LIVRE BLANC | JUIN 2016

Au-delà du masquage et de la définition de sous-ensembles : concrétiser la vraie valeur de la gestion des données de test

Huw Price
CA Technologies

ca technologies

Table des matières

Section 1

Introduction

Le concept de gestion des données de test (*Test Data Management*, TDM) n'est peut-être pas nouveau, mais il est souvent négligé, sous-estimé et méconnu, et sa valeur ajoutée réelle pour les entreprises rarement concrétisée. Bon nombre d'organisations et de fournisseurs perçoivent cet aspect comme une problématique relevant uniquement de l'environnement, et le réduisent à la copie, au masquage et, potentiellement, à la définition de sous-ensembles de données de production. Ces données migrées sont ensuite considérées comme une « gold copy » (ou version de référence), qui sera utilisée dans les environnements d'assurance qualité (QA) et de développement. L'objectif d'une gestion TDM améliorée est alors de prendre en charge plus rapidement cette version, puis de répondre aux demandes de nouvelles versions.

La génération de données synthétiques, le cas échéant, est utilisée seule et au cas par cas pour chaque projet. Cela s'explique par l'hypothèse, largement acceptée, que la génération de données est une bonne pratique pour une équipe ou un projet spécifique, mais n'est pas viable à l'échelle de l'entreprise. Les organisations modernes disposent généralement de bases de données complexes et volumineuses, contenant des millions d'enregistrements stockés sous des formats variés et utilisant des outils disparates. L'argument est donc qu'il est nécessairement plus facile de copier des données de production existantes, plutôt que de tenter de profiler et de modéliser des données aussi complexes, comme l'exigerait la génération de données synthétiques réalistes.

Cette hypothèse est cependant discutable, et les organisations souhaitant réaliser pleinement les avantages d'une gestion TDM améliorée doivent réévaluer la possibilité d'appliquer à l'échelle de l'entreprise la génération de données synthétiques, ainsi que la manière dont elles stockent, gèrent et provisionnent les données. La génération de données synthétiques est non seulement plus efficace en termes de temps, de qualité et de coût, mais elle s'avère aussi souvent plus simple et plus sûre qu'un masquage total des données de production, pour autant que les organisations adaptent leurs technologies, processus et structures d'équipes en conséquence.

Le présent document met en balance les problématiques courantes et les bénéfices potentiels de la gestion des données de test d'un point de vue environnemental, en tenant compte des processus, des technologies et des structures d'équipe les plus courants au sein d'une organisation[1].

Section 2

Les inconvénients d'une approche purement logistique de la gestion des données de test

Pour les organisations qui dépendent des données de production pour les environnements de développement et de test, le masquage et la définition de sous-ensembles de données sont des passages obligés. Le masquage est nécessaire pour assurer la conformité vis-à-vis de la législation en vigueur concernant la protection des données, tandis que la définition de sous-ensembles de données de production permet de réduire des coûts d'infrastructure excessivement élevés.

Toutefois, si la gestion TDM commence et finit par le masquage et la définition de sous-ensembles de données, sa valeur réelle pour l'entreprise n'est pas pleinement concrétisée. Cela peut être prouvé sur la base de deux hypothèses : la première est que les données recueillies par une organisation moderne contiennent des informations d'identification personnelle qui, d'après la loi en vigueur, ne peuvent pas quitter l'environnement de production sous une forme reconnaissable ; la seconde est que les données masquées seront utilisées à des fins de développement et de test, et doivent donc être en quantité suffisante pour tous les plans de test à exécuter, et toute nouvelle fonctionnalité.

Sur la base de ces deux hypothèses, nous allons démontrer qu'en réalité, le masquage n'est pas la méthode la plus simple ni la plus efficace pour provisionner des données sécurisées adaptées aux environnements de développement et de test. Nous étudierons ensuite les structures d'équipe, les technologies et les processus mis en œuvre dans une organisation type, démontrant que la faible couverture des données de production, ajoutée à la façon dont celles-ci sont habituellement gérées, implique que les problématiques récurrentes de retard de projet, de dépassement de budget et de défauts en production ne seront pas résolues, si la TDM s'appuie uniquement sur le masquage et la définition de sous-ensembles de données.

