn Schepers](#) possède plus de 18 ans d'expérience internationale en tant que spécialiste/responsable de tests de performances avec les modèles Agile (Scrum, SAFe) et Waterfall. Il a travaillé en Europe, en Nouvelle-Zélande et en Australie en tant qu'ingénieur de performance pour tester et optimiser des solutions d'entreprises complexes telles qu'Oracle : Financials, Forms, Web Center, Applications CRM, JD Edwards, solutions J2EE, Vignette, SIEBEL, SAP, Cúram, IBM Websphere et Mule. Stijn est un leader visionnaire possédant des connaissances techniques approfondies dans la conception de tests de performances et de leur pilotage dans l'entreprise moderne.

Stijn pense que les tests de performances consistent à examiner « tous les points (données brutes) » et qu'il ne s'agit pas de rechercher des moyennes. Selon lui, ce sont les données brutes qu'il faut mesurer : chaque requête émise par chaque utilisateur, car elle vous apporte les informations dont vous avez besoin pour comprendre le comportement du système. Il souligne que la diversité des informations peut être déterminée à partir des données recueillies et que, lorsque les testeurs se concentrent sur le comportement moyen, ils omettent d'importants problèmes dans le système. Une analyse plus vaste est nécessaire.

Dans sa présentation, Stijn applique sa pensée à une variété d'exemples en utilisant des données exportées à partir de NeoLoad en vue d'une analyse visuelle dans Tableau. En guise de conclusion, il souhaite que son auditoire s'assure que tout outil utilisé pour effectuer des tests permette de produire les données brutes des résultats des tests. Sans cela, les praticiens des tests seront considérablement limités.

Pour vous procurer l'enregistrement de la session complète, [cliquez ici](#). Vous trouverez également le blog de Stijn Schepers sur le sujet [ici](#).



Stephen Townshend

L'art oublié de l'analyse des logs

[Stephen Townsend](#) est un spécialiste des performances logicielles basé à Auckland, en Nouvelle-Zélande. Selon lui, les performances sont plus que de simples tests : il s'agit de risques commerciaux. Ses principales tâches consistent à identifier, hiérarchiser et gérer le risque de performance en fonction de la situation unique du client. Selon lui, il ne s'agit pas seulement de tester la performance mais de limiter les risques commerciaux. Son métier consiste à identifier, hiérarchiser et gérer le risque de performance au cas par cas, en fonction de la situation de chaque client.

Au cours de son exposé, Stephen a parlé d'une approche « diviser pour régner ». Il a montré comment analyser et interpréter différents types de logs, en portant une attention particulière à l'accès aux serveurs Web, à la plateforme et aux journaux des tables de base de données pour comprendre et isoler les goulots d'étranglement des performances.

Stephen a réduit l'analyse des journaux pour assurer la stabilité du système. Il a décrit les moyens permettant d'utiliser l'analyse des journaux pour obtenir des résultats utiles avec des outils modernes et traditionnels, explorant divers types d'informations qui peuvent être extraites des journaux.

- Requête
- Événement
- Débit
- Timing
- Erreur

Stephen a insisté sur l'importance de recueillir auprès des membres de l'équipe des informations sur l'objectif et la signification des journaux standards et des journaux propres à l'entreprise. Selon lui, "vous devez parler aux gens pour comprendre ce que les journaux vous disent."

La présentation comprenait également un examen des quatre directives opérationnelles pour l'analyse des données de journal :

- Taille de l'échantillon : "Assurez-vous de disposer de suffisamment de données pour prendre des décisions éclairées."
- Centiles : "Ne pas examiner uniquement le centile de manière isolée. Vous devez également examiner les données brutes."
- Données brutes : "Les scalaires ne montrent pas très bien la densité (occurrence de valeurs). C'est à ce moment que les moyennes, les centiles et la somme ou les valeurs sont utiles."
- Angles multiples : "Arrêtez d'examiner les données depuis un seul angle. Ne prenez pas simplement une moyenne en pensant que c'est « ma réponse »."

La présentation de Stephen transcende la théorie. Il explique avec succès la meilleure façon de composer un analyseur de journaux dans le langage Python. Enfin, il met son contenu en boucle par une démonstration – comment visualiser les données de test de performance dans [Tableau](#).

Pour vous procurer l'enregistrement de la session complète, cliquez [ici](#). Vous trouverez également le blog de [Stephen Townsend](#) sur le sujet [ici](#).



Srivalli Aparna

Des bases de données bien conçues

[Srivalli Aparna](#) est consultante senior en tests pour The Testing Consultancy, en Nouvelle-Zélande. Au cours des dix dernières années, elle a travaillé pour certains clients basés au Royaume-Uni et en Nouvelle-Zélande, les aidant à améliorer les performances de leurs logiciels. Le point central de la présentation PAC de Srivalli est l'importance de tester autant que possible l'émulation du système de production. Par exemple, l'exécution d'un test de performance sur une base de données ne comportant qu'une centaine d'enregistrements produira des résultats différents de ceux d'une base de données de production contenant des milliers d'enregistrements.

De plus, les données de la base de données de test, ainsi que la structure de la base de données, doivent émuler de près le système de production. Par exemple, assurez-vous que le partage est similaire et que la distance physique entre le système de test client et la base de données de test correspond étroitement au vide physique entre les clients appelants et la base de données en production.

Pour vous procurer l'enregistrement de la session complète, cliquez [ici](#). Vous trouverez également le blog de [Srivalli Aparna](#) sur le sujet [ici](#).



Thomas Rotté

Comment la business intelligence basée sur l'IA contribue à l'automatisation des tests

[Thomas Rotté](#) est directeur technique à Probit. Sa mission consiste à aider les gestionnaires d'applications et les propriétaires d'applications Web à utilisation intensive à réduire les coûts et à améliorer la qualité en créant, en maintenant à jour et en exécutant des scénarios générés automatiquement en fonction du comportement de la production.

Thomas utilise son expérience pour diffuser des solutions tout en travaillant sur un système d'analyse utilisateur combinant les données Dynatrace et l'IA pour obtenir plus d'informations et ainsi fournir une valeur commerciale permettant d'améliorer le mode d'exécution de l'automatisation des tests.

Thomas ne néglige pas la difficulté des tests et rappelle rapidement à quel point cela peut coûter cher, d'où l'utilisation de l'IA comme une approche judicieuse de la réduction des délais et des coûts. Au cours de sa session, il a montré comment utiliser l'IA pour segmenter les activités des utilisateurs en différents clusters comportementaux. Il explique également comment utiliser un outil d'analyse et de surveillance tiers conventionnel, tel que Dynatrace, pour extraire les données absorbées par les algorithmes de l'IA. En conclusion, il passe en revue Probit, le produit développé par son équipe, afin de recueillir des informations sur les tendances comportementales des utilisateurs.

Pour vous procurer l'enregistrement de la session complète, cliquez [ici](#).



