

WHITE PAPER | APRILE 2016

Test Data Management

Perché può rappresentare l'unico metodo per promuovere
la continuous delivery

Huw Price
CA Technologies



Sommario

Sezione 1:	3
Introduzione alla continuous delivery	
<hr/>	
Sezione 2:	3
Un TDM di scarsa qualità ostacola la continuous delivery	
<hr/>	
Sezione 3:	6
Approccio al TDM basato sui requisiti	
<hr/>	
Sezione 4:	8
Riepilogo	
<hr/>	
Sezione 5:	9
Riferimenti	
<hr/>	
Sezione 6:	9
L'autore	

Sezione 1

Introduzione alla continuous delivery

Nel settore dello sviluppo software, lo slogan del momento è "continuous delivery". Vista l'importanza attribuita al concetto, numerosi vendor promettono di trasformare in realtà i principi della continuous delivery offrendo i propri strumenti come rimedio alle tradizionali cause di ritardo e di fallimento dei progetti. L'idea è che, adottando questi strumenti, le aziende saranno in grado di innovare continuamente e fornire software di qualità nei tempi previsti e nel rispetto del budget.

L'attrattiva della continuous delivery è facilmente comprensibile. Nell'odierna economia delle applicazioni, le aziende contano sul software per offrire valore ai propri clienti. Di conseguenza, le esigenze dell'IT e del business sono allineate in modo sempre più stretto e la posizione di un'azienda sul mercato dipende fortemente dalla sua capacità di fornire valore ai consumatori su base quotidiana. I team IT devono essere in grado di rispondere rapidamente ai mutamenti del mercato e delle aspettative dei clienti, sviluppando software che risponda ai cambiamenti nelle esigenze critiche del business e allo stesso tempo riducendo i costi e i tempi del testing.

Anche se molti strumenti dichiarano di favorire la continuous delivery, offrono per lo più servizi di tipo logistico e possono essere implementati solo in una fase avanzata del ciclo di vita di sviluppo. L'approccio DevOps è affrontato solo in termini di Operations, con strumenti che supportano i test di sviluppo e regressione di software già progettato. I problemi sistemici che insorgono nelle fasi precedenti, fin dalla fase di raccolta dei requisiti, restano irrisolti e provocano costosi colli di bottiglia e ritardi nei progetti, che rendono impossibile un'implementazione efficace della continuous delivery.

In particolare, viene trascurata l'importanza del Test Data Management (TDM), ovvero della capacità di fornire i dati giusti al momento giusto e nel posto giusto a scopi di testing. La progettazione inadeguata dei test case e il provisioning inefficiente di dati di qualità scadente costringono i team a lavorare senza disporre dei dati necessari per il testing completo di un sistema. La qualità del software viene così sacrificata a favore del rispetto della tempistica e del budget.

Sezione 2

Un TDM di scarsa qualità ostacola la continuous delivery

Per la maggior parte delle aziende e dei vendor, il TDM inizia e finisce con la copia, il masking e a volte il subsetting dei dati di produzione. Queste strategie TDM hanno uno scopo puramente logistico e si concentrano solo sul trasferimento dei dati, che vengono invece trascurati nella fase di progettazione dei requisiti di progetto. Dalla copia e dalla migrazione dei dati di produzione in un ambiente di sviluppo, test o QA si ottiene la cosiddetta "gold copy", ovvero un set perfetto di dati di test che può essere utilizzato per eseguire qualsiasi test necessario per la verifica completa di un sistema.

Il masking e il subsetting rappresentano un buon punto di partenza e permettono di risolvere in parte le criticità tradizionali associate alla compliance e ai costi di infrastruttura. Il masking e il subsetting di un database di produzione, tuttavia, sono soggetti a molti degli stessi problemi intrinseci all'uso dei dati di produzione, proprio quelli che rendono impossibile il successo nell'implementazione della continuous delivery.

