



GUIDE DEVOPS :

Comment utiliser l'APM
pour améliorer le test des
performances

CHAPITRE 1 :

Introduction

Ce bref eBook explique comment l'automatisation des tests de performance associée à des solutions de gestion des performances applicatives (APM) peut être un puissant moteur de mise en œuvre d'une approche DevOps dans l'entreprise.

La mise en œuvre de DevOps s'impose rapidement comme une norme pour les entreprises qui veulent améliorer la qualité et la rapidité du déploiement de leurs applications face aux exigences de la Révolution digitale. Augmenter la vitesse et la qualité des versions des applications est essentiel pour rester compétitif et agile afin d'apporter des modifications à la demande.

Associée aux tests des performances, une solution APM est idéale pour assurer aux principales applications de fournir une excellente expérience aux clients, même en cas de pic de charge.

Mais, considérons tout d'abord les tests de performance : quels sont-ils et pourquoi sont-ils plus importants que jamais ?

Tests de performance

VUE D'ENSEMBLE

Les tests de performance sont un exercice conçu pour simuler et évaluer la stabilité et la réactivité d'une application sous une charge particulière. Ils peuvent également servir à investiguer, mesurer, valider ou vérifier d'autres attributs de la qualité du système, par exemple l'évolutivité, la fiabilité et l'utilisation des ressources.

Il existe différents tests de performance, ayant chacun leur propre finalité :

Test de charge (ou test de charge de travail)

- Simuler le « jour de pointe en production » actuel.

Test de résistance

- Examiner la réaction d'un système sous conditions de charge extrême, comme par exemple 200 % du pic de production actuel.

Test de capacité

- Examiner si l'empreinte et le nombre de serveurs sont correctement jaugés.

Optimisation des appareils, du front-end et du back-end, selon :

- L'effondrement des appels
- La réduction de la latence
- La réduction de la bande passante et de la taille des ressources (css, JS, images)

Test d'endurance

- Exécuter sur une longue période pour déceler toute défaillance.

Le test des performances est assuré par automatisation et n'est pas adapté à une pratique manuelle. La principale capacité apportée par une automatisation est la possibilité d'enregistrer et de simuler par un script l'activité de l'utilisateur final, puis de la reproduire à grande échelle selon des conditions visées, tout en mesurant les performances des applications et du système.

Il existe de nombreuses solutions d'automatisation des tests de performance actuellement disponibles sur le marché IT (sous licence ou en open-source) correspondant à différents budgets et besoins. Beaucoup d'entre elles s'intègrent directement à des solutions APM, telles qu'AppDynamics.

Les tests de performance sont une discipline à part entière et se distinguent de toute autre forme de test des applications. On les qualifie généralement de « non fonctionnels », car ils visent à tester des caractéristiques applicatives autres que celles de leur fonction pure (p. ex. évolutivité et disponibilité sous charge).

Des tests de performance efficaces s'appuient sur la saisie d'exigences non fonctionnelles (NFR) mais essentielles, parmi les suivantes.

Au niveau de l'entreprise

- Identification de la portée (stratégique, tactique)
- Confirmation des éléments livrables escomptés (éviter toutes les surprises).
- Confirmation des plateformes client à tester (fixe, mobile, clients du service, lot).
- Identification des principaux cas d'utilisation, des exigences pour le test et des scénarios de test.
- Identification des indicateurs clés de performance devant être saisis pour mesurer précisément les performances applicatives.
- Définition d'un modèle de charge exact (primordial).
- Décision de la source des tests de performances en interne ou externalisée.

Au niveau technique

- Gestion et dimensionnement d'environnements adaptés aux tests de performance.
- Sélection d'outils d'automatisation des performances appropriés au besoin technique, aux compétences disponibles et au budget (idéalement par preuve de concept).
- Approche du monitoring (idéalement sur la base du monitoring des performances applicatives).
- Recueil des données, analyse, corrélation et rapport.