Ramya Ramalinga Moorthy

Tester la performance en continu tôt et tester la performance au niveau système

[Ramya Ramalinga Moorthy](#) est une analyste des performances passionnée affichant plus de 15 ans d'expérience dans les tests de performances, la modélisation et la planification de la capacité d'application. Elle est diplômée en génie informatique et titulaire d'une maîtrise en systèmes logiciels (MS) de BITS PILANI, en Inde. Elle dirige une université en ligne (<https://elearning.qaelitesouls.com>) destinée aux ingénieurs des performances. Elle est également lauréate du prix prestigieux Bill Mullen Award 2017 du Computer Measurement Group pour son travail sur la détection des anomalies.

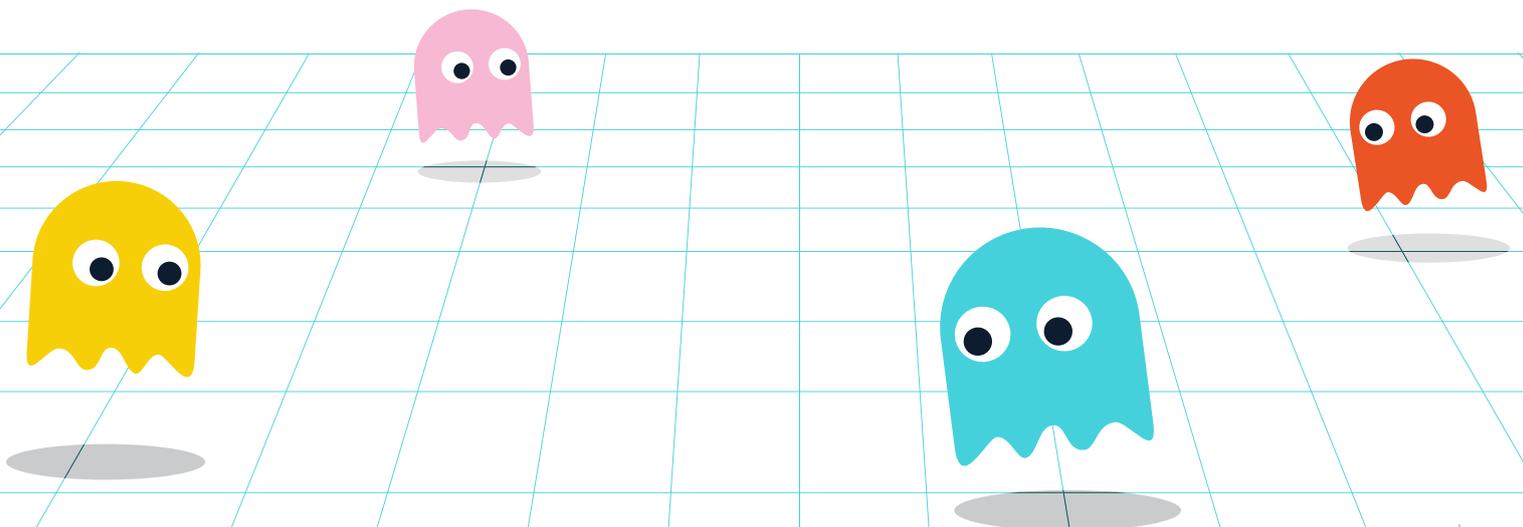
La présentation du PAC de Ramya a mis en évidence la meilleure approche de la stratégie de test des performances. Ce faisant, elle a fourni une liste des choses à faire et à ne pas faire lors de tests dans le cadre d'un processus CI/CD, par opposition aux tests de niveau système.

Elle confirme qu'une bonne stratégie de test des performances inclut le respect des meilleures pratiques et techniques relatives à l'identification et à la vérification des exigences non fonctionnelles (NFR). Il faut également renforcer l'importance des types de tests, des objectifs, de l'identification des cas d'utilisation, des exigences d'infrastructure et de l'analyse des KPI.

Ramya affirme que les tests de performances sont indispensables à la réussite du changement culturel de DevOps. Il existe selon elle sept points à considérer.

1. Maintenir de bons objectifs de test définis en fonction de contrats de niveau de service (SLA) bien définis.
2. Inclure un analyste des performances le plus tôt possible au sein de l'équipe scrum.
3. Définir quand/quoi tester dès le départ – par exemple, tests de performances/de charge CI et tests de performances de niveau système.
4. Documenter les résultats des tests automatisés (critères réussite/échec : UC, temps de réponse).
5. S'assurer que la stratégie de test des performances décrit les éléments à tester en fonction de l'environnement.
6. Allouer suffisamment de temps lors de la planification des tests pour sélectionner le bon outil de test de performance.
7. Provide sufficient time during the planning process to choose the proper Application Performance Monitoring (APM) tool.

Pour vous procurer l'enregistrement de la session complète, cliquez [ici](#). Vous trouverez également le blog de [Ramya Ramalinga Moorthy](#) sur le sujet [ici](#).





Amir Rozenberg

Test de performance en amont (Shift Left)

Durant ce PAC, [Amir Rozenberg](#) était Director of Product Management chez Perfecto. Il possède une vaste expérience dans l'industrie numérique, ainsi qu'une expertise dans des domaines incluant le développement d'applications, les tests, la livraison et la surveillance.

Amir a mené avec succès des collaborations avec, entre autres, Microsoft, HP et bien sûr Neotys afin d'optimiser la fourniture de valeur aux clients. Il a également fondé Adva Mobile, une start-up de marketing mobile destinée directement aux fans.

Amir affirme qu'il existe une erreur dans la conception classique des tests non fonctionnels, tels que les tests de performance, d'accessibilité et de sécurité. On pense traditionnellement qu'il est impossible de réaliser des tests non fonctionnels lors d'une session Agile. Ce n'est pas le cas. La réalité est que certaines parties des tests de performance peuvent être dans la session, automatisées de manière à ce que les défauts de réactivité soient exploités dans les minutes qui suivent la validation du code, indépendamment de la plate-forme.

Dans sa présentation PAC, il définit la transformation numérique comme « une extension des interfaces numériques qui touchent nos utilisateurs finaux, qu'il s'agisse d'un schéma B to B ou B to C, sur un nombre accru et étendu d'écrans/de types (par exemple, téléphones mobiles, téléphones, tablettes et objets intermédiaires tels que les dispositifs IoD et les écrans de véhicules motorisés).

Dans son analyse, Amir souligne l'expansion des interfaces et des expériences numériques et la pression accrue qu'elles exercent sur les entreprises afin que ces dernières commercialisent rapidement de nouvelles fonctionnalités et différencient les nouvelles interfaces et expériences sur le marché. De plus, maintenant que les clients gagnent en puissance, les tests de performances doivent mettre l'accent sur les tests côté client de la même manière que sur les tests côté serveur. De ce fait, les entreprises se tournent vers une démarche Agile pour augmenter leur productivité et réduire leurs délais de mise sur le marché.

Pour Amir, adopter une démarche Agile signifie mener l'activité des tests, y compris les tests de performances, plus tôt dans le cycle de développement – l'intention même du mouvement Shift Left.