Sprechi di tempo

La maggior parte delle aziende non dispone di un servizio di provisioning o di un team TDM centrale, quindi i dati devono essere reperiti o creati dai team locali. I tester possono dover dedicare fino al 50% del tempo lavorativo alla ricerca di dati e il tempo trascorso in attesa dei dati può rappresentare addirittura il 20% del ciclo di vita totale del software. Questo crea colli di bottiglia nel testing che rendono impossibili sia una reale agilità, sia la continuous delivery. Grid-Tools (ora parte di CA Technologies) ha osservato il caso di un team per cui erano stati pianificati "sprint" di tre settimane, ma che aveva impiegato quattro settimane solo per preparare i dati per lo sprint.

Il masking o il subsetting dei dati di produzione implicano la presenza di team di test dedicati esclusivamente alla raccolta dei dati da utilizzare. Questo può essere un compito lungo e gravoso, ulteriormente complicato dallo storage incoerente dei dati in fogli di calcolo non controllati. Per fare un esempio, se ci sono tre colonne per le carte di credito, ma casualmente un cliente ne ha quattro, potrebbero esserci informazioni sensibili archiviate in una colonna "note". Per di più, i dati devono essere intatti dal punto di vista referenziale. Più i dati sono complessi, più questo obiettivo è difficile da raggiungere e il tempo e gli sforzi necessari per il masking dei dati sono spesso superiori a quelli necessari per crearli da zero.

La creazione manuale dei dati da zero, d'altronde può essere anch'essa molto dispendiosa in termini di tempo. Inoltre, poiché i dati vengono in genere creati per test case specifici, diventano rapidamente obsoleti e non pertinenti in un'ottica di continua evoluzione. Un esempio tipico è rappresentato dai modelli di trading e dai tassi di cambio, in cui i dati diventano obsoleti da un giorno all'altro. In questi casi è necessario aggiornare i dati, con un ulteriore accumulo di ritardo, o, più spesso, elaborarli. Lo stesso si verifica quando una versione cambia, se viene richiesto un aggiornamento dei dati o quando è necessario creare i dati ogni volta che viene apportata una modifica a un ambiente virtuale.

Raramente è possibile condividere e riutilizzare i dati creati manualmente tra team diversi. Questa incapacità di utilizzare gli sforzi profusi in precedenza nella creazione dei dati crea lavoro extra e costringe le aziende a fare una scelta indesiderabile tra riduzione dei costi/time-to-market e delivery di applicazioni software di valore dotate di tutte le funzionalità richieste.

Vincoli di dipendenza

I tester sprecano moltissimo tempo in attesa del provisioning dei dati o che i dati diventino disponibili. Spesso aziende e vendor considerano il ciclo di vita di sviluppo del software come una serie di fasi lineari, in cui un team completa un'attività e quindi passa il lavoro al team successivo. Di conseguenza, i team devono attendere la disponibilità di dati "idonei allo scopo" dai team a monte e possono trovarsi nella situazione di non poterli utilizzare perché al momento sono utilizzati da un altro team.

Questo è contrario ai principi del parallel development agile su cui si fonda il concetto di continuous delivery, in cui tutto il tempo e tutte le capacità di innovazione di tutti gli individui sono valorizzati al massimo. Inoltre, una modifica apportata da un team può incidere sui dati fino al punto che, quando vengono utilizzati da un altro team, i test sulle applicazioni falliscono apparentemente senza motivo.

Se i team devono creare manualmente i dati o attendere che siano disponibili per giorni o settimane, non ci si può aspettare che rispondano rapidamente ai cambiamenti dei requisiti, né che forniscano software completamente testato.