Une fois saisies, ces NFR permettent de définir des attributs de test qui sont une fidèle représentation du comportement de l'application, notamment :

- Des scripts d'utilisation qui représentent de manière réaliste l'activité des utilisateurs sur un navigateur ou un appareil mobile. (Ils peuvent également représenter des appels d'API d'un client du service.)
- Un modèle de charge de travail exact qui définit :
 - Les types de tests de performances requis.
 - La distribution des utilisations en script selon les tests de performance et la taille de la charge à générer.
- Un environnement de test qui est adapté au niveau de test requis. Ainsi, pour un test complet, le système doit être une copie conforme ou très proche de la production. Pour les besoins d'autres tests, comme ceux des performances d'un composant dans le développement, il peut s'agir d'un système de taille suffisante pour vérifier le respect des accords de niveau de service pour un sous-groupe distinct de fonctionnalités applicatives.

MONITORING ET ANALYSE

La simulation du trafic des utilisateurs ne représente, bien sûr, qu'une partie de l'exercice. Il faut également pouvoir capturer et comprendre les performances des applications sous charge. Les solutions d'automatisation des performances fournissent traditionnellement des indicateurs semblables aux suivants :

- **Temps de réponse** du jeu complet ou de sous-ensembles désignés des scripts d'utilisation. Un utilisateur peut, par exemple, se connecter, effectuer une recherche, puis ajouter un article dans son panier. Chacune de ces actions peut généralement être chronométrée.
- **Le nombre d'utilisateurs simultanés** à tout moment pendant l'exécution d'un test. Cet indicateur est généralement corrélé au temps de réponse pour démontrer comment l'augmentation de la charge affecte le temps de réponse de l'application.
- **Le nombre et le type d'erreurs** qui surviennent pendant l'exécution d'un test. La mise en évidence d'erreurs à un certain niveau de charge est un signe révélateur de problèmes liés à l'évolutivité d'une application.

Même si ces indicateurs sont excellents pour déceler la présence d'un problème, ils n'indiquent pas nécessairement la cause initiale. Pour identifier la cause initiale, il faut saisir des données de performance sur l'intégralité de la pile, incluant le réseau, les serveurs d'hébergement et les bases de données.

Les solutions d'automatisation des performances fournissent depuis longtemps une intégration à des solutions de monitoring ponctuel, telles que Windows Performance Monitor ou SNMP. Même si elles offrent une meilleure visibilité des performances d'une application, elles ne couvrent pas toute la pile et ne signalent que les indicateurs choisis pour les serveurs ou les composants applicatives suivis.

TENDANCES DES TESTS DE PERFORMANCE ACTUELLES

Face à la Révolution digitale et aux exigences commerciales, les solutions d'automatisation de tests de performance ont évolué pour accueillir :

- Une vitesse de sortie accélérée des versions, catalyseur de changement.
- Des versions plus fréquentes, moins importantes.
- Une commercialisation plus rapide des nouvelles fonctionnalités et des corrections.
- Une réduction des difficultés liées aux « déploiements monolithiques géants ».
- Les besoins du client : des applications plus rapides, de meilleure qualité, moins d'attente.

Les fournisseurs d'outils proposent généralement aujourd'hui :

- Une prise en charge ciblée pour des technologies modernes de développement Web, telles que Node.js, AngularJS, SPF et WebSockets.
- L'intégration de solutions de tests fonctionnels bien établies, telles que Selenium.
- L'intégration de serveurs d'automatisation de test, tels que Jenkins comme composante d'une intégration ou d'une prestation continue.
- **Intégration à APM**

Cette dernière évolution est essentielle, car l'intégration à des solutions APM comme AppDynamics procure une perspective et une compréhension inégalées sur les raisons pour lesquelles une application ne parvient pas à évoluer ou à répondre à une certaine charge, et de manière plus importante sur l'impact sur les utilisateurs.

CHAPITRE 2 :

Difficultés liées aux performances

Des défaillances de performance non détectées à n'importe quelle étape du cycle de développement des systèmes (SDLC) peuvent affecter la capacité d'une application à fournir le niveau de service et la disponibilité attendus par l'entreprise et le client.