Le masquage n'est pas une solution simple

Commençons d'abord par expliquer pourquoi il est plausible de remettre sérieusement en question la croyance selon laquelle une copie et un masquage sécurisés des données de production destinées aux environnements de développement et de test est l'alternative la plus simple dans une structure IT complexe. En réalité, l'effort nécessaire pour masquer les données de production tout en préservant l'intégrité référentielle excède souvent celui nécessaire pour modéliser les données, alors même qu'un masquage efficace requiert de toute manière un profilage des données. D'autre part, certains aspects complexes des données sont souvent laissés non masqués, en une sorte de compromis, ce qui ne fait pas du masquage la solution la plus sûre qui soit dans nombre de situations.

Lorsque les données sont masquées pour une utilisation dans des environnements autres que de production, les relations intercolonnes au sein des données d'origine doivent être préservées, même lorsque le contenu sensible est masqué. Au premier niveau, cela signifie que l'intégrité référentielle est préservée. À un niveau plus élevé, toutefois, cela signifie que les relations de colonne complexes sont maintenues. Par exemple, imaginons un champ « total », qui calcule le total de plusieurs colonnes ; ces colonnes doivent donc rester alignées dans les données masquées. Plus complexes, les relations de données temporelles et causales (par exemple, un total basé sur une valeur temps), qui sont beaucoup plus difficiles à masquer de manière homogène.

En outre, ces relations doivent non seulement être masquées de manière cohérente au sein d'une base de données (intrasystème), mais, pour une organisation de grande taille classique, possédant plusieurs types de base de données et des applications composites complexes, elles doivent également être préservées entre les différentes bases de données (intersystèmes). Plus les données sont compliquées, plus cette opération est difficile et, parallèlement, plus les données sont faciles à pirater, car il y a davantage d'informations à mettre en corrélation.

La plupart du temps, le processus de masquage des données est centré principalement sur le contenu, ainsi que sur les relations intersystèmes et intrasystème, dans le but de garantir l'intégrité référentielle. La difficulté à préserver simultanément le contenu, les relations intercolonnes et les relations temporelles a généralement pour conséquence que l'un de ces éléments est sacrifié ; souvent, ce sont les relations temporelles entre les colonnes qui restent inchangées. Bien qu'il ne s'agisse pas à proprement parler de contenu sensible d'un point de vue juridique ou technique, ces informations non modifiées peuvent être utilisées pour identifier des informations sensibles, en les mettant en corrélation avec des informations extérieures aux données.

Prenez, par exemple, un ensemble de journaux de transactions masqués. Le pirate sait qu'une personne donnée a réalisé une transaction d'un certain montant à une heure donnée. Bien que cette information ne soit pas présente dans le contenu sensible qui a été masqué, les informations temporelles (heure de la transaction) peuvent encore être présentes dans la base de données masquée, du fait qu'il est extrêmement difficile de parvenir à masquer le contenu, les valeurs numériques totales et les heures de transaction de façon homogène. Une fois qu'il a identifié ces informations sensibles, le pirate peut les retrouver à nouveau dans tous les systèmes et bases de données, car l'intégrité intersystèmes et intrasystème a été préservée. La conséquence est une désanonymisation de fait des données.

Même si vous parvenez à profiler et masquer correctement les données dans toute leur complexité, les données ne pourront pas être considérées comme totalement sécurisées, car certaines informations seront malgré tout conservées dans leurs aspects les plus complexes. Par exemple, un pirate pourrait identifier les effets de causalité en cascade d'un journal de transactions en combinant les informations causales, intrasystème et intersystèmes avec des méthodes temporelles, telles qu'une comparaison par instantanés. Il serait ensuite à même de déduire la façon dont les données ont été modifiées durant une opération, ainsi que tous les éventuels effets de causalité en cascade existants (p. ex., les déclencheurs de base de données). À nouveau, la solidité des données masquées n'est égale qu'à la solidité de son maillon le plus faible : une fois cette information identifiée, un pirate peut remonter les relations intercolonnes, déchiffrer des exigences commercialement sensibles ou, pire encore, des informations d'identification personnelle.