Scarsa qualità

Quando si utilizzano dati di produzione in un ambiente non di produzione, il vero problema è la qualità. I dati di produzione offrono in genere una copertura funzionale pari solo al 10-20%, pertanto è improbabile che qualsiasi metodo di campionamento possa fornire dati che soddisfino tutti i test case necessari per sviluppare un nuovo sottosistema. I dati di produzione sono in buona misura molto simili, in quanto vengono tratti da transazioni comuni o "di normale amministrazione" e sanitizzati per loro stessa natura in modo da escludere i dati errati che danneggerebbero i sistemi. Per questo motivo, il testing tende a porre particolare attenzione agli "happy path" e meno ai test negativi o non funzionali.

I test negativi, al contrario, dovrebbero rappresentare l'80% del testing perché sono i valori erratici e gli scenari estremi a causare il crash dei sistemi. Se il testing è focalizzato sugli happy path, i difetti arriveranno invariabilmente alla fase di produzione, causando la necessità di rielaborazione, l'aumento dei costi e potenzialmente il fallimento del progetto. Una ricerca di settore ha dimostrato che il tempo necessario per risolvere un bug durante il testing è 50 volte superiore a quello necessario se individuato durante la fase di definizione dei requisiti¹. Tali ritardi e l'incapacità di gestire i cambiamenti rendono impossibile la continuous delivery.

La copertura funzionale dei dati di produzione può essere integrata dalla creazione manuale, ma si tratta di un metodo lungo, inesatto e non scientifico. Non c'è modo di verificare che sia stata raggiunta la massima copertura, né di verificare se i dati sono rimasti intatti dal punto di vista referenziale.

Costi elevati

Infine, oltre ai costi elevati della rielaborazione e dei ritardi nella delivery, l'effettiva copia dei dati di produzione è un processo eccessivamente lungo e costoso. I dati di settore mostrano che alcune aziende arrivano a possedere fino a 20 copie dello stesso database, con spese elevatissime in termini di hardware, licenze e costi di supporto.

Inoltre, lo storage di dati in grandi volumi può essere molto costoso. Una ricerca correlata sullo storage di terze parti evidenzia che spostare i dati in tier di costo inferiore, compreso il cloud, riduce temporaneamente i costi di storage, ma non risolve il problema della crescita dei dati. In altre parole, qualsiasi policy TDM che si limiti ai processi di migrazione, subsetting e masking dei dati non consente di affrontare il problema di come archiviare in modo economicamente conveniente i dati che vengono elaborati dalle sempre più complesse applicazioni composite moderne.

A tutto questo occorre sommare i potenziali costi associati alla mancanza di compliance, un problema che non è possibile risolvere neanche con il masking dei dati di produzione. Uno studio indipendente ha rivelato che nel 2014 il costo medio di una violazione dei dati è aumentato del 15%, raggiungendo i 3,5 milioni di dollari per record. La nuova direttiva dell'Unione Europea sulla protezione dei dati, che entrerà in vigore nel 2016, fissa l'ammenda massima applicata a una violazione dei dati a 100 milioni di euro² o il 5% del fatturato globale, se superiore.

Il masking dei dati di produzione non garantisce la compliance, perché il vero pericolo resta l'errore umano: oltre il 50%³ delle violazioni di dati può essere ricondotto a comportamenti impropri di utenti interni. Inoltre, quando si esegue il masking, l'integrità referenziale dei dati deve rimanere intatta, o il testing delle applicazioni può fallire. Più i dati sono complessi, tuttavia, maggiore è il rischio di rompere l'integrità referenziale, poiché è possibile che più tipi di informazioni siano correlate.

L'unico vero modo per evitare costi di infrastruttura tanto elevati e per assicurarsi che i dati sensibili non vengano divulgati è non usare i dati di produzione. Anche se queste spese non sono una prerogativa delle aziende che desiderano implementare un framework di continuous delivery, certamente sono in contrasto con i suoi principi. Impediscono infatti di fornire software di qualità che risponda all'evoluzione delle esigenze di business in modo rapido e nel rispetto del budget.