Le « Shift left » est un terme décrivant le changement par rapport à l'approche classique des tests des applications. Face à l'adoption massive des approches DevOps et d'agilité, les tests « Shift left » ciblent un test précoce et fréquent, pas seulement pour l'assurance qualité mais aussi pour le développement. Cela procure des avantages importants sur une approche de tests d'assurance qualité classiques :

- Plus les défaillances de SDLC sont détectées tôt, plus leur correction est simple et peu coûteuse.
- En détectant et en corrigeant les défaillances de performance de façon précoce, on peut réduire la quantité de tests nécessaires en améliorant leur vitesse et leur qualité.

Ceci étant dit, la question qui vient est : « Comment empêcher les défaillances de performance de se propager dans le SDLC ? ».

Voici ci-dessous quelques causes habituelles d'apparition de défaillances de performance dans les étapes IT suivantes :

- Conception
- Développement
- Test
- Déploiement
- Évolutivité

CONCEPTION

Au niveau de la conception, cette responsabilité incombe clairement aux intervenants IT et métiers, ainsi que quiconque ayant part à la conception de l'application.

Pour simplifier, si une application n'est pas conçue dès le départ pour évoluer, il est très peu probable qu'elle remplira les exigences de performances de l'entreprise ou de l'utilisateur final. De plus, ces problèmes peuvent rester non-détectés jusqu'à une étape tardive du SDLC, une fois que toute correction est devenue une intervention coûteuse et longue.

DÉVELOPPEMENT

Le modèle classique de développement assigne des équipes de programmeurs par groupes de tâches et d'éléments de travail. Parmi les domaines posant un risque aux performances :

- Un manque de contrôle ou un contrôle minimal sur les normes de programmation.
- Le manque d'une vision de grande envergure sur l'application de la part des développeurs.
- Les développeurs ne tiennent généralement pas compte des SLA quand ils codent.
- Les développeurs n'effectuent pas de tests de performance.
- Les développeurs ne communiquent pas avec l'équipe opérationnelle.

La plupart des aspects précédents sont en réalité en train d'évoluer, en grande partie grâce à la large adoption d'un style agile de développement. Toutefois, l'image est toujours contrastée au sein de nombreuses organisations en cours de passage à une approche agile.

Les considérations des performances restent souvent évaluées à un haut niveau même dans une planification mixte agile-sprint et sont souvent reléguées face à la pression du déploiement.

TEST

L'automatisation des tests de performance a évolué d'un simple moteur de reproduction d'actions à des cadres de test et de monitoring sophistiqués. Il y a beaucoup de choix sur le marché, notamment des offres en open-source bien établies comme JMeter.

La bonne nouvelle vient du fait que la plupart des entreprises, quelle que soit leur taille, utilisent à présent une certaine forme d'outil d'automatisation des performances dans leur processus d'assurance qualité.

La mauvaise nouvelle vient du fait que les tests de performance sont souvent mis en œuvre comme une fonction purement tactique, sans gestion ni processus formels alors qu'ils devraient faire partie intégrante d'une planification IT stratégique. Cela peut entraîner une saisie incomplète des NFR de performance et la prise de décisions concernant la sortie des applications sur la base de résultats de test incomplets ou illusoire.

DÉPLOIEMENT

En matière de décision de déploiement, le risque potentiel lié aux performances est considérable.

Si les responsables du déploiement n'ont aucune visibilité sur les indicateurs de performance adaptés pour évaluer les versions candidates, il leur manque un élément essentiel pour prendre une décision.

En d'autres termes, il doit exister des indications claires de respect ou non des SLA de la part de l'application en matière de temps de réponse, évolutivité et disponibilité sous le pic de charge anticipé.