Si nous partons de l'hypothèse que la quasi-totalité des données d'entreprise contiennent des informations d'identification personnelle, le masquage ne constitue en aucun cas une méthode sûre ou efficace pour provisionner les données à des environnements autres que de production. C'est encore plus vrai au vu de l'injonction croissante faite aux organisations de sécuriser leurs données : l'amende moyenne en cas de violation de données a augmenté de 13 % en 2013, pour atteindre 3,5 millions de dollars [5, Ponemon, 2014], tandis que le prochain règlement général européen sur la protection des données (GDPR) a pour objectif de renforcer encore la mise en œuvre de la législation existante, qui interdit l'utilisation d'informations personnelles pour tout motif autre que celui pour lequel elles ont été recueillies.

De plus, si nous tenons compte du rapport de Symantec « State of Privacy Report 2015 » qui révèle que les problématiques de sécurité des données déterminent où et comment 88 % des consommateurs européens font leurs achats, il apparaît clairement qu'il n'est plus envisageable de prendre le risque d'utiliser des données masquées dans des environnements hors production. À présent que nous avons établi les risques juridiques d'une migration des données de production vers les environnements de développement et de test, nous pouvons évaluer l'efficacité et la rentabilité d'un tel processus.

Équipes

Dépendances de données

Il est rare qu'une organisation dispose d'une équipe centralisée responsable de la gestion des données de test (TDM). La plupart du temps, différentes équipes sont chargées de gérer, d'identifier et de créer leurs propres données. Elles travaillent alors isolées les unes des autres, dans leurs propres environnements de développement ou de test, mais en partageant souvent les mêmes sources de données. Ce manque de centralisation et de collaboration engendre des contraintes de dépendances de données : lorsqu'une équipe apporte des modifications à une base de données, celles-ci affectent toutes les équipes. Frustration, retards et remaniements incessants en sont la conséquence, avec des tests qui échouent sans raison apparente, les équipes ne parvenant pas à identifier si cet échec est dû à un défaut de code ou à une erreur de données. En outre, les équipes sont rarement en mesure de gérer les versions des données de test, de les paramétrer ou encore de restaurer les bases de données si d'autres équipes les ont « cannibalisées ».

Le cycle de développement logiciel (SDLC) est davantage perçu comme un enchaînement d'étapes linéaires, dans lequel chaque équipe exécute sa tâche avant de transférer le résultat à l'équipe responsable de l'étape suivante. Dans un tel contexte, il est trop fréquent que les équipes de développement et de test se retrouvent bloquées en raison de retards en amont. Elles doivent alors attendre que les données souhaitées deviennent disponibles, ou encore que des spécifications et des flux de données soient finalisés par d'autres équipes. Une part importante du cycle SDLC peut alors se passer à attendre des données.

Ce manque de parallélisme est à l'opposé de la volonté actuelle de nombreuses organisations de fonctionner en mode livraison continue, dans lequel les équipes doivent accéder à des données spécifiques à chaque étape du SDLC. Il est ainsi impossible d'effectuer un développement de fin de vie efficace sur les anciennes versions d'un système, tout en utilisant les données existantes pour les nouvelles versions. Au lieu de cela, les équipes se retrouvent avec un nouvel environnement de développement, mais aucune donnée pour le remplir.

Technologies

Dépendances système

Outre les dépendances de données, les contraintes matérielles et système sont des problématiques classiques des équipes de développement et de test modernes. La complexité des applications n'a fait qu'augmenter au cours des vingt dernières années, avec un nombre de dépendances croissant vis-à-vis des autres systèmes, tant au sein de l'organisation qu'en dehors. Cela engendre souvent des problèmes lors des tests du système, car les services sont parfois instables ou indisponibles. Une autre problématique rapportée par les équipes de développement et de test est le manque d'environnements pleine taille. Les équipes peuvent constater qu'une autre équipe a priorité sur l'environnement qu'elles souhaitent utiliser, ou bien ne pas parvenir à obtenir les données souhaitées parce qu'elles sont employées par une autre équipe. Bien que, au départ, de telles contraintes ne paraissent pas directement problématiques en matière de données, nous découvrirons plus tard qu'une meilleure infrastructure TDM est nécessaire à leur résolution.

Stockage des données

Les organisations modernes conservent d'importants volumes de données de production. Créer des copies de ces bases de données et les exécuter sur des machines de développement est une opération à la fois lente et onéreuse. Le stockage de ces données est ainsi responsable en grande partie de la hausse des coûts d'infrastructure, et notamment des coûts de matériel, de licences et de support. Les serveurs subissent en outre des demandes croissantes : ils doivent livrer d'importants volumes de données simultanément pour différents jobs, tout en exécutant des connexions ouvertes, en maintenant des fichiers ouverts et en gérant des injections de données[2].