La présentation d'Amir comprend des réponses aux questions concernant la viabilité de la mise en œuvre de tests non fonctionnels au début du cycle de développement. Selon lui, certaines choses ne peuvent pas être déplacées dans la session. Par exemple, les tests de performances manuels intensifs nécessitent des systèmes entièrement fonctionnels/intégrés. Cependant, les tests de réactivité, d'accessibilité et de sécurité peuvent être menés en Shift Left. Il souligne l'importance de mettre en place une journalisation appropriée tout au long du processus et une surveillance adéquate de niveau bas des activités de la machine et du réseau.

La définition intrinsèque est également importante dans la mesure où les tests de réactivité doivent être effectués sur les clients tels qu'ils sont dans le « monde réel ». Une définition intrinsèque bien définie décrit un environnement de périphérique client dans lequel les applications autres que celle testée s'exécutent en arrière-plan (Pandora étant utilisé simultanément avec l'application testée). Comme Amir l'indique, tous les tests doivent être exécutés dans un environnement imitant le véritable état de la situation de production.

Pour vous procurer l'enregistrement de la session complète, cliquez [ici](#). Vous trouverez également le blog d'[Amir Rozenberg](#) sur le sujet [ici](#).



Thomas Steinmaurer

Validation continue des performances de cluster @ Dynatrace

[Thomas Steinmaurer](#) apporte à son poste actuel plus de 15 ans d'expérience dans le domaine des bases de données relationnelles. Impliqué dès le premier jour dans la construction de la nouvelle solution APM SaaS et On-Premise nouvelle génération (principalement responsable du back-end Cassandra), il se concentre maintenant sur les aspects liés aux tests de charge, aux performances et à l'évolutivité du cluster Dynatrace sur une base quotidienne.

Dans sa présentation PAC, Thomas souligne que par le passé, Dynatrace livrait deux nouvelles versions par an. Aujourd'hui, confirme-t-il, Dynatrace est réputé pour avoir produit 26 versions au cours d'une session de deux semaines ! Il résume que deux facteurs ont permis à Dynatrace d'accélérer la commercialisation.

1. Adoption d'un process Agile
2. Ajustement de la philosophie de test

Partant de là, Thomas examine de près l'architecture de Dynatrace, décrivant l'approche de l'entreprise, la livraison continue et le feed-back (CDF).

Dynatrace adopte une approche de la commercialisation en trois phases.

1. Développement
2. Acceptation
3. Production
 - a. Remarques :
 - i. En règle générale, le développement dure deux semaines, tout comme l'acceptation.
 - ii. Le développement et l'acceptation impliquent des tests rigoureux de performances/charge.
 1. Les tests de développement sont effectués quotidiennement.
 2. Les tests d'acceptation ont lieu tous les deux jours. Une surveillance complète est intégrée aux deux phases.
 - a. Il en résulte des mises en production bihebdomadaires.

Dynatrace utilise largement l'émulation lors des tests, notamment pour émuler la charge client. Les émulations vont de petits locataires à des locataires importants. Maintenir les émulations proches du monde réel fournit la fiabilité dont Dynatrace a besoin pour avoir confiance dans ses pratiques de test. Cependant, en matière de test de régression, Dynatrace utilise une charge déterministe, inchangée.

Dynatrace n'utilise pas de solution de test de charge standard telle que JMeter. À la place, l'entreprise a créé sa propre solution pour simuler avec précision la charge alignée sur ses agents et les interactions des utilisateurs.

Dynatrace déclenche manuellement les mises à jour des clusters à grande échelle pour permettre aux utilisateurs de surveiller la condition de l'activité en aval afin d'y rechercher les anomalies opérationnelles avant de procéder aux versions. Néanmoins, l'entreprise rencontre des problèmes de production. La leçon tirée est que même s'il est essentiel de procéder à une émulation appropriée et à des tests complets en aval avant de passer à la production, des problèmes se produiront dans l'environnement de production. Avant tout, recommande-t-il, les entreprises doivent rester vigilantes.

Pour vous procurer l'enregistrement de la session complète, cliquez [ici](#).



Bruno Da Silva

Approche innovante des tests de performance continus

[Bruno Da Silva](#) a rejoint Leanovia après avoir terminé ses études à l'UTC de Compiègne, en France. Il est passionné par les tests de performances informatiques, et peut s'y intéresser tous les jours grâce à son poste actuel. Bruno est certifié sur les technologies NeoLoad et Dynatrace.

La présentation PAC de Bruno met en évidence les trois moteurs fondamentaux de la performance : application, charge et ressources. L'application est la logique qui détermine le comportement. La charge quant à elle, correspond aux utilisateurs (à la fois des machines et des êtres humains) qui interagissent avec le système. Les ressources sont le matériel et le réseau qui prennent en charge le système.

Selon lui, l'objectif des tests de performances continus est de contrôler la robustesse du code. Il rappelle les nombreux avantages des tests continus et leur contribution à l'amélioration des performances des applications.

- Réduction des coûts
- Atténuation des risques
- Amélioration du code
- Automatisation
- Surveillance

Bruno ajoute également certaines conditions préalables aux tests de performances continus :

- Définir au préalable les exigences de performances
- Assurer la testabilité du logiciel
- Disposer d'un pipeline de type production
- Créer un plan efficace

Il décrit la manière dont les outils peuvent être utilisés pour alléger le travail inutile pendant un test de performance continu, en passant en revue les outils pouvant être appliqués :

- Industrialisation CI/CD avec le pipeline Jenkins (Blue Ocean, Git, SonarQube, Nexus)
- Architecture conteneurisée avec Openshift
- Monitoring en utilisant Dynatrace
- Test de performance avec NeoLoad

La principale préoccupation de Bruno est qu'il ne faut pas hésiter à tirer parti des fonctionnalités de chaque outil (plug-ins, API et intelligence artificielle) pour accélérer le processus de validation des performances.

Pour vous procurer l'enregistrement de la session complète, cliquez [ici](#).



Bruno Audoux

Scripts autonomes

[Bruno Audoux](#) écrit du code depuis l'âge de dix ans. Il a poursuivi sa passion pour l'informatique en travaillant dans les données fluides et l'architecture informatique de Control Data, en France. Il a créé sa société, Bench, en 2002 pour créer des produits destinés aux services Web fiscaux français. En 2010, il crée la société Aerow Performance (à présent WAPSI), où il occupe désormais le poste de responsable technique.

La présentation de Bruno est centrée sur le partage d'« astuces » en matière de scripts. Son message principal est que les scripts modernes pourraient nécessiter une durée de conservation supérieure à une session. Sa recommandation : utilisez de petits scripts pouvant être agrégés pour répondre aux besoins actuels. Beaucoup plus facile à gérer qu'un script vaste, un petit script doit se limiter à une seule transaction (par exemple, un processus de connexion).

Il poursuit avec une démonstration de certains aspects des scripts dans l'entreprise moderne, en proposant davantage d'astuces pour faciliter les échanges de données entre scripts via un fichier CSV, une base de données et une file d'attente partagée.

Pour vous procurer l'enregistrement de la session complète, cliquez [ici](#).