ÉVOLUTIVITÉ

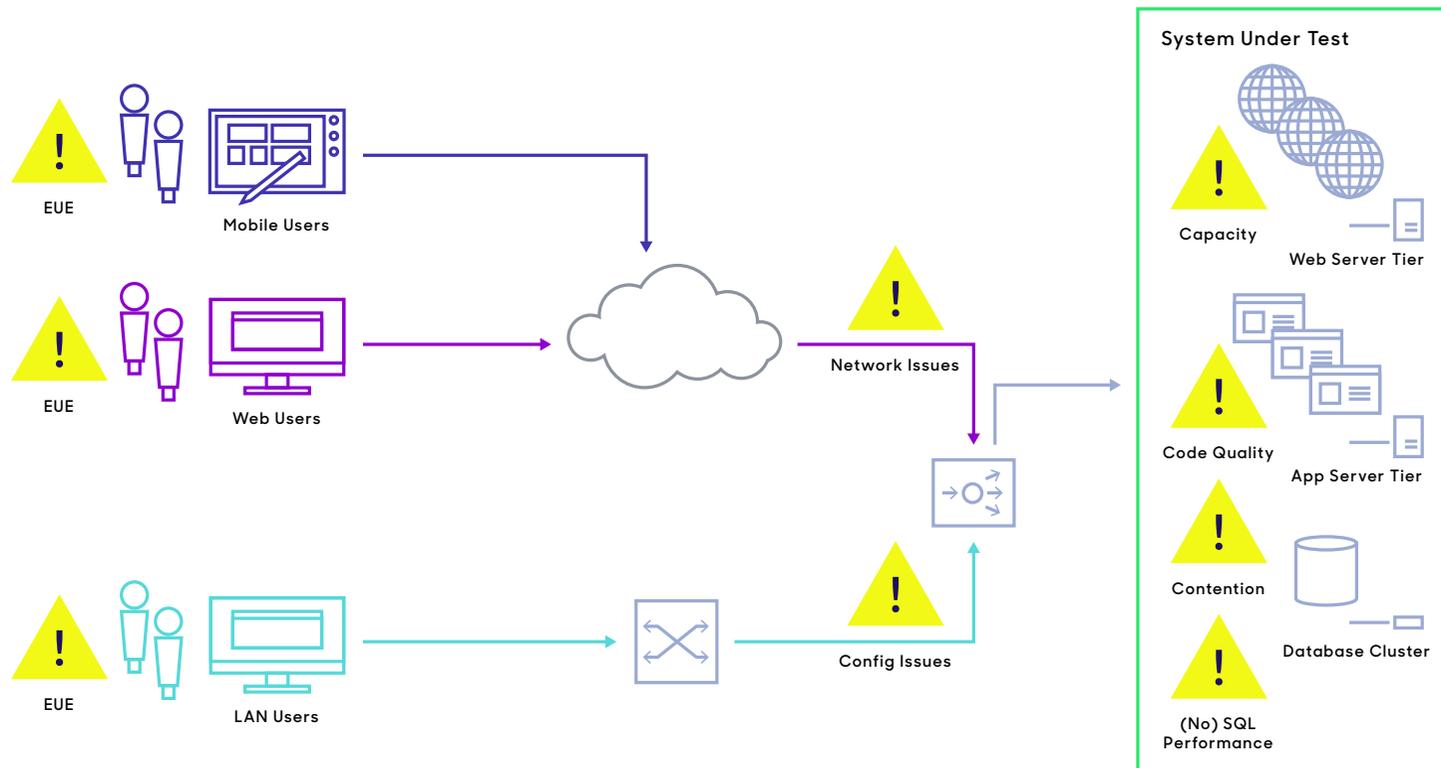
Une application qui ne peut pas évoluer est très vulnérable à tout pic de charge imprévu, pouvant entraîner une perte partielle ou complète de sa disponibilité. Même si elle reste disponible, ses performances au niveau de l'utilisateur peuvent être dégradées au point où elle paraît indisponible.

L'apparition du cloud a permis à de nombreuses applications qui n'étaient pas conçues au départ pour évoluer de continuer à desservir des utilisateurs avec une augmentation de leur charge.

Cette solution temporaire donne un peu de temps, mais elle engage inévitablement des coûts supplémentaires quand l'expansion des instances dans le cloud doit mettre des ressources en œuvre pour répondre à la demande.

Le manque d'évolutivité est souvent le fruit d'un manque de conception, il est donc peu probable qu'il soit anodin (ou peu coûteux) à corriger.

Fig. 1 – Difficultés liés aux performances applicatives



CHAPITRE 3 :

Le processus

Maintenant que nous avons vu les risques liés aux performances et comment ils se présentent pendant le SDLC, penchons-nous sur les manières de limiter ces risques à l'aide de bonnes pratiques de test des performances et de la gestion des performances applicatives (APM).