Sans une réévaluation de la manière dont les données sont gérées, il est peu probable que ces coûts puissent être réduits. La quantité de données collectées et stockées par une entreprise moyenne double chaque année[3] et, avec l'avènement du Big Data, les organisations parlent aujourd'hui en pétaoctets, et non plus en téraoctets. Le stockage des données occupe donc une part de plus en plus conséquente dans les budgets IT, avec des chiffres de croissance atteignant 20 % pour le secteur du stockage de données, selon les estimations[4]. Les organisations d'aujourd'hui doivent donc se demander si chaque copie des données de production est vraiment nécessaire, certaines entreprises conservant même plusieurs copies d'une même base de données. La réponse est vraisemblablement non.

Exploration et profilage des données

Sans une technologie partiellement ou entièrement automatisée, la détection des données est l'une des problématiques majeures pour les équipes souhaitant livrer dans les temps des logiciels pleinement testés, tout particulièrement dans un contexte de développement agile ou dans un cadre de livraison continue. Les testeurs peuvent passer plus de la moitié de leur temps de travail à rechercher des données, ce qui les oblige à faire de douloureux compromis : exécuter tous les tests nécessaires pour éviter que des défauts coûteux ne parviennent jusqu'en production ou livrer le logiciel dans les délais.

Cette situation est encore compliquée par un stockage hétérogène des données dans des feuilles de calcul non contrôlées (présentant peu, voire pas d'indexation ou de référencement croisé, par exemple) plutôt que dans un référentiel ou un entrepôt centralisé. En outre, les bases de données sont rarement documentées correctement, car les organisations sont souvent dépourvues de dictionnaires centralisés pour les attributs des données de test et les requêtes SQL associées. L'exploration et l'affectation des données sont donc entravées, et les équipes sont dans l'impossibilité de demander des données à l'aide de modèles ou de formulaires standard, ou encore sur la base des critères qui les intéressent. Par conséquent, elles doivent souvent trouver manuellement un petit ensemble de données adéquat pour leurs besoins et leurs plans de test spécifiques, une opération à la fois longue et source d'erreurs.

L'absence d'outils automatisés pour l'exploration et le profilage des données augmente également le risque de non-conformité. Les données étant stockées dans des feuilles de calcul non contrôlées, il est possible de trouver des informations sensibles partout, par exemple, dans une colonne de remarques s'il n'existait pas de champ de colonne approprié pour ces données ou si tous les champs pertinents étaient déjà pleins. Sans possibilité de rechercher des champs spécifiques, il est moins probable que ces informations soient identifiées et elles risquent donc de se retrouver dans des environnements hors production. Ce mode de fonctionnement est contraire aux réglementations en matière de protection des données, qui stipulent que les données peuvent être utilisées uniquement dans le but pour lequel elles ont été recueillies. Les entreprises ne respectant par ces réglementations encourent une amende moyenne de 3,5 millions de dollars[5] et risquent une perte de chiffre d'affaires importante.

Processus

Les données de production ne constituent pas une « gold copy »

L'incapacité à trouver des données adéquates pour répondre aux besoins et aux plans de test nous amène au principal problème lié à l'utilisation de données de production dans des environnements hors production : celles-ci ne peuvent tout simplement pas pourvoir à toutes les demandes de données qu'une organisation recevra vraisemblablement, tout particulièrement celles en provenance des environnements de développement. Comme nous l'avons expliqué, les équipes de développement et de test ont besoin de données à chaque étape du cycle de développement. Une base de données de référence, ou « gold copy », correctement prise en compte, doit donc contenir un ensemble standard de données, sur la base duquel les tests seront effectués de manière

répétée, ainsi que les données nécessaires pour pouvoir réaliser tous les tests possibles. En outre, ces données doivent être à jour et réalistes (conformes à de vraies données de production), incluant notamment des valeurs hors normes et toutes les données antérieures. Les données de production ne répondent qu'à deux de ces conditions : elles sont réalistes et peuvent contenir toutes les données antérieures. Elles ne constituent donc en aucun cas une « gold copy ».

