

DOCUMENTAÇÃO TÉCNICA | ABRIL DE 2016

Gerenciamento de dados de teste

Como esta pode ser a única maneira de viabilizar a Entrega contínua

Huw Price
CA Technologies



Sumário

Seção 1:	3
Introdução à Entrega contínua	
<hr/>	
Seção 2:	3
TDM ineficaz é entrave à Entrega contínua	
<hr/>	
Seção 3:	6
Abordagem ao TDM com base em requisitos	
<hr/>	
Seção 4:	8
Resumo	
<hr/>	
Seção 5:	9
Referências	
<hr/>	
Seção 6:	9
Sobre o autor	

Seção 1

Introdução à Entrega contínua

No universo do desenvolvimento de software, o termo Entrega contínua tornou-se uma espécie de frase de efeito. Diante disso, diversos fornecedores prometem transformar o termo em realidade, oferecendo suas ferramentas como solução aos tradicionais motivos de atrasos e falhas em projetos. Ao adotar essas ferramentas, as organizações poderão sempre inovar e produzir software de qualidade dentro dos prazos e orçamentos, sugerem os fornecedores.

A Entrega contínua tem um atrativo compreensível. Na moderna economia de aplicativos dos nossos dias, as organizações dependem de softwares para gerar valor aos seus clientes. Há, portanto, um maior alinhamento entre as necessidades de negócios e de TI, e a capacidade de uma empresa gerar valor diário ao cliente definirá sua posição no mercado. Sendo assim, as equipes de TI deverão estar prontas para se posicionar rapidamente diante do mercado e das expectativas dos clientes (caracteristicamente inconstantes) desenvolvendo software capaz de suprir as necessidades fundamentais de negócios e reduzir, ao mesmo tempo, os custos e o tempo gasto.

Embora muitas ferramentas prometam viabilizar a Entrega contínua, sua natureza é basicamente logística, podendo ser implantadas apenas nos estágios avançados do ciclo de vida do desenvolvimento. A aplicação da metodologia de DevOps diz respeito apenas a Operações, e as ferramentas utilizam software já existente para, então, promover o desenvolvimento e os testes de regressão. O surgimento de problemas sistemáticos logo no início, na fase de apuração dos requisitos e daí em diante, persiste e cria gargalos e atrasos dispendiosos nos projetos; isso torna impossível a implementação bem-sucedida da Entrega contínua.

Particularmente, a importância do Gerenciamento de dados de teste (TDM), ou o envio dos dados certos ao lugar certo, no momento certo, para fins de teste, é negligenciada. Casos de teste mal-elaborados e o provisionamento ineficaz de dados de má qualidade são sinônimos de equipes de teste sem os dados necessários para testar completamente os sistemas. A entrega do software dentro do prazo e do orçamento é feita em detrimento da qualidade.

Seção 2

TDM ineficaz é entrave à Entrega contínua

Para a maioria das organizações e dos fornecedores, o TDM começa e termina com a cópia, o mascaramento e a possível subdivisão dos dados de produção. Tais estratégias de TDM, puramente logísticas, concentram-se unicamente na migração de dados, enquanto os dados propriamente ditos são ignorados na elaboração dos requisitos do projeto. Uma vez copiados e migrados para um ambiente de desenvolvimento, teste ou controle de qualidade, os dados são normalmente chamados de "cópia de ouro", isto é, um conjunto perfeito de dados com os quais os testadores poderão executar qualquer teste necessário à verificação completa dos sistemas.

Mascaramento e subdivisão são um excelente ponto de partida e percorrem boa parte do caminho rumo à eliminação dos tradicionais problemas relacionados a conformidade e custos de infraestrutura. No entanto, a subdivisão e o mascaramento de bancos de dados de produção carregam em si muitos dos problemas inerentes ao uso de dados de produção propriamente ditos. Esses problemas tornam impossível a implementação bem-sucedida da Entrega contínua.

Tempo desperdiçado

A maioria das organizações carece de uma equipe de serviços de provisionamento ou de TDM central para que os dados sejam localizados ou criados pelas equipes locais. Os testadores gastam até 50% do seu tempo procurando dados, enquanto cerca de 20% do ciclo de vida do desenvolvimento de software (SDLC) é desperdiçado na espera desses dados. Isso gera gargalos na execução de testes e torna impossível oferecer agilidade e Entrega contínua genuínas. Em certa ocasião, a Grid-Tools (hoje parte da CA Technologies) se viu diante de uma equipe que deveria realizar "sprints" em três semanas, mas acabou gastando quatro semanas preparando os dados para o sprint.