Sezione 3

Approccio al TDM basato sui requisiti

Le aziende che intendono implementare un modello di continuous delivery devono necessariamente ripensare i propri processi di testing e sviluppo anziché semplicemente rimanergli, riconsiderando il proprio approccio al TDM. Un approccio end-to-end completo al test data management, elaborato in base ai requisiti, permette all'azienda di implementare il testing con approccio "Shift Left", ridurre i rischi e minimizzare i difetti, creando così software di qualità elevata in tempi più rapidi e a costi più bassi.

Anziché pensare ai dati sulla base dei singoli test case, le aziende dovrebbero pensarli in termini di decisioni di progettazione, ovvero in termini di requisiti, progettando test case con dati direttamente collegati. Personalizzare i dati di test in base ai requisiti assicura che siano "idonei allo scopo", mentre la capacità di renderli rapidamente disponibili consente ai team di test di rispondere altrettanto rapidamente al mutare delle esigenze del business.

A questo proposito, la continuous delivery non può iniziare in una fase avanzata del ciclo di vita di sviluppo del software: deve iniziare con un'idea (i requisiti stessi), in modo che i team di sviluppo e test possano rispondere velocemente al mutare delle esigenze lavorando al passo con l'evoluzione di tale idea. I team devono essere in grado di convalidare e verificare velocemente i requisiti, derivandone direttamente test case collegati ai risultati previsti e a dati virtuali. Questo tipo di testing basato su modello racchiude l'intero concetto di "Shift Left" e continuous delivery di software: condensare tutto il lavoro del ciclo di vita di sviluppo nella fase di raccolta dei requisiti, in modo che tutto il lavoro successivo fluisca facilmente da questa fase anche quando i requisiti cambiano.

Sviluppo di requisiti migliori

I requisiti devono contenere tutte le informazioni qualitative su un sistema necessarie per il testing, in modo che i tester possano derivare use case e test case direttamente dall'idea iniziale, introducendone la tracciabilità. La tracciabilità è indispensabile per permettere agli addetti di aggiornare rapidamente i test al mutare dei requisiti.

Con gli strumenti appropriati, la modellazione basata su diagrammi di flusso consente di produrre un diagramma di flusso che, nonostante la semplicità di progettazione del flusso stesso, contiene tutte le informazioni qualitative su un sistema necessarie per il testing. CA Agile Requirements Designer (in precedenza Grid-Tools Agile Designer) supporta un diagramma di flusso con tutta la logica funzionale necessaria per derivare automaticamente il minor numero possibile di test case con la massima copertura, senza lasciare dubbi come può avvenire con la modellazione basata su causa-effetto o gli approcci pairwise².

Un diagramma di flusso aumenta la probabilità che i requisiti siano completi e privi di ambiguità, suddividendo inoltre i diagrammi complessi e densi di testo che normalmente racchiudono i requisiti in piccoli blocchi facilmente elaborabili. Questi processi riflettono la logica di causa ed effetto di un sistema, creando in pratica una serie di istruzioni "what if, then this".

Questo non solo consente di ridurre del 56%⁴ i difetti che emergono dalle ambiguità nei requisiti, ma costringe anche il team addetto a pensare in termini di modellazione di un sistema, ovvero di vincoli, restrizioni e condizioni limite. Da questi requisiti, modellati sotto forma di diagramma di flusso, è possibile identificare ogni possibile percorso in un sistema e quindi creare test case che offrono una copertura funzionale pari al 100%, inclusi i percorsi negativi e i risultati imprevisi³.

Con un diagramma di flusso attivo, i test case possono essere collegati a metriche di complessità, dati virtuali, dati di test, script di automazione, risultati previsti e backlog, concentrando il lavoro SDLC nella fase di raccolta dei requisiti.