La plupart des données de production sont très semblables, couvrant les transactions habituelles de l'organisation, et excluent par nature les données qui pourraient entraîner une panne du système. Elles ne contiennent donc aucune valeur hors normes, ce qui exclut les nouveaux cas de figure et les scénarios problématiques lors de la réalisation des tests. Ce sont pourtant ces résultats inattendus, ces valeurs hors normes et ces conditions limites qui causent généralement une panne de système, et l'objectif des phases de développement et de test devrait précisément consister à tester ces scénarios problématiques, négatifs et imprévus. Lorsque vous vous basez uniquement sur des méthodes d'échantillonnage, des défauts se retrouvent en production, où leur résolution peut coûter jusqu'à 1 000 fois plus[6] et nécessiter jusqu'à 50 fois plus de temps que s'ils avaient été réglés durant les phases précédentes[7].

D'autre part, il est probable que les données échantillonnées à partir de l'environnement de production ne soient jamais à jour, du fait que l'environnement évolue et que les demandes métier reçues par les équipes IT changent constamment. Les données n'étant pas stockées indépendamment des environnements, lorsqu'un environnement change, une actualisation du système, voire une mise à jour de la version, est nécessaire. Cela peut cannibaliser les données et les scénarios déjà créés à partir de multiples sources de données de production, et entraîner la perte d'ensembles de données utiles. Ceux-ci devront ensuite être recréés manuellement, une tâche véritablement fastidieuse. En outre, ces actualisations du système sont généralement des processus très lents. Par exemple, nous avons travaillé par le passé avec des organisations où ce type de processus a duré jusqu'à neuf mois.

La création manuelle des données peut offrir une solution temporaire, permettant une exécution immédiate des plans de test disponibles. Cependant, ces données étant créées pour des besoins ou des plans de test bien spécifiques, il est probable qu'elles deviennent immédiatement obsolètes. Les taux de change des devises et les structures commerciales en sont de bons exemples. Dans ce type de contexte, les données deviennent obsolètes d'un jour à l'autre, ce qui signifie que les données créées manuellement ne peuvent être réutilisées et sont quasiment à usage unique. Il est alors nécessaire de déployer d'immenses efforts pour créer de nouvelles données pour chaque test, avec pour conséquence un retard dans la réalisation des projets, tandis que les tests sont repoussés au sprint suivant ou à la phase suivante du cycle de développement.

Si un masquage efficace sur plusieurs bases de données exige que les données soient tout d'abord profilées, sans pour autant garantir leur sécurité ou leur qualité, la question se pose : pourquoi, une fois les données de production correctement profilées, l'organisation ne choisit-elle pas de simplement générer les données requises synthétiquement ?

Section 3

L'alternative idéale : équipes, processus et technologies

Améliorer la politique TDM implique d'adopter une approche structurée, mais surtout centralisée, dans la gestion des données à l'échelle de l'entreprise. Cela permet non seulement de résoudre nombre des problèmes évoqués précédemment, mais également de bénéficier d'un moyen souvent meilleur marché, plus simple et plus efficace pour provisionner les données dans les environnements de développement et de test, avec les technologies adéquates.

En stockant les données utiles en tant que ressources extérieures aux environnements, pour les y intégrer ensuite à la demande, vous éliminez les problèmes environnementaux de la gestion TDM, qui se concentre alors sur la façon de récupérer les données. À son tour, cette opération peut être réalisée de manière efficace grâce à une approche centralisée en matière de stockage des données, dans laquelle les données sont modélisées en tant que ressources réutilisables, pouvant être extraites à la demande,

sous la forme de sous-ensembles précis. Au lieu de se contenter d'un masquage et d'une définition de sous-ensembles de données en les considérant comme des passages obligés, la génération de données synthétiques présente un objectif stratégique qui, une fois intégré à une politique TDM globale, permet de livrer des logiciels entièrement testés, dans les délais et le budget impartis.

Technologies

Profilage automatique des données

Nous avons déjà démontré que, pour effectuer un masquage efficace des données de production, celles-ci devaient tout d'abord être profilées. Toutefois, même ainsi, les données ne seront pas pleinement sécurisées et ne conviendront pas parfaitement aux exigences en matière de développement et de test. Il existe cependant des technologies automatisées permettant de réduire la charge de travail nécessaire au profilage des données dans les environnements IT complexes. Pour profiler les données, il est tout d'abord nécessaire de les « enregistrer », en collectant le plus de métadonnées possible. Ces métadonnées incluent, entre autres, les noms de tables, ainsi que les noms et les types de colonnes, et elles existent même pour les systèmes de base de données non relationnelle, sous la forme de copybooks pour les systèmes mainframe et de documents de mappage pour les fichiers plats délimités ou à largeur fixe.