O mascaramento ou a subdivisão de dados de produção implica a criação de equipes de testes com a missão de ir em busca dos dados a serem utilizados. Essa é uma tarefa árdua e cansativa que se torna ainda mais difícil devido ao armazenamento inconsistente de dados em planilhas sem controle. Por exemplo, as informações pessoalmente identificáveis (PII) poderiam ser incluídas em uma coluna de "notas" se, suponhamos, houvesse três colunas para cartões de créditos, mas o cliente, por acaso, possui quatro delas. Além disso, os dados devem ser mantidos intactos para fins de referência. Quanto mais complexos os dados, mais difícil obtê-los; com isso, o tempo e o esforço gastos mascarando-os geralmente supera o tempo que os testadores usariam para produzi-los do zero.

Criar dados manualmente do zero pode ser também um processo demorado. Além do mais, os dados perdem rapidamente sua atualidade e importância, pois o mundo real não para de mudar, e eles normalmente visam casos de teste específicos. Um exemplo comumente citado são os padrões de negociação e as taxas de câmbio, nos quais a desatualização dos dados é diária. Quando isso acontece, os dados precisam ser atualizados (outro grande consumo de tempo) ou, mais frequentemente, "queimados". Esse também é o caso quando há mudança de versão, ou alguém solicita uma atualização de dados, ou é preciso criar dados a cada vez que um ambiente virtual muda.

Raramente é possível compartilhar e reutilizar dados entre equipes criados manualmente. A incapacidade de aproveitar os esforços iniciais gastos na produção de dados gera trabalho extra, obrigando as organizações a fazer escolhas radicais e indesejadas entre reduzir custos e encurtar o tempo de lançamento no mercado ou fornecer aplicativos de software importantes com todas as funcionalidades exigidas.

Limitações de dependência

O testador passa boa parte do seu tempo aguardando o provisionamento ou a disponibilização de dados. Quase sempre, organizações e fornecedores encaram o ciclo de vida do desenvolvimento de software como uma série de etapas lineares, nas quais a equipe A conclui algo e entrega o bastão à equipe B. Assim, as equipes são obrigadas a esperar que as outras equipes disponibilizem os dados adequados a determinada finalidade; com isso, acabam se vendo incapazes de usar esses dados porque a outra equipe ainda está trabalhando neles.

Isso contraria os princípios do desenvolvimento paralelo ágil na essência da Entrega contínua, em que cada momento do tempo e da habilidade do indivíduo em inovar é usado para explorar seu valor máximo. Além disso, determinada mudança feita por uma equipe poderá afetar os dados de tal forma que, quando a outra equipe os utilizar, os testes de aplicativos apresentarão falha sem nenhum motivo aparente.

Se as equipes estão sendo obrigadas a criar dados manualmente ou aguardar dias ou semanas por sua disponibilização, não se pode esperar delas uma resposta rápida às mudanças nos requisitos nem a entrega de software totalmente testado.

Baixa qualidade

O verdadeiro problema envolvendo o uso de dados de produção em ambientes não relacionados à produção está na qualidade. Normalmente, os dados de produção preveem apenas de 10% a 20% de cobertura funcional; portanto, qualquer método de amostragem está fadado a não disponibilizar dados capazes de atender a todos os casos de teste exigidos para desenvolver um novo subsistema. Grande parte dos dados de produção é bastante semelhante, sendo extraída de transações "rotineiras" e selecionadas de acordo com sua essência para excluir os dados inválidos que causam interrupções nos sistemas. Por essa razão, os testes normalmente se concentram mais no "que dá certo" e menos no que não funciona ou dá errado.

Entretanto, testar o que não dá certo deveria constituir cerca de 80% das verificações, pois são esses elementos estranhos e cenários marginais os responsáveis pela falência dos sistemas. Devido a essa preocupação em testar o que dá certo, invariavelmente haverá deficiências na produção, e isso acarreta retrabalho, atrasos comprometedores, custos incontroláveis e possíveis falhas nos projetos. Segundo pesquisas do setor, o tempo gasto para corrigir erros na fase de testes é 50 vezes maior do que o tempo gasto para identificá-los na etapa¹ dos requisitos, pois atrasos como este, além da incapacidade de reagir diante da mudança, tornam impossível a entrega contínua.