Provisioning di dati "idonei allo scopo"

L'automazione dei processi di progettazione dei test case tramite strumenti come CA Agile Requirements Designer consente di generare i test case giusti, necessari per verificare i requisiti in modo ottimale e in base alle specifiche esigenze. Questo ci porta all'affermazione centrale di questo documento, ovvero che per la continuous delivery è necessario un TDM migliore. I tester devono avere accesso, al momento giusto e nel luogo appropriato, a dati "idonei allo scopo" che consentano di ottenere una copertura dei test case pari al 100%.

Qualità

Come si è detto, i dati di produzione non possono fornire la copertura necessaria per il testing completo di un sistema. La generazione sintetica di dati di test, al contrario, consente di produrre set di dati piccoli e completi che coprono tutti i possibili scenari, anche quando un evento non si è mai verificato in precedenza. Poiché si può considerare ogni scenario del mondo reale come un altro punto dati, è possibile creare dati sintetici anche per scenari nuovi e futuri. CA Test Data Manager (in precedenza CA Data Finder o Grid-Tools Data Maker) usa tecniche intelligenti di profilazione dei dati per creare un quadro accurato di un modello di dati, generando su questa base dati ricchi e sofisticati che forniscono il 100% di variazioni funzionali.

Con un approccio basato sui requisiti, i dati generati vengono abbinati a un test case, garantendo che sia idoneo alle esigenze di ogni singolo tester. Generare i dati di test in questo modo aumenta la probabilità che i difetti vengano individuati in una fase molto precoce, evitando le lunghe attività di rielaborazione che rendono la continuous delivery impossibile.

Tempi

Dedicare del tempo alla creazione o alla manipolazione manuali dei dati oppure perdere tempo in attesa che siano disponibili è in contrasto con i principi dello sviluppo continuo così come descritto.

Al contrario, strumenti automatizzati per la creazione dei dati, come CA Test Data Manager, possono lavorare direttamente con i livelli delle API RDBMS o ERP, permettendo agli utenti di generare dati alla massima velocità consentita dalla potenza di elaborazione. L'uso di script di massa può raddoppiare la quantità di dati nella disponibilità di un'azienda, alla massima velocità che l'infrastruttura di database è in grado di gestire. Questo assicura la disponibilità di dati "idonei allo scopo" nel giro di poche ore e non di settimane, in modo che i tester abbiano i dati necessari nel momento in cui ne hanno bisogno nell'ambito di uno sprint.

Inoltre, una potente funzionalità di abbinamento di test permette, se i dati esistono già, di identificarli e raccogliarli da più origini eterogenee e quindi di abbinarli e allocarli ai test case appropriati. L'abbinamento di test ha dato prova di ridurre del 95% il tempo necessario per la ricerca e il provisioning dei dati rispetto ai processi manuali. Inoltre, consente ai team di sapere quali test avranno esito negativo a causa di problemi relativi ai dati già prima dell'esecuzione.

Archiviare i dati di test in un data warehouse centralizzato aiuta a ridurre ulteriormente i colli di bottiglia causati dall'attesa della disponibilità dei dati a valle. Il portale Test Data on Demand di CA Test Data Manager offre una funzionalità di creazione dinamica dei form, che consente agli utenti di selezionare il tipo di dati desiderato sulla base di criteri specifici, ad esempio tipo di carta di credito o posizione geografica. Il form allocherà i dati necessari per il test case oppure, se non ne esistono, creerà nuovi dati. Questo vuol dire che i tecnici avranno più tempo a disposizione per correggere i difetti evidenziati dal testing, anziché perderne quasi metà in attesa di ricevere i dati necessari. Vuol dire anche che le aziende possono centralizzare la proprietà dei dati assegnandola al team di sicurezza IT, fornendo i dati sensibili solo al personale autorizzato che ne faccia richiesta.