Une fois l'enregistrement effectué, vous pouvez appliquer des algorithmes mathématiques de détection des données. À l'aide de CA Test Data Manager (anciennement Data Maker de Grid-Tools), par exemple, cette opération s'effectue tout d'abord individuellement pour chaque schéma, avec détermination des informations d'identification personnelles (PII) et ingénierie inverse des relations de base de données, si nécessaire. Une fois cette opération effectuée, les systèmes peuvent être réunis. Ce faisant, CA Test Data Manager utilise des vues cubiques pour profiler toutes les relations entre les données, même les plus complexes, créant ainsi un ensemble de données multidimensionnel où chaque dimension du cube représente un attribut des données. Ce profilage permet aux organisations de savoir exactement quelles sont les données existantes et où celles-ci sont stockées, ainsi que d'identifier les éventuelles brèches dans la couverture fonctionnelle.

Génération de données synthétiques

Après avoir établi une vue d'ensemble précise des données existantes et identifié les données supplémentaires requises pour les environnements de test, il est ensuite possible de générer automatiquement les données manquantes. Chaque environnement réel pouvant être envisagé comme un point de données, il est possible de modéliser et de créer des données de manière à couvrir 100 % des variations fonctionnelles. Ces données incluront les scénarios futurs n'ayant encore jamais été observés, ainsi que des « données erronées », des valeurs hors normes et des résultats imprévus. Cela permet de réaliser des tests négatifs efficaces et de développer un nouveau système ou sous-système. Vous bénéficiez également ainsi d'une méthode de test systématique, afin que les résultats et les scénarios imprévus qui n'ont pas été envisagés par les testeurs n'entraînent pas de panne du système et que les défauts soient détectés avant leur passage en production.

Si vous ne disposez pas de données suffisantes pour effectuer des tests répétés, vous pouvez également créer de gros volumes de données à l'aide d'une technologie automatisée. CA Test Data Manager propose un outil automatisé fonctionnant directement avec les couches API des systèmes de gestion de bases de données relationnelles (SGBDR) ou des ERP, qui permet de générer des données aussi rapidement que la puissance de calcul du système le permet. Ses scripts de regroupement peuvent multiplier le volume de données de l'organisation par deux, aussi rapidement que son infrastructure le permet. Pour résumer, cette technologie automatisée permet de créer rapidement des données réalistes pour réaliser tous les tests, y compris des tests négatifs, et en volume suffisant pour exécuter des tests répétés.

Gestion centralisée des données

Grâce aux améliorations technologiques susmentionnées, votre organisation est déjà bien avancée dans la création d'une « gold copy », telle que définie dans ce document (voir « Les données de production ne constituent pas une gold copy »). En stockant ensuite les données, modélisées en tant qu'objets réutilisables, dans un entrepôt de données de test centralisé, vous pouvez également commencer à mettre en place une fonctionnalité permettant d'identifier rapidement des sous-ensembles de données spécifiques aux environnements de développement et de test, à la demande.

Clonage de données

Une fois les données modélisées en tant qu'objets dans un entrepôt de données de test ou « datamart de test », et après avoir créé les dictionnaires de ressources de données et les requêtes associées, vous pouvez identifier des sous-ensembles de données spécifiques, puis les cloner et les injecter dans les environnements de développement et de test.

Le module de clonage de données de CA Test Data Manager, par exemple, extrait de petits ensembles de données de test cohérents à partir de multiples systèmes de développement et de production interconnectés, remplaçant le processus long et coûteux utilisé auparavant, qui consistait à copier et déplacer des bases de données volumineuses et complexes. Cette capacité à extraire, copier et fournir uniquement les données nécessaires signifie pour une organisation qu'il n'est plus nécessaire d'entretenir un grand nombre de copies complètes des bases de données de production.

De plus, l'existence d'un entrepôt de données centralisé et la possibilité de cloner des données permettent également d'éliminer les dépendances de données entre les équipes, en séparant le provisioning et l'utilisation des données. Autrement dit, les données peuvent être clonées, puis livrées à plusieurs équipes en parallèle, ce qui évite de devoir attendre que des données en amont soient disponibles et d'influencer négativement le travail d'autres équipes en apportant des modifications aux données.