Gestion des performances applicatives (APM)

UN BREF HISTORIQUE

L'APM a évolué d'un simple profilage du code habituellement utilisé dans le développement à une analyse de la pile complète des applications sur l'intégralité du

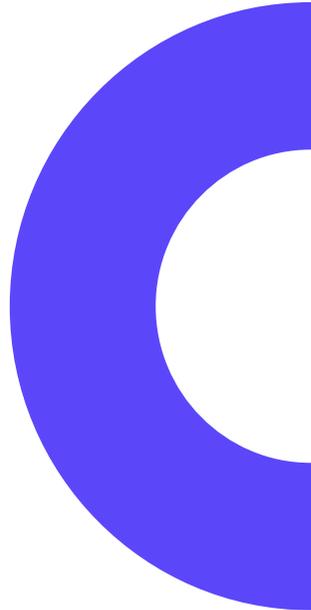
SDLC : du dernier kilomètre au premier avec une attention toujours plus importante à l'expérience client.

La vue transactionnelle des actions de l'utilisateur (du clic dans le navigateur jusqu'à l'appel de la base de données) fournie par APM est la manière la plus efficace de trier et de résoudre les problèmes de performances des applications dans le cadre de test de performances ou de monitoring en production.

APM devient rapidement la solution qui s'impose pour une gestion centralisée des performances et de la disponibilité, ainsi que le monitoring des applications principales et des utilisateurs.

Surtout, la capacité d'APM à couvrir tout le SDLC en fait un choix naturel d'association avec l'automatisation des tests de performance.

Reprenons les étapes IT que nous avons abordées au chapitre 2 pour voir comment une solution APM et des tests de performance peuvent réduire les risques de performance et permettre une approche DevOps.



FAIRE FACE AUX RISQUES DES TESTS DE PERFORMANCE

Conception

Au stade de la conception, il n'existe généralement rien de tangible pour travailler. Toutefois, on peut tirer parti des tests de performance et d'une solution APM afin de décider au moment de la conception en exécutant des essais de preuve de concept sur les technologies candidats aux composants clés de l'application.

La comparaison des performances d'une base de données NoSQL en est un exemple. Les solutions APM comme AppDynamics assure la prise en charge d'une large variété de bases de données NoSQL, notamment un monitoring et une analyse complets.

Du point de vue DevOps, une stratégie peut également être mise en place pour le monitoring APM sur tout le SDLC afin d'identifier les principaux indicateurs de SLA :

- Indicateurs clés de performances applicatives
- Indicateurs clés de performances de service
- Indicateurs clés de performances d'infrastructure

Ceux-ci peuvent être reconduits et conservés au fur et à mesure de l'avancée de l'application dans son SDLC.

Développement

Pendant le développement, une solution APM telle qu'AppDynamics ajoute de la valeur à une approche DevOps en offrant une méthode de test précoce des performances au niveau des composants applicatifs, puis en étendant la solution APM aux environnements de développement.

En développement, le luxe de tester quoi que ce soit intégralement n'existe pas en général, mais on peut toutefois effectuer des tests de performances efficaces. On se concentre sur la qualité du code, aux problèmes de conflit et aux performances API, ce qui peut être facilement suivi par des tests de performance et une solution APM.

Les défaillances des performances peuvent être détectées de manière précoce et corrigées avant de se propager à des niveaux ultérieurs de déploiement et de test.

De plus, des tests de performance peuvent être intégrés aux batteries de test nocturne et une solution APM utilisée pour comparer les performances entre des versions de code et ainsi déclencher des alertes d'homologation / de rejet dans le cadre de l'intégration au serveur d'automatisation de tests.

Test

Le stade de l'assurance qualité a tendance à être la première opportunité pour effectuer un test complet, c'est pourquoi on peut profiter de la vision sur la pile complète des performances applicatives fournie par APM pour de nombreux aspects, notamment celui, si important, de l'expérience client.

Une fois que les tests de performance en développement deviennent un processus normal, il est possible de réduire significativement les tests itératifs de performance pour l'assurance qualité. Ces derniers deviennent alors une confirmation des indicateurs de performance restés non affectés par un déploiement complet.