A cobertura funcional dos dados de produção pode ser complementada por criação manual; entretanto, este é um método impreciso, moroso e nada científico. Não há uma forma de verificar se a cobertura máxima foi atingida nem se os dados permanecem intactos para fins de referência.

Altos custos

Por fim, além dos altos custos do retrabalho e da entrega tardia, a cópia dos dados de produção é um processo lento e dispendioso a ser evitado. Segundo relatórios, algumas organizações acumulam até 20 cópias de um único banco de dados; o resultado disso são gastos elevados com hardware, licenças e suporte.

Da mesma forma, o armazenamento de altos volumes de dados pode ser extremamente dispendioso. Uma pesquisa relacionada sobre armazenamento de dados realizada por terceiros revela que a migração de dados para soluções menos custosas, incluindo a nuvem, reduz temporariamente o custo de armazenamento, mas não soluciona nem trata dos problemas fundamentais de crescimento dos dados. Em outras palavras, qualquer diretiva de TDM cujo fim seja migrar, subdividir e mascarar dados não tem condições de oferecer uma solução econômica para o problema de como armazenar dados processados por aplicativos cada vez mais complexos, modernos e compostos.

Há também o perigo ainda mais dispendioso da não conformidade, algo sem solução mesmo com o mascaramento de dados. Um estudo independente revelou que o custo da violação de dados em 2014 sofreu um aumento médio de 15%, atingindo o patamar de US\$ 3,5 milhões por registro, enquanto a nova diretiva de proteção de dados da União Europeia (UE), a ser promulgada em 2016, prevê uma multa máxima cobrada sobre a violação de dados no montante equivalente a € 100 milhões² ou 5% da movimentação global, das duas a maior.

O mascaramento de dados de produção não garante a conformidade, pois o perigo real ainda é o erro humano, segundo o qual 50%³ das violações de dados podem estar relacionadas a atitudes internas. Além disso, no mascaramento, a integridade referencial dos dados deve permanecer intacta; de outra forma, poderá haver falhas nos testes de aplicativos. No entanto, quanto mais complexos os dados, mais fácil sofrer violações, pois mais informações podem estar correlacionadas.

A única forma genuína de evitar os altos custos de infraestrutura e impedir o vazamento de dados sigilosos é não usar dados de produção. Embora despesas como essas não sejam peculiares a organizações interessadas em implementar uma estrutura de entrega contínua, elas certamente contradizem seus princípios. Tais despesas tornam as organizações incapazes de disponibilizar software de qualidade capaz de se adequar rapidamente às dinâmicas demandas de negócios e manter-se dentro do orçamento.

Seção 3

Abordagem ao TDM com base em requisitos

As organizações interessadas em implementar a Entrega contínua devem repensar seus processos de testes e desenvolvimento, em vez de simplesmente reorganizá-los. É preciso reavaliar sua abordagem de TDM. Uma abordagem completa e abrangente ao Gerenciamento de dados de teste, orientada por requisitos, permitirá às organizações executar testes preliminares, reduzir riscos e minimizar a proliferação de falhas, produzindo software de qualidade mais rapidamente e gastando menos.

Em vez de considerar os dados com base nos resultados de cada caso de teste, as organizações deveriam avaliar os dados em termos das decisões de design, isto é, dos requisitos propriamente ditos, elaborando casos de teste com dados diretamente vinculados a eles. O ajuste de dados de teste aos requisitos garante sua "adequação a determinada finalidade", enquanto a capacidade de provisioná-los rapidamente às equipes de teste significa que estas estão aptas a lidar prontamente com as inconstantes demandas dos negócios.

Sendo assim, a Entrega contínua não pode ser iniciada em um estágio avançado do ciclo de vida do desenvolvimento de software: ela deve partir de uma ideia inspirada nos próprios requisitos; assim, as equipes de desenvolvimento poderão atender às necessidades dinâmicas de negócios, trabalhando a partir dessa ideia igualmente dinâmica. As equipes devem estar aptas a facilmente validar e verificar os requisitos, produzindo casos de teste vinculados aos resultados esperados e aos dados virtuais, diretamente destes. Testes como esses, com base em modelos, representam o que chamamos de "testar logo no início" e produzir software continuamente de modo a condensar todo o trabalho do ciclo de vida de desenvolvimento na etapa de coleta dos requisitos; com isso, todo o trabalho subsequente fluirá sem dificuldades a partir do trabalho preliminar, mesmo se houver mudanças nos requisitos.