La modélisation et le stockage central des données en tant qu'objets flexibles et réutilisables permettent également de reproduire facilement des scénarios pertinents et des bogues. Les données étant explicitement définies dans la notification de bogue, une technologie de clonage rapide et flexible permet d'exécuter de manière répétée des tests complexes et rares, sans mobilisation des données. Cette fonctionnalité est particulièrement précieuse lorsque vous effectuez une actualisation des données, car il n'est alors pas nécessaire de fusionner les données et vous évitez de perdre des ensembles de données intéressants.

Contraintes matérielles

Combinée à des outils de virtualisation, la génération de données synthétiques peut parfois aider à résoudre les problèmes de contraintes matérielles et système. Vous pouvez ainsi émuler les couches de message à l'aide des métadonnées d'un système, de manière à profiler et à générer avec précision des messages de service réalistes (y compris les fichiers SOAP, REST et MQ, ainsi que les fichiers plats). Un moteur de génération automatique de données est placé sous une machine virtuelle, afin de rédiger des réponses de message réalistes, par exemple des paires requête/réponse.

La virtualisation de machines complètes permet de créer plusieurs environnements de développement. Les équipes peuvent alors travailler sans soucis dans ces environnements, même lorsque des composants interdépendants sont indisponibles, ce qui évite les retards en amont tout en virtualisant les systèmes et équipements hérités coûteux à des fins de test.

Processus

Développement en parallèle et possibilité de réutilisation

Outre qu'elle résout les problèmes environnementaux, la capacité à identifier et cloner les données en fonction d'attributs spécifiques permet aussi de résoudre une autre problématique majeure de la gestion des données de test, à savoir comment assurer un provisioning efficace des données aux équipes de test ou aux testeurs. Cette fonction permet de demander, de partager et de réutiliser les données en parallèle, à la demande.

En accédant à un portail Web centralisé de données de test à la demande, comme celui proposé par CA Test Data Manager, les testeurs et les développeurs peuvent envoyer des requêtes portant exactement sur les données dont ils ont besoin pour leur tâche en cours. Ils y saisissent leurs critères (c'est-à-dire, les attributs des données de test souhaitées), le portail envoie un job au moteur de traitement par lots, qui recherche les données appropriées dans les systèmes back-end ou clone les données demandées, puis les renvoie. Plus besoin de rechercher ou de créer manuellement les données, ce qui réduit considérablement le temps nécessaire au traitement des requêtes de données.

Il est recommandé de normaliser au maximum les questions du formulaire, ce qui permet de réutiliser le travail effectué par chaque équipe. Par exemple, si vous disposez de données synthétiques créées précédemment, vous pouvez paramétrer les entrées et les présenter dans des listes déroulantes sur le portail, de sorte que tout utilisateur puisse demander ces données même lorsque le plan de test est différent. Outre les données de test, vous pouvez stocker des structures de création de données, des tests unitaires, des ressources virtuelles et des scripts d'automatisation, puis les utiliser comme éléments de base pour un travail ultérieur.

Contrôle de version

Un contrôle de version efficace rend possible le travail en parallèle nécessaire pour développer en permanence de nouveaux systèmes, dans les délais et les budgets impartis. Par exemple, l'entrepôt de données de test de CA Test Data Manager permet à une équipe de copier des données depuis un référentiel, qui en « hérite », au même titre que les pointeurs renvoyant vers les versions précédentes. Les données peuvent ainsi évoluer d'une version à une autre, car elles sont verrouillées pour une équipe donnée, tout en pouvant être restaurées facilement, transférées ou rapprochées avec différentes versions. Lorsqu'une modification est apportée à l'une des versions, le numéro de celle-ci augmente ou diminue, mais l'original reste intact. Imaginons qu'une équipe ait besoin d'ajouter une nouvelle colonne sur l'ensemble d'une base de données : grâce à CA Test Data Manager, s'il existe des parents codés en dur, l'équipe peut retrouver tous les enfants liés et définir des valeurs par défaut, ou bien générer des données à l'aide de séquences ou de fonctions standard.