Helen Bally

Temps de réponse inférieur à 1 seconde en gestion d'entrepôt

[Helen Bally](#) a commencé sa carrière en informatique comme première stagiaire SAP chez Price Waterhouse à Londres. Les déploiements SAP l'ont menée en Irlande, à Houston et à Bruxelles, où elle a dirigé des projets mondiaux de technologie SAP dans le secteur du pétrole et du gaz. Arrivée en Suisse en 2003, Helen a rejoint Itecor et s'est intéressée au monde des tests. Aujourd'hui, elle dirige le groupe Itecor Enterprise Testing et a aidé de nombreuses entreprises suisses et internationales dans leur stratégie de test, l'automatisation des tests et les tests de performances. Elle adore les projets techniques complexes où elle peut générer des résultats en associant technologie, personnel et processus.

Dans son exposé, Helen s'intéresse principalement à la manière dont une importante société de fabrication et de distribution a réalisé son objectif consistant à atteindre un temps de réponse inférieur à la seconde pour les exploitants d'entrepôts. Son exposé montre comment l'entreprise a mis en œuvre un système de gestion d'entrepôt d'entreprise SAP (SAP EWM) dans l'espoir de garantir que toutes les fonctions d'opérateur d'entrepôt atteindraient l'objectif d'un temps de réponse inférieur à la seconde.

Alors que la plupart des tests de performances concernent les utilisateurs simultanés, le cas d'Helen est différent. Elle se concentre sur un utilisateur unique avec une micro-analyse de chaque clic d'utilisateur dans des conditions réelles.

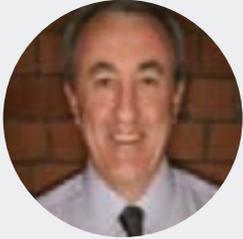
Sa présentation PAC inclut un examen des défis uniques associés au scénario de gestion d'entrepôt. Par exemple, l'état de l'entrepôt (notamment s'il est plein ou pas) et son impact sur les performances.

L'équipe d'Helen adopte une approche innovante en matière de tests. Elle a isolé l'environnement de test et mis en œuvre l'utilisation de services virtualisés à l'aide de la technologie SAP RFC ([Remote Function Call](#)) dans des scripts de test automatisés (appels de fonction à distance). Le script de test a relevé des traces pour analyser les performances au niveau du code. En utilisant son cadre de test reproductible, l'équipe a pu prouver que les performances s'étaient améliorées.

L'équipe a tiré des enseignements précieux de son expérience :

- La configuration/l'isolation du service testé compte
- La réussite des tests dépend de spécifications efficaces
- L'environnement de test doit être complètement indépendant de l'activité en continu de l'entrepôt afin de ne pas affecter les activités quotidiennes
- Vous devez vous assurer que les articles que vous stockez dans l'entrepôt sont représentatifs de l'inventaire réel
- Le tracking, même au sein d'un ERP, est essentiel
- La portée des tests doit surveiller les performances pour détecter la régression et la comparer à une valeur de base
- Les tâches de test de performances doivent reproduire précisément l'activité du monde réel

Pour vous procurer l'enregistrement de la session complète, cliquez [ici](#).



Ian Molyneaux

Maintien des performances des applications dans le cadre de la transformation numérique

[Ian Molyneaux](#) est consultant senior chez Infuse. C'est purement par hasard qu'il s'est retrouvé dans l'informatique après avoir postulé à un poste d'opérateur informatique junior qu'il décrit comme « enrichissant » au milieu des années 70. 36 ans plus tard, Ian est toujours dans l'informatique, tirant parti de toutes ses expériences et avec un intérêt particulier pour les performances des applications.

Ian est un auteur publié dans le domaine des tests de performances et les disciplines associées (The Art of Application Performance Testing, O'Reilly 2009 et 2014). Il se concentre actuellement sur le conseil et l'assistance proposés sur des thèmes tels que l'adoption d'une démarche DevOps, les tests de performances et le dépannage, ainsi que la gestion des performances des applications (APM). Intervenant régulier de l'industrie, Ian travaille également en tant que conseiller auprès de nombreux fournisseurs d'outils informatiques.

Sa conférence PAC présente la transformation numérique comme une évolution en trois parties.

1. Une tendance chez les entreprises à passer au cloud pour en tirer des avantages en termes de coûts et de performances.
2. Une nouvelle architecture de solution pour s'adapter à un nouvel environnement informatique basé sur le cloud.
3. L'accent mis sur l'expérience utilisateur avec des technologies de support qui permettent la migration vers des environnements distribués basés sur le cloud.
 - a. Remarques :
 - i. Les appareils mobiles dont le fonctionnement repose sur l'accès au cloud apportent une toute nouvelle dimension à l'expérience utilisateur.

Aujourd'hui, en raison de l'Internet des objets (IdO, IoT), les appareils mobiles se déplacent au-delà du téléphone portable et de la tablette numérique. De nouveaux types d'interaction font désormais partie de l'expérience utilisateur (par exemple, l'utilisation d'un appareil basé sur la voix tel qu'Alexa). La dynamique de l'expérience utilisateur est intrinsèquement différente. De ce fait, l'approche des tests de performances doit être modifiée. Selon Ian, "l'expérience utilisateur est une partie immense de ce processus de transformation numérique."

Ian a également souligné que chaque partie de cette évolution comporte son propre ensemble de risques de performance. Par conséquent, lorsque vous envisagez la façon d'aborder les tests de performances d'une transformation numérique, il est préférable de concentrer les activités de test sur une seule application dans la transformation numérique. « N'ayez pas la folie des grandeurs. » Limitez la portée du risque. Définissez également une base de données pour les performances des applications pré/post-comparaison dans la migration vers le cloud. Enfin, lors de la transformation numérique vers le cloud, de nombreuses entreprises confient l'exploitation et la maintenance d'une application à un tiers. Pour assurer leur sécurité, elles doivent disposer d'un plan de reprise après sinistre si les choses ne se passent pas comme prévu.

Concernant la réarchitecture, Ian a observé que les technologies cloud sont soumises à un degré élevé de choix et de changement. « Une entreprise peut adopter certaines approches pour concevoir une architecture SOA (Service Oriented Architecture), et certains produits peuvent être appliqués à une conception donnée. » Par exemple, pour l'orchestration de conteneur, il existe Kubernetes, Mesos ou Docker Swarm. En outre, les technologies de stockage de données d'appui peuvent varier des bases de données traditionnelles aux bases de données basées sur NoSql. Disposer dès le départ d'une approche des tests de performances planifiée/segmentée convenablement exécutée et faisant l'objet de rapports continus est essentiel lors du déploiement de systèmes dans une transformation numérique.

Pour vous procurer l'enregistrement de la session complète, cliquez [ici](#).



Brian Brumfield

L'émergence de sites Web pour clients lourds : comment les identifier et les tester en termes de performances ?

[Brian Brumfield](#) est directeur de l'ingénierie de la performance et des services managés chez Foulk Consulting. Il est spécialiste des solutions APM et d'ingénierie de la performance à Mercury. Il travaille également avec ATS/VIP, J9 Technologies, SOASTA, Keynote, Dynatrace et Foulk Consulting.