Desenvolvimento de requisitos mais bem-elaborados

Por sua vez, os requisitos devem conter todas as informações qualitativas sobre um sistema a ser testado; assim, os testadores poderão produzir casos de uso e de teste diretamente da "ideia" inicial, introduzindo a rastreabilidade entre eles. Essa rastreabilidade é necessária se for preciso que os testadores atualizem rapidamente seus testes quando houver mudança nos requisitos.

Com as ferramentas adequadas, a modelagem de fluxogramas produz fluxogramas contendo todas as informações qualitativas sobre determinado sistema necessárias aos testes, independentemente da simplicidade de como o fluxo propriamente dito é elaborado. O CA Agile Requirements Designer (anteriormente, Grid Tools Agile Designer) oferece suporte à criação de fluxogramas contendo toda a lógica funcional necessária à geração do menor número de casos de teste possível com cobertura máxima, sem deixar a empresa "boquiaberta" como ficaria com os modelos de causa e efeito ou as abordagens pareadas².

Além disso, as chances de produzir requisitos inequívocos e completos são maiores com os fluxogramas. Eles rompem o "muro desigual das palavras" e os incômodos diagramas que normalmente decompõem os requisitos em pequenas porções digeríveis. Esses processos refletem a lógica da causa e efeito de um sistema e constituem, na verdade, uma série de afirmações "hipotéticas e conclusivas".

Isso não apenas ajuda a reduzir os 56%⁴ de falhas resultantes de ambiguidades nos requisitos, mas obriga as equipes responsáveis por estes a adotar uma lógica com base na modelagem de sistemas em termos de limitações, restrições e condições marginais. A partir desses requisitos, modelados na forma de um fluxograma, é possível então identificar qualquer provável caminho através de um sistema. Isso viabiliza a criação de casos de teste capazes de oferecer total cobertura funcional, isto é, os testes cobrem todos os caminhos possíveis através de um sistema, incluindo caminhos negativos e resultados inesperados³.

Com um fluxograma ativo, esses casos de teste podem ser ainda vinculados a métricas complexas, dados virtuais, dados de teste, scripts de automação, resultados esperados e pendências, concentrando o esforço do ciclo de vida do desenvolvimento de software na etapa de apuração dos requisitos.

Provisionamento de dados adequados a determinada finalidade

A automação dos processos de design de casos de teste com o uso de ferramentas como o CA Agile Requirements Designer permite aos testadores criar os casos de teste certos que serão necessários para testar exaustivamente os requisitos e garantir a satisfação do testador. Com isso, chegamos à argumentação central deste documento: a entrega contínua requer um melhor TDM. Os testadores devem ter acesso aos dados adequados a determinada finalidade com os quais poderão alcançar uma cobertura total dos casos de teste, enviados ao lugar certo, no momento certo.

Qualidade

Conforme discutido, dados de produção não podem oferecer a cobertura necessária para testar um sistema na íntegra. Por outro lado, a geração sintética de dados é capaz de produzir pequenos conjuntos complexos de dados que cobrem todos os prováveis cenários, mesmo que um evento jamais tenha ocorrido antes. Cada cenário real pode ser concebido como sendo outro ponto de dados; assim, é possível criar dados sintéticos até mesmo para cenários novos e futuros. O CA Test Data Manager (anteriormente, CA Data Finder ou Data Maker da Grid-Tools) emprega técnicas inteligentes de criação de perfis de dados para formar uma imagem precisa de um modelo de dados, gerando dados complexos e sofisticados que oferecem 100% de variações funcionais com base neles.

Com uma abordagem com base em requisitos, os dados gerados são correspondidos a um caso de teste, garantindo sua adequação para atender às necessidades de cada testador em particular. Essa "geração de dados de teste" proporciona mais chances de as equipes de teste detectarem falhas nas etapas iniciais, evitando o desgastante retrabalho que torna a entrega contínua impossível.

Tempo

O tempo gasto na criação manual ou manipulação de dados ou na espera por sua disponibilização contradiz os princípios do desenvolvimento contínuo, conforme discutimos.

Por outro lado, ferramentas automatizadas para a criação de dados, como o CA Test Data Manager, podem trabalhar diretamente com níveis de RDBMs ou APIs de ERP; dessa forma, os usuários terão meios de gerar dados na rapidez permitida por seu poder de processamento. Scripts volumosos podem duplicar a quantidade de dados de uma organização na mesma rapidez com que a infraestrutura de banco de dados pode lidar com eles. Isso garante a existência de dados adequados a determinada finalidade em questão de horas, não de semanas, ou seja, os testadores contarão com os dados de que precisam quando tiverem de usá-los em um sprint.