Déploiement

Comme nous l'avons vu, un obstacle courant pour choisir la version candidate la mieux adaptée pour le déploiement est le manque d'indicateurs de performance pour étayer cette décision.

Quand un test des performances est mis en œuvre comme partie intégrante du processus formel, les indicateurs de performance sont fournis à chaque étape des tests durant tout le SDLC. Un rapport formel montrant la conformité des valeurs des indicateurs de performance d'une application doit être un facteur obligatoire au processus de décision concernant le déploiement d'une version.

Les données supplémentaires concernant les performances fournies par une solution APM permettent de confirmer qu'une application est prête pour le déploiement. De plus, les indicateurs clés de performance reconduits depuis la conception et conservés lors des tests en développement et de l'assurance qualité peuvent servir de base au monitoring en production, fourni idéalement par APM.

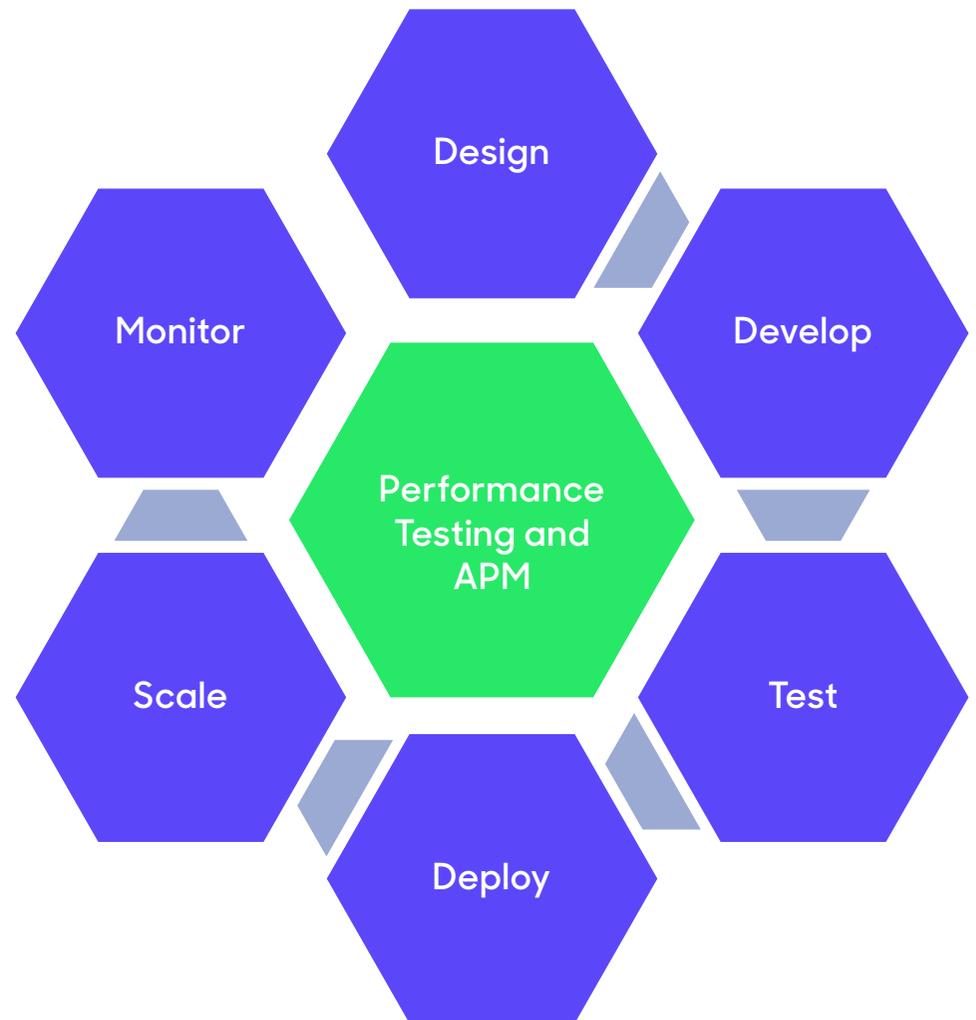
Évolutivité

Enfin, il est primordial qu'une application puisse faire face aux pics de charge (prévus ou imprévus). Les tests de performance, en eux-mêmes, montrent les limites de capacité d'une application mais n'indiquent pas nécessairement les raisons de son échec à évoluer.