Dans l'ensemble, nous voyons davantage d'approches de conception sur le Web basées sur le développement de clients lourds. Aujourd'hui, bon nombre de sites Web se comportent comme des applications mobiles. En tant que tels, ils présentent de nombreux défis pour les méthodes traditionnelles d'ingénierie de la performance. Alors, comment identifier et aborder ces sites ?

Dans le passé, de nombreux développeurs intégraient des sites Web complets dans des environnements mobiles tels que Cordova, Sencha, etc. Nous voyons maintenant des développeurs prendre des architectures d'applications mobiles et les transférer vers le Web, en les plaçant dans le navigateur. Naissance d'un client Web "lourd".

Comme Brian l'indique dans sa présentation PAC, "les développeurs n'ont plus peur de placer du contenu dans le navigateur. Ils chargent sans retenue le navigateur, en mégaoctets." Ce défi soulève une question fondamentale que Brian pose, "Comment testez-vous cela ?"

Brian insiste sur le fait que ces nouveaux sites Web clients lourds présentent des défis uniques et qu'il est vital de pouvoir accéder à l'API à faible latence. En outre, il y a beaucoup plus d'activités de rendu générées par la logique métier de traitement de code JavaScript côté client. De nouvelles techniques de test de performance sont nécessaires pour garantir la réactivité et la précision en temps voulu de ces nouveaux types de sites Web lourds.

Brian montre comment utiliser un service d'optimisation de la vitesse et des performances de sites Web, tel que GTmetrix, pour collecter des informations sur les performances des applications. Il souligne que les sites peu performants ne sont pas uniques. En particulier, de nombreuses applications commerciales chargeront des systèmes d'entreprise complets dans le navigateur. Cela entraîne des frais généraux importants et a un impact en termes de performances des tests de l'application. Il rappelle des situations dans lesquelles il espérait pouvoir créer un millier d'utilisateurs virtuels à utiliser dans des scénarios de test, mais était réduit à n'en utiliser que soixante-dix en raison des frais généraux imposés par le code JavaScript côté client.

Les clients Web lourds posent également un problème en raison d'un traitement excessif côté client difficile à détecter. Dans le passé, une page Web typique pouvait faire un rappel détectable à un serveur chaque fois que l'utilisateur cliquait sur un bouton. Une application client Web lourde peut entraîner un traitement qui monopolise les ressources côté client, telles que l'utilisation du processeur, ce qui affecte les performances.

Ces types de goulots d'étranglement des performances sont indétectables à l'aide des outils traditionnels de surveillance des requêtes/réponses.

Brian recommande que vous adoptiez une approche taxonomique des tests de performances dans laquelle toutes les analyses sont considérées « du seul point de vue de la charge de travail, du client au serveur. » Par conséquent, le concepteur a pour tâche de rassembler et d'organiser toutes les données pertinentes pour l'interaction entre l'utilisateur et l'application. Cela signifie aller au-delà de la mesure des temps de requête/réponse seuls. Les testeurs doivent non seulement être en mesure de signaler le débit indiquant l'origine des données (par exemple, livrées par la périphérie ou par le serveur), mais également examiner la charge imposée par l'application sur l'ordinateur côté client hébergeant le navigateur. Différents outils seront nécessaires pour arriver à une compréhension complète des performances globales. Il incombe au concepteur du test de rendre cela possible en ce qui concerne la taxonomie actuelle.

Il souligne également que les outils de test doivent être modernisés pour fournir des capacités de test plus complètes. Les scripts doivent aller au-delà des scénarios à usage unique. La réutilisation est nécessaire pour réduire les coûts tout en répondant aux demandes de tests en cours. Comme Brian le dit, « pour aborder ces problèmes, une perspective de charge de travail taxonomique est probablement la meilleure solution. Il faudra juste que ce soit le cas (de cette façon) jusqu'à ce que nous ayons des tests en masse évolutifs basés sur le navigateur, où nous pourrions exécuter des milliers d'utilisateurs virtuels. C'est simplement une réalité aujourd'hui. Ce sera. Quelqu'un va résoudre ce problème. Espérons que ce soit Neotys. »

Pour vous procurer l'enregistrement de la session complète, cliquez [ici](#).



Todd DeCapua

Télémétrie : l'ingrédient essentiel du succès avec Agile, DevOps et SRE

[Todd DeCapua](#) est Technology Evangelist, directeur logiciel passionné et chef d'entreprise. Certains de ses postes/titres incluent : directeur exécutif de Global Technology au sein de [JPMorgan Chase & Co.](#) directeur senior de la technologie et de l'innovation produit à [CSC](#), responsable en chef de la technologie à [Hewlett-Packard Enterprise](#), co-fondateur de [TechBeacon.com](#), vice-président de l'innovation et comité de direction pour la stratégie à [Vivit Worldwide](#), et comité de direction indépendant à [Apposite Technologies](#). Il est également un auteur et contributeur en ligne actif de l'ouvrage publié par O'Reilly, Effective Performance Engineering.

Selon Todd, alors que les entreprises continuent de se concentrer sur la transformation vers les sensibilités Agile, DevOps et Site Reliability Engineering (SRE), de nombreuses équipes sont mises au défi de réussir. Pour relever les défis, elles doivent utiliser la télémétrie afin de permettre à l'entreprise de réaliser les objectifs de rapidité, de qualité et d'automatisation nécessaires à de bons résultats. Selon lui, « la télémétrie devient la clé, la colonne vertébrale et l'élément fondamental que nous devons tous voir. »

Dans sa présentation PAC, Todd indique que quatre problèmes s'appliquent à la télémétrie :

1. La vitesse accrue
2. La réduction du temps avant réparation (MTR)
3. La collaboration accrue au sein de l'entreprise
4. La livraison de la plus grande valeur au client

Pour Todd, un système de télémétrie est composé de trois éléments : les collecteurs, les données et la visualisation. Un collecteur est le moyen par lequel les données et les informations sont collectées. il peut consister en une performance d'application ou en un outil de surveillance de la sécurité.

Les données sont les informations recueillies par le collecteur. Il souligne que, compte tenu du volume de données qu'un collecteur peut récupérer, les entreprises doivent avoir une stratégie de collecte des données et un moyen de comprendre et de structurer les données collectées.

La visualisation est le moyen par lequel les utilisateurs interagissent avec les données collectées. Déterminer la visualisation est essentiel pour apporter de la valeur aux utilisateurs et aux clients. Un exemple de définition d'une stratégie de visualisation consiste à concevoir et à mettre en œuvre un tableau de bord qui capture les différents points de données présentant un intérêt pour l'utilisateur.

La présentation de Todd explore les cinq types de collecteurs que les ingénieurs doivent utiliser tout au long du développement des applications, de l'ingénierie de l'architecture et de la gestion de la production pour atteindre les objectifs que les entreprises modernes orientées sur les performances exigent des systèmes de télémétrie.