Équipes

Enfin, des changements structurels des équipes peuvent venir compléter ces améliorations technologiques et procédurales, apportant leur contribution dans la résolution des problématiques précédemment discutées. La centralisation de la gestion des données de test au sein d'une équipe dédiée est à associer avec la centralisation des ressources de stockage, de gestion et de provisioning des données, ce qui permet de mieux répondre aux besoins de l'entreprise. Cette équipe dédiée peut prendre en charge le provisioning des données, ainsi que la gestion des données elles-mêmes, mais aussi être responsable de la création de nouvelles données et du profilage, si nécessaire.

Cela permet non seulement d'éviter les contraintes de dépendance de données entre les équipes, mais signifie aussi que les requêtes de données et le reporting de bogues peuvent être régis par une même entité. Vous pouvez ainsi mettre en place des « barrières qualité », tandis que la propriété des données est centralisée au niveau de l'équipe de sécurité IT. La fonction de création de formulaire dynamique proposée par le portail de données de test à la demande de CA Test Data Manager contribue à cet objectif, car elle va au-delà des droits d'accès basés sur les rôles, provisionnant les données sensibles uniquement au personnel autorisé qui en fait la demande.

Section 4

Références

1 Les inconvénients de l'utilisation des données de production dans des environnements hors production sont abordés dans un autre document. Reportez-vous notamment aux articles de Huw Price, « Reduce Time to Market with Test Data Management » et « How better Test Data Management is the only way to drive Continuous Delivery ».

2 Jacek Becla et Daniel L. Wang, « Lessons Learned from managing a Petabyte », p. 4. En date du 19/02/2015, via le site http://www.slac.stanford.edu/BFROOT/www/Public/Computing/Databases/proceedings/

3 « Lessons Learned from managing a Petabyte »

4 http://www.computerweekly.com/feature/Meeting-the-demand-for-data-storage

5 http://www.ponemon.org/blog/ponemon-institute-releases-2014-cost-of-data-breach-global-analysis

6 http://benderrbt.com/Bender-Requirements%20Based%20Testing%20Process%20Overview.pdf

7 http://www.softwaretestingclass.com/why-testing-should-start-early-in-software-development-lifecycle/

Section 5

Avantages de CA Technologies

CA Technologies (NASDAQ : CA) fournit des solutions de gestion des systèmes d'information qui aident ses clients à gérer et à sécuriser des environnements informatiques complexes pour supporter des services métier agiles. Les organisations s'appuient sur les logiciels et les solutions SaaS de CA Technologies pour accélérer l'innovation, transformer leur infrastructure et sécuriser les données et les identités, du cœur des data centers jusqu'au Cloud. CA Technologies s'engage à ce que ses clients atteignent les résultats souhaités et la valeur métier attendue grâce à l'utilisation de sa technologie. Pour en savoir plus sur nos programmes de succès client, rendez-vous sur le site **ca.com/customer-success**. Pour plus d'informations sur CA Technologies, rendez-vous sur le site **ca.com/fr**.

Section 6

À propos de l'auteur

Avec une carrière qui a débuté il y a plus de 30 ans, Huw Price a travaillé comme architecte technique principal pour plusieurs éditeurs de logiciels européens et américains. Il a apporté son aide à des banques internationales, de grandes agences de service public et des fournisseurs de soins de santé, pour la conception d'une architecture sophistiquée. Élu « Directeur informatique de l'année 2010 » par QA Guild, Huw s'est spécialisé dans la conception d'outils d'automatisation de tests et a lancé plusieurs produits innovants qui ont redéfini le modèle de test utilisé par l'industrie du logiciel. Il intervient désormais dans le cadre d'événements de renommée mondiale et son travail a été publié dans de nombreuses revues spécialisées, parmi lesquelles Professional Tester et CIO Magazine.

Sa dernière entreprise, Grid-Tools, a été acquise par CA Technologies en juin 2015. Depuis près de 10 ans déjà, Grid-Tools a redéfini la manière dont les grandes entreprises abordent leur stratégie de test. Grâce au leadership et à l'approche visionnaire de Huw, l'entreprise a adopté une solide approche de la phase de test, axée sur les données, en lançant de nouveaux concepts formulés par Huw comme les « objets-données », l'« héritage des données » et « un entrepôt centralisé de données de test ».

ca technologies

Restez connecté à CA Technologies sur ca.com/fr

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

CS200-205127