En associant le test des performances à une solution APM, l'évolutivité d'une application peut être directement reliée à la base de donnée hôte et à l'activité du code, ce qui permet de situer précisément les raisons de l'échec de la prise en charge du niveau requis.

Cette visibilité renforcée peut aussi servir à explorer les limites supérieures de l'évolutivité d'une application afin de planifier de façon fiable toute augmentation future de la capacité d'infrastructure.

Fig. 2 - Test des performances et APM à travers SDLC



CHAPITRE 4 :

Évaluer la réussite

Examinons enfin les manières d'évaluer la réussite. Nous avons parlé des difficultés liées aux performances et de la façon de limiter les risques et d'activer une approche DevOps grâce aux tests de performance et à une solution APM, mais comment évaluer l'efficacité de notre approche ?

Liste de vérification pour réussir

Afin de mesurer notre réussite, nous devons d'abord confirmer que nous suivons les bonnes pratiques en matière des tests de performance et d'APM. Revenons à nos étapes IT une dernière fois et établissons la liste de vérification des actions principales.

CONCEPTION (POUR LA PERFORMANCE)

- Faites des performances une considération fondamentale de la conception d'une application.
- Définissez un ensemble d'indicateurs clés de performance qui convient pour chaque nouvelle application ou chaque nouveau groupe de fonctionnalités :
 - Temps de réponse
 - Simultanéité
 - Débit
 - Disponibilité
- Évaluez toutes les exigences de modification selon leur impact sur les performances.

DÉVELOPPEMENT (POUR LA PERFORMANCE)

- Testez les performances en développement (c.-à-d. shift left)
- Intégrez les tests de performance au processus APM standard.
- Intégrez les tests de performance à APM dans la batterie de tests automatisés nocturnes.
- Suivez le nombre de défaillances de performance séparément de celui des défaillances fonctionnelles.
 - Suivez s'ils augmentent ou diminuent.
- Intégrez l'évaluation des performances à la planification des scripts.
- Utilisez APM pour suivre les environnements de test en développement.

L'examen régulier de la liste précédente devrait servir de guide pratique afin de mettre en œuvre une gestion efficace des performances applicatives dans votre entreprise.

TEST (POUR LA PERFORMANCE)

- Test de performance en assurance qualité.
- Intégrez les tests de performance au processus APM standard.
- Suivez le nombre de défaillances de performance séparément de celui des défaillances fonctionnelles.
 - Suivez s'ils augmentent ou diminuent.
- Utilisez APM pour suivre les environnements de test en assurance qualité.

DÉPLOIEMENT (POUR LA PERFORMANCE)

- Faites des indicateurs de performances formels basés sur les indicateurs clés de conception, un facteur obligatoire au processus de décision de déploiement d'une version.
- Utilisez APM pour suivre les applications principales en production.
- Utilisez APM pour suivre l'expérience utilisateur de toutes les applications principales.
- Utilisez les indicateurs clés de performance en conception comme références des performances applicatives en production.
- Évaluez la qualité de chaque version en termes de défaillances de performance signalées.

ÉVOLUTIVITÉ (POUR LA PERFORMANCE)

- Ayez connaissance des exigences d'évolutivité des applications au moment du déploiement et dans les perspectives de croissance.
- Concevez des applications qui évoluent.

La détermination de votre réussite s'effectuera en suivant trois indicateurs clés :

1. **Le nombre et la localisation des défaillances des performances :**

Vous devriez idéalement remarquer la réduction du poids de la découverte des défaillances et leur passage de la production et de l'assurance qualité à la conception et au développement à mesure que s'implantera la sensibilité aux performances DevOps.