- Performance
- Capacité
- Monitoring
- Incident de production
- Sécurité

Todd mentionne que l'utilisation de collecteurs en streaming se développe dans l'environnement de la télémétrie. Il souligne aussi que davantage de systèmes de télémétrie intègrent l'apprentissage par machine, l'intelligence artificielle et l'analyse prédictive dans leur fonctionnement en temps réel.

Todd estime que les systèmes télémétriques doivent être conçus en fonction des préférences de l'utilisateur. Comme le dit Todd, « Ce [tableau de bord] est-il ce que recherchent votre clientèle et votre entreprise ? » Todd recommande de prendre le temps de comprendre comment les alertes doivent être générées et comment les informations doivent être affichées dans un tableau de bord donné.

En examinant la nature et l'application de la télémétrie aux tests de performances critiques, Todd offre cette sagesse : « Où s'inscrit la télémétrie ? Où commence-t-elle ? Où finit-elle ? Elle est partout. Elle englobe tous ces éléments (planifier, coder, construire, tester, publier, déployer, utiliser, mesurer). Nous avons tous vu les outils individuels. Nous savons que ce sont les données qui proviennent d'eux. Comment ce fait-il que vous puissiez instrumenter chacune de ces fonctionnalités, en fournissant ces données presque en temps réel, de sorte que vous

puissiez effectuer différents types de corrélations pour aboutir à la télémétrie en tant que résultats afin de permettre à vos clients et à votre entreprise de générer la réussite ? »

Todd recommande de lire les ouvrages suivants pour en savoir plus sur DevOps dans l'ensemble et sur la télémétrie en particulier.

- [Agile Software Development with Scrum](#)
- [The Phoenix Project](#)
- [Site Reliability Engineering](#)

Pour vous procurer l'enregistrement de la session complète, cliquez [ici](#). Vous trouverez également le blog de [Todd De Capua](#) sur le sujet [ici](#).



Gayatree Nalwadad

Life Shift in Performance Testing

[Gayatree Nalwadad](#) a dix ans d'expérience en tant qu'ingénieur principal d'assurance qualité à Fidelity Investments. Elle est responsable de toute la stratégie et l'exécution des tests de performances pour son unité commerciale. Avant Fidelity Investments, Gayatree a travaillé chez Infosys Technology Ltd (Inde) en tant qu'ingénieur des tests de performance, au service de clients tels que BofA et Adidas. Elle a également obtenu les certifications LoadRunner et Software Testing de SQE Training (une société TechWell).

Dans sa présentation PAC, Gayatree décrit son approche des tests de performances dans un monde Agile. Elle explore également les points sensibles associés à la transition de Waterfall à Agile.

L'introduction de sa présentation : la clé d'une ingénierie de la performance efficace explique qu'il faut toujours rester en phase avec les besoins et les priorités des utilisateurs/cadres de l'entreprise. Elle décrit les six étapes de l'ingénierie de la performance proactive : (1) Collecte des exigences non fonctionnelles, (2) conception, (3) développement, (4) exécution des tests, (5) analyse et reporting.

Gayatree a la ferme conviction que tous les membres de l'équipe de développement doivent avoir reçu une formation adéquate sur les principes de base des concepts, des pratiques et de la terminologie relatifs aux tests de performances. Elle se donne également pour mission de s'assurer que les ingénieurs de la performance adaptent leurs techniques de test au processus de développement utilisé.

En ce qui concerne les tests de performance dans un environnement Agile, Gayatree aime ajuster les étapes des tests de performances proactifs dans le modèle d'exécution de la session. Les exigences non fonctionnelles et la conception de la charge de travail ont lieu pendant la planification de la mission. La configuration des données de test, la rédaction des scripts de performances et le test de développement indépendant (DIT) ont lieu pendant la session. Les tests de performance, de capacité, de stress, de pic, de longévité et de sortie ont lieu lors de la rétrospective post-session. La collecte des données de surveillance et de capacité des applications est effectuée au moment du déploiement. Les données recueillies feront l'objet d'une analyse avancée des performances.

Gayatree explique également comment procéder au dépannage et à l'identification des goulots d'étranglement, en précisant qu'elle s'assure généralement que les outils qu'elle utilise fonctionnent comme prévu. Il n'est pas inhabituel qu'elle inspecte de près les scripts de performance. Elle se concentre ensuite sur les données. Enfin, pour s'assurer que l'application est opérationnelle, les serveurs et la base de données se comportent correctement.

Pour Gayatree, un ingénieur de la performance performant porte plusieurs casquettes, de l'investigateur du dépannage à l'ingénieur système, selon les aspects sur lesquels il se concentre durant la session.

Pour vous procurer l'enregistrement de la session complète, cliquez [ici](#). Vous trouverez également le blog de [Gayatree Nalwada](#) sur le sujet [ici](#).



Alexander Podelko

Tests de performance continus : mythes et réalités

[Alex Podelko](#) se spécialise dans la performance depuis 1997, notamment comme ingénieur et architecte de la performance pour plusieurs entreprises. Il est actuellement membre consultant du personnel technique d'Oracle, responsable des tests et de l'optimisation des performances des produits de gestion des performances d'entreprise et de Business Intelligence (Hyperion).

Les liens et les documents liés à la performance d'Alexander (y compris des articles et des présentations récents) sont disponibles [ici](#) ; son [blog](#) et compte Twitter sous [@apodelko](#). Il est également directeur du Computer Measurement Group (CMG), une organisation de professionnels de la planification des performances et de la capacité.

Si vous le lui demandez, Alexander vous dira qu'en matière de test de performances, « le contexte est roi ! » Il se souviendra aussi probablement très bien d'hier...

L'ancienne méthode d'exécution des tests de performance de charge est généralement centrée sur la cascade.

Les tests de performance, ne serait-ce que pour l'entreprise en raison des dépenses, ont été effectués juste avant le déploiement. L'enregistrement de l'activité de test a été effectué à un niveau bas, généralement au niveau du protocole. L'environnement de test s'apparentait davantage à un laboratoire qu'à un environnement de production réel. Les véritables tests consistaient à vérifier la case à cocher d'un cahier des charges ou d'une partie d'un accord de niveau de service définissant la réussite d'une application.

Comme il se souvient de la façon dont les choses se passaient, Alexander le confirmera : « Nous n'en sommes plus là ! » Maintenant, nous avons différents types de test de performance en jeu, et pas seulement en tant que dernière étape de pré-production. »

Au cours de sa présentation PAC, il souligne ce qu'il considère comme un paradoxe apparent. Bien que la portée et la pratique des tests de performances se soient améliorées sous Agile et le processus CI/CD, le même problème persiste. Lorsqu'il s'agit de choisir entre tester la fonctionnalité ou les performances, la priorité est donnée à la fonctionnalité. Cette attitude doit changer.