Costi

Infine, una migliore policy TDM riduce il rischio di sfornamento dei costi. Individuare e risolvere i difetti in una fase precoce consente di ridurre il numero fino al 95%⁵ e di realizzare risparmi superiori a 50.000 dollari⁵ per difetto. In più, l'uso di subset di dati più completi e di dimensioni inferiori consente di ridurre i costi di infrastruttura fino a 50.000 dollari⁵ per database, mentre eseguire un numero inferiore di test, ma di qualità migliore, consente di ridurre notevolmente i tempi e i costi del testing. L'uso di dati sintetici consente di eliminare il pericolo di mancata compliance e i costi correlati, in quanto non è necessario che le informazioni sensibili escano dagli ambienti di produzione.

Sezione 4

Riepilogo

La copia di dati di produzione in ambienti non di produzione come "source of truth" impedisce un'implementazione efficace della continuous delivery. Capita spesso che i team di test non abbiano a disposizione dati idonei al testing completo del software, trovandosi nell'incapacità di rispondere rapidamente all'evolversi dei requisiti di business. Anche se gli strumenti di subsetting e masking possono risolvere alcune criticità, i dati di produzione non sono di una qualità sufficiente per individuare e risolvere i difetti.

Adottare un approccio end-to-end basato sui requisiti allo sviluppo del software e trattare i dati di test come una risorsa preziosa e riutilizzabile è fondamentale per il successo della continuous delivery. Lavorando al passo con l'evoluzione di un'idea, i team possono rispondere rapidamente al mutare delle esigenze di business usando dati "idonei allo scopo" forniti al momento giusto e nel posto giusto, per testare al fondo il software con il minor numero possibile di cicli di test.

Questo approccio consente alle aziende di creare efficienze nel provisioning dei dati di test, riducendo inoltre il rischio di ritardi, la necessità di rielaborazione e l'aumento dei costi che rendono la continuous delivery di fatto impossibile. Grazie a tutto questo, saranno in grado di sviluppare in modo continuativo software che risponde ai cambiamenti nelle esigenze critiche del business in tempi più rapidi e a costi più bassi.

Sezione 5

Riferimenti

1 <http://www.softwaretestingclass.com/why-testing-should-start-early-in-software-development-life-cycle/>

2 <http://www.agile-designer.com/resources/test-case-generation-bloor-market-report/>

3 Per un'analisi matematica e concettuale dei vantaggi della modellazione basata su diagrammi di flusso, consulta l'articolo di Llyr Wyn Jones "A critique of testing primer", disponibile all'indirizzo <http://www.agile-designer.com/resources/critique-testing-primer/>

Sezione 6

L'autore

Cofondatore di Grid-Tools e VP di CA Technologies



Votato "Direttore IT dell'anno 2010" da QA Guild, Huw Price è stato Lead Technical Architect per diverse aziende software in Europa e Stati Uniti. Specializzato in strumenti per l'automazione dei test, ha lanciato numerosi prodotti innovativi che hanno trasformato il modello di testing usato nel settore del software.



Entra in contatto con CA Technologies all'indirizzo ca.com/it



CA Technologies (NASDAQ: CA) crea software che promuove l'innovazione all'interno delle aziende, consentendo loro di sfruttare le opportunità offerte dall'economia delle applicazioni. Il software rappresenta il cuore di qualsiasi business, in ogni settore. Dalla pianificazione allo sviluppo, fino alla gestione e alla sicurezza, CA Technologies lavora con le aziende di tutto il mondo per cambiare il nostro modo di vivere, interagire e comunicare, in ambienti mobile, cloud pubblici e privati, distribuiti e mainframe. Per ulteriori informazioni sui programmi a supporto dei nostri clienti, visita ca.com/customer-success. Ulteriori informazioni all'indirizzo ca.com/it.

1 Ponemon Institute: 2014 Cost of Data Breach

2 Regolamento sulla protezione dei dati proposto dall'Unione Europea

3 2013 Data Breach Investigations Report

4 Report Bender: Requirements Based Testing

5 Metriche raccolte dall'esperienza di implementazione di Grid-Tools