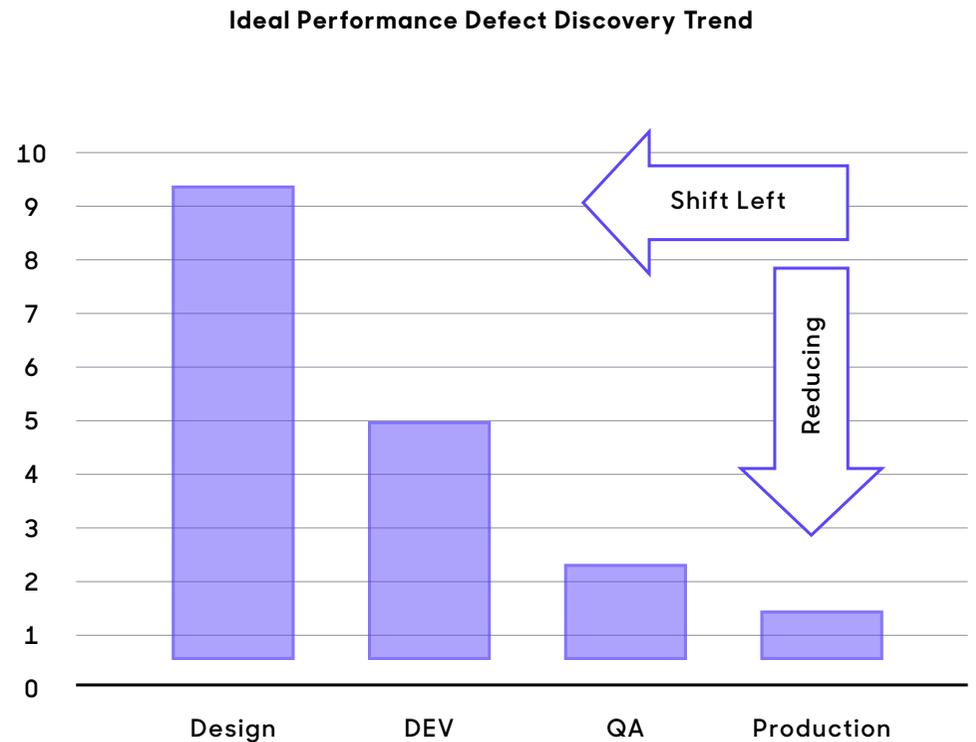
2. **La rapidité et la qualité des versions :**

Une amélioration de la rapidité et de la qualité des versions devrait confirmer l'avantage d'une détection « shift left » et d'une réduction du nombre de défaillances de performance.

3. **Des utilisateurs satisfaits :**

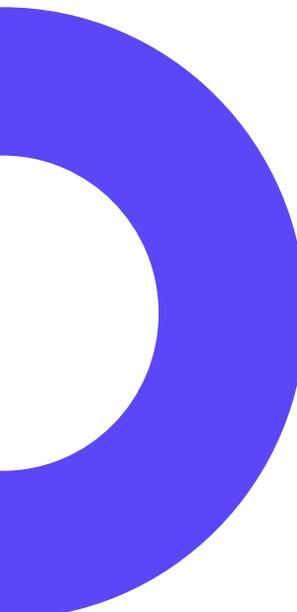
Des versions déployées toujours à temps qui respectent, voire dépassent, les attentes des entreprises et des utilisateurs. Les performances seront la confirmation ultime d'une bonne mise en œuvre de la gestion des performances applicatives dans votre entreprise.

Fig. 3 – Découverte des défaillances des performances – Approche Shift left





CONCLUSION



La valeur apportée par les tests de performance et la capacité d'assurer les performances des applications est incontournable. Et ces tests sont considérablement renforcés par APM.

Nous avons vu comment tirer parti des perspectives sur les performances apportées par APM sur l'ensemble du SDLC afin d'améliorer la vitesse et la qualité du déploiement des nouvelles versions.

Nous avons également abordé les difficultés habituelles en matière de performance, les raisons pour lesquelles elles surviennent et nous avons proposé des manières de les éviter.

Les solutions APM arrivent à maturité et font à présent partie intégrante d'un plan de tests des performances. Découvrez plus en détail comment [AppDynamics](#) peut améliorer les tests de performances applicatives dans votre entreprise.