[Alexander](#) décrit ensuite l'état actuel des tests de performance dans les environnements Agile et CI/CD. Il évoque les éléments à prendre en compte lors de l'intégration des tests automatisés : « En règle générale, les API sont plus stables et se prêtent bien à l'automatisation. » Au cours de sa discussion, il évalue ensuite la valeur de l'automatisation par rapport au coût, en résumant qu'une « automatisation complète » n'est ni faisable, ni recommandée. Il faut plutôt arriver à un équilibre hybride entre tests manuels et tests automatisés en fonction des besoins.

Pour vous procurer l'enregistrement de la session complète, cliquez [ici](#). Vous trouverez également le blog d'[Alexander Podelko](#) sur le sujet [ici](#).



Mark Tomlinson

Cognitive Biases in Performance, Wut?

[Mark Tomlinson](#) est actuellement architecte de la performance à grande échelle et également producteur du célèbre podcast sur la performance, [PerfBytes](#). Sa carrière de testeur a débuté en 1992 avec un test complet de deux ans pour un système de transport essentiel à la vie – un projet qui a suscité son intérêt pour les tests de logiciels, l'assurance qualité et l'automatisation des tests. Mark a travaillé à Microsoft, Hewlett-Packard et PayPal, où il

a accumulé une grande expérience dans les tests de scénarios réels de systèmes vastes et complexes. Mark est considéré comme un leader visionnaire en matière d'automatisation des tests de logiciels, doté d'une expertise spécifique de la performance.

Selon Mark, l'ingénierie de la performance est plus vaste qu'on ne le pense généralement. Comme il l'indique, « Pour la plupart des pratiques actuelles en ingénierie de la performance, l'accent est mis sur les mesures objectives pour le système testé : temps de réponse, CPU, disque, mémoire, réseau, profondeur de la file d'attente. Nous collectons les données et analysons les chiffres. Nous projetons la capacité future. Nous dormons bien. Mais, que se passe-t-il si les mesures objectives limitent potentiellement notre compréhension de la performance ? Et si ces algorithmes élaborés nous avaient convaincus que nous avions de la valeur ou que nous avions raison ? Et s'il y a plus à dire dans l'histoire ? »

Le contenu PAC de Mark vise principalement à raconter cette histoire inédite de préjugés cognitifs dans les tests de performance et leur impact sur la capacité d'une personne à étendre ses performances au-delà des métriques considérées.

Mark décrit les quatre quadrants du codex des préjugés cognitifs :

De quoi il faut se souvenir | Trop d'informations

-----+-----

Besoin d'agir vite | Sens insuffisant

Mark décrit certains types de préjugés cognitifs liés aux quatre quadrants du codex des préjugés cognitifs et aux tests de performance.

- **L'effet de fondu** : Nous avons conçu des processus et des méthodes exceptionnels . Nous n'avons eu aucun problème pendant plusieurs années. Alors nous nous sommes demandés, faut-il arrêter les tests ? Ce n'est pas utile.
- **L'effet de modalité** : Nous ne réagissons que lorsque nous voyons quelque chose virer au rouge
- **Négliger la durée** : Comme les problèmes sont fréquents, nous commençons à nous habituer à avoir un comportement familier, et nous ne faisons donc pas attention.
- **Préjugés en matière de distinction** : À mesure que nous comparons les métriques dans le temps, les occurrences fréquentes (superpositions) deviennent évidentes dans notre esprit.
- **Perception sélective** : Nous ne prêtons attention qu'aux choses qui nous intéressent, par exemple le temps de réponse lent.
- **Exagération** : Faire grand cas d'une chose insignifiante pour justifier notre travail.
- **Illusion sur l'argent** : Le coût d'un dysfonctionnement est plus élevé qu'on ne le pense
- **Illusion inverse sur l'argent** : Appliquer une technologie ou technique non pertinente au coût de mise en œuvre d'une situation ne coûte pas cher
- **Cause de connaissance** : Nous supposons que tout le monde comprend les idées et le langage qui nous sont familiers (notre jargon)

Un autre sujet abordé par Mark concerne Cynefin. Selon Wikipedia, Cynefin propose cinq contextes ou « domaines » de prise de décision (évident que l'on nommait " simple" jusqu'en 2014, compliqué, complexe, chaotique et désordonné), qui aident les responsables à identifier leur perception des situations et donnent un sens à leur propre comportement et à celui des autres. Le cadre s'appuie sur la recherche sur la théorie des systèmes, la théorie de la complexité, la théorie du réseau et les théories de l'apprentissage.

Mark détaille davantage les descriptions des domaines de prise de décision :

- **Évident** : étroitement contenu, "meilleure pratique"
- **Compliqué** : étroitement lié, "bonne pratique"
- **Complexe** : faiblement lié, "pratique émergente"
- **Chaotique** : manque de contrainte, "pratique nouvelle, non testée"
- **Désordre** : aucun des cas ci-dessus

Cynefin présente l'avantage que la prise de décision basée sur les risques vous aide à déterminer la bonne décision. La stratégie de test et l'application des solutions sont les mieux considérées avec Cynefin.

Globalement, les testeurs de performances doivent décider de ce qui doit être retenu, de ce qui est important et de la manière d'éviter une surcharge d'informations. Toutes ces considérations font l'objet de préjugés cognitifs.

Pour vous procurer l'enregistrement de la session complète, cliquez [ici](#).



Zak Cole

Ingénierie de performance de la blockchain

[Zak Cole](#) Zak Cole est directeur technique de Whiteblock, une société d'ingénierie de la performance orientée sur les technologies de blockchain et de registre distribué. Chez Whiteblock, Zak a mis au point une plate-forme de test de la blockchain, la première de son genre, pour aider les équipes de développement à construire des systèmes de blockchain convaincants et évolutifs.

En tant qu'investisseur précoce dans Bitcoin et Ethereum, il est impliqué dans le domaine de la blockchain depuis 2013, ayant même été ingénieur réseau au sein du United States Marine Corps. Il a également signé un contrat de développeur front-end avec Google et a travaillé en tant que professionnel de la sécurité des systèmes d'information et chef de produit technique à Apposite Technologies.

Bien que le domaine de la blockchain en soit encore à ses balbutiements, la technologie du registre distribué laisse présager des solutions précieuses et uniques pour une variété de problèmes mondiaux. Cependant, pour assurer une véritable adoption par la majorité, la communauté du développement de la blockchain s'emploie d'arrache-pied à surmonter plusieurs problèmes de performances clés qui entravent ces efforts d'adoption.

Au cours de sa présentation PAC, Zak résume les problèmes de performance (et les solutions) liés à la technologie de la blockchain. Sa présentation appelle à l'action pour que la communauté de la performance apporte sa contribution, en particulier les personnes ayant une expérience en entreprise.

Les participants découvrent également certains résultats associés à une série de tests récemment effectués par Zak à Whiteblock, un bref aperçu des méthodologies appliquées.

Zak termine par un examen de la technologie sous-jacente à la blockchain, abordant des produits spécifiques tels que [Hyperledger](#), [Ethereum](#) et [Plasma](#). Il décrit la façon dont la blockchain est utilisée pour prendre en charge la crypto-monnaie, en plongeant dans la dynamique de l'exploitation de bitcoins dans le cadre de la blockchain, de son interopérabilité et de sa sécurité.

Pour vous procurer l'enregistrement de la session complète, cliquez [ici](#).





Wilson Mar

Les leçons apprises en installant le Client MacOS

[Wilson Mar](#) est Principal DevOps Performance Evangelist chez JetBloom, où il se concentre sur la proactivité face aux problèmes de performances/capacité tout en mettant en œuvre des logiciels complexes dans les entreprises et les organisations entrepreneuriales. Sa passion est de faire en sorte que les vastes systèmes sécurisés complexes fonctionnent correctement – un défi difficile qu'il s'efforce de relever tous les jours.

Au fil des ans, Wilson a mis au point une technique qui lui permet d'installer les nombreuses applications qu'il utilise généralement sur n'importe quel ordinateur portable. Il a appliqué cette technique pour aider les autres développeurs à maintenir également à jour leurs ordinateurs.

La présentation PAC de Wilson vise principalement à partager les leçons apprises en tentant de mettre en œuvre des scripts d'installation efficaces, rapides et axés sur le développeur pour les clients OS X. Sur l'une de ses diapositives, il cite [Mary Meeker](#) à Kleiner Perkins (à propos de l'adoption de la technologie) : « Il a fallu 75 ans pour que le lave-vaisselle s'impose comme une technologie courante, alors que l'accès des consommateurs à Internet a nécessité moins de dix ans. »

Wilson encourage à tirer parti de [Bash](#) en tant qu'outil go-to pour l'installation du logiciel côté client. La maîtrise de Bash est une compétence essentielle pour ceux qui souhaitent soutenir la fourniture efficace d'outils de développement aux équipes de la manière la plus simple possible.

Wilson termine en offrant à son auditoire quelques conseils de professionnel :

- L'optimisation automatisée est désormais une fonctionnalité attendue, en particulier dans les déploiements cloud.
- Nous devons devenir des scientifiques des données pour mieux comprendre l'informatique et les statistiques quotidiennes.
- Il faut oublier la capacité. C'est dynamique et infini.
- Les tests de performance sont toujours nécessaires dans l'environnement sans serveur.
- Lorsque vous planifiez une session, prévoyez du temps pour mettre à niveau vos outils avant chaque itération.
- Prévoyez du temps dans la planification des sessions pour vous mettre d'accord sur un ensemble d'outils d'équipe.
- Aussi évident que cela puisse paraître, tous les scripts d'installation doivent utiliser des variables et des schémas de configuration, et non des valeurs codées en dur.
- Lorsque vous utilisez un script de données téléchargé à partir d'un environnement cloud particulier, exécutez-le en dehors de l'environnement cloud donné.
- Utilisez [pushd et popd](#) pour vous rendre temporairement à un répertoire.
- Utilisez des gestionnaires de paquets tels que [Windows Chocolatey](#), [Brew](#) pour Mac, [apt-get](#) pour ubuntu/debian, [yum](#) pour Red Hat et Centos et [zypper](#) pour Suse.
- Contrôlez les scripts Bash et Python avec Lint.
- Testez vos scripts en version bêta avant de les diffuser à un large public.

L'avantage d'utiliser des scripts pour gérer les logiciels sur les ordinateurs clients est que cela permet une meilleure utilisation du temps au cours de la journée de travail. Comme le dit Wilson, « j'ai passé le plus clair de mon temps à déboguer ma pensée plutôt qu'à déboguer mes fautes de frappe. »

Pour vous procurer l'enregistrement de la session complète, cliquez [ici](#).



Jonathon Wright

Cycle de vie des performances numériques : apprentissage cognitif (AIOps)

[Jonathon Wright](#) est un thérapeute numérique, un leader d'opinion stratégique et un éminent adepte de la technologie. Il se spécialise dans les technologies émergentes, l'innovation et l'automatisation et possède plus de 18 ans d'expérience en commerce international au sein d'organisations mondiales. Il est actuellement CTO de Digital-Assured à Oxford, au Royaume-Uni.

Au cours de sa session, J. Wright a défini l'expérience numérique (Digital Experience, DX) et le rôle primordial qu'elle joue dans le cycle de vie des performances numériques (Digital Performance Lifecycle, DPL). Elle aide en effet les entreprises à adopter les performances minimales viables (Minimum Viable Performance, MVPx), les performances en tant que code (Performance as code, PaC) et les données de test en tant que code (Test data as Code, TDaC) nécessaires pour doper le pipeline de livraison de performances continues (Continuous Performance, CPx) à l'aide robots (ChatOps) afin de fournir des logiciels de plus haute qualité à un rythme toujours plus soutenu. Selon lui, les systèmes commerciaux numériques doivent être extrêmement réactifs au changement et résistants aux modes de défaillance imprévisibles du M2M (Machine to Machine), de l'Internet des objets (IoT) et des points terminaux des microservices.

Finalement, J. Wright a recommandé à son public d'acquérir dès à présent les connaissances et la compréhension nécessaires pour combler le fossé des connaissances cognitives et créer une offre axée sur la valeur. Il a ajouté qu'une prise de décision mieux informée via l'adoption cognitive des capacités et des technologies numériques est plus qu'une nécessité.

Pour vous procurer l'enregistrement complet de la session, cliquez [ici](#). Vous trouverez également le blog de Jonathon Wright sur ce sujet [ici](#).



Tingting Zong

Intégration des tests de performance en DevOps

[Tingting Zong](#) est titulaire d'un master en informatique de l'Université du Zhejiang. Elle dirige actuellement une équipe de 20 ingénieurs des performances chez Insigma Hengtian Software Ltd.

Avec plus de huit ans d'expérience en ingénierie des performances, T. Zong a supervisé avec succès plus de 200 projets de tests de performance de logiciels.

Au cours de sa session, T. Zong a expliqué au public comment intégrer au mieux plusieurs tests de performance sous forme d'étapes DevOps individuelles associées à de nombreuses tâches. Son message principal - la criticité d'un processus de livraison de logiciel standardisé. Elle a également renforcé l'idée selon laquelle l'intégration et les tests continus sont des exigences qui permettent de préparer rapidement le processus de soumission, de rétroaction et d'efficacité.

Pour vous procurer l'enregistrement de la session complète, cliquez [ici](#).

Conclusion

La deuxième édition du Performance Advisory Council, en Juin 2018 de Neotys a présenté des informations incontournables pour un professionnel des tests. Les présentations ont porté sur un éventail de sujets axés sur les outils/techniques adaptés aux tests de performance modernes et sur la manière dont les praticiens des tests peuvent développer et améliorer leur approche. Les tests de performance demeurent un aspect critique du cycle de vie du développement logiciel, et nous voulons nous assurer de continuer à fournir un endroit et une plate-forme de partage et d'apprentissage.

Les informations et les conseils partagés lors du PAC 2018 contribueront grandement, nous l'espérons, à améliorer la manière dont les logiciels sont développés à travers le monde. Mais plus important encore, les logiciels résultant des nouvelles techniques et pratiques partagées au cours de ces discussions amélioreront la vie des créateurs et des utilisateurs du monde entier.