Uma funcionalidade de correspondência de testes eficaz implica também que, se os dados existem, é possível identificá-los e miná-los de várias fontes distintas antes de correspondê-los e aloca-los aos casos de teste adequados. Comparada aos processos manuais, essa correspondência de testes provou ser capaz de reduzir em 95% o tempo gasto na busca e provisionamento de dados e também oferece meios de as equipes determinarem quais testes irão falhar por problemas nos dados antes de serem executados.

O armazenamento de dados de teste em um data warehouse centralizado pode ainda evitar os gargalos provocados enquanto as equipes aguardam a disponibilização e a liberação dos dados. O portal de dados de teste sob demanda do CA Test Data Manager oferece criação dinâmica de formulários, permitindo aos usuários selecionar o tipo de dados desejado com base em critérios específicos, como o tipo de cartão de crédito ou a localização geográfica. O formulário faz a alocação dos dados necessários ao caso de teste ou cria novos dados se nenhum existir. Assim, os engenheiros terão mais tempo para eliminar as falhas reveladas nos testes, em vez de gastar 50% do seu tempo esperando pelo provisionamento dos dados. Da mesma forma, as organizações poderão centralizar o gerenciamento dos dados na equipe de segurança de TI e provisionar dados sigilosos apenas a pessoal autorizado, mediante solicitação.

Custo

Por fim, uma diretiva de TDM mais bem-estruturada reduz o risco dos excedentes de custo. A detecção e eliminação precoce de falhas podem reduzir sua proliferação em até 95%⁵ e gerar economias de mais de US\$ 50 mil⁵ por falha. Além disso, o uso de subconjuntos menores e mais complexos pode reduzir os custos de infraestrutura em até US\$ 50 mil⁵ por banco de dados, enquanto a execução de testes em menor número, mas com mais qualidade, pode também reduzir significativamente o tempo e os custos com testes. O perigo da dispendiosa não conformidade é igualmente eliminado com o uso de dados sintéticos, pois as informações sigilosas não precisam sair dos ambientes de produção.

Seção 4

Resumo

Reproduzir dados de produção como fonte da "verdade" em ambientes que não são de produção impede a implementação bem-sucedida da entrega contínua. Possivelmente, as equipes de teste se verão desprovidas de dados adequados a determinada finalidade quando estes forem necessários para executar testes completos de software. Em outras palavras, essas equipes estarão incapacitadas de se posicionar com rapidez diante das mudanças nos requisitos de negócios. Embora as ferramentas de subdivisão e mascaramento solucionem certos problemas, os dados de produção não têm qualidade suficiente para identificar e eliminar falhas.

A adoção de uma abordagem abrangente ao desenvolvimento de software com base em requisitos e ao tratamento dos dados de teste como valiosos ativos reutilizáveis são fundamentais à implementação bem-sucedida da entrega contínua. Ao trabalhar a partir do desenvolvimento de uma "ideia", as equipes poderão atender rapidamente às dinâmicas necessidades de negócios, utilizando dados adequados a determina finalidade, enviados ao lugar certo, no momento certo, para testar softwares integralmente, dentro do menor número possível de rodadas de teste necessárias.

Essa abordagem permite às organizações criar eficiências em seu provisionamento de dados de teste e, ao mesmo tempo, reduzir o risco de atrasos, retrabalho e custos incontroláveis que tornam impossível a Entrega contínua. A abordagem promove ainda a entrega contínua de software valioso capaz de suprir as necessidades fundamentais de negócios dentro do prazo e do orçamento.

Seção 5

Referências

- 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 Consulte Primer of A Critique of Testing de Llyr Wyn Jones para obter um tratamento conceitual e matemático das vantagens do modelamento de fluxogramas, disponível em <http://www.agile-designer.com/resources/critique-testing-primer/>
-

Seção 6

Sobre o autor

Cofundador da Grid-Tools e VP na CA Technologies



Eleito "Diretor de TI do ano de 2010" pela QA Guild, Huw Price foi o arquiteto técnico chefe em diversas empresas de software dos EUA e da Europa. Especialista em ferramentas de automação de testes, ele lançou inúmeros produtos inovadores que reformularam o modelo de testes usado na indústria de software.



Conecte-se à CA Technologies em ca.com/br



A CA Technologies (NASDAQ: CA) cria software que acelera a transformação das empresas e permite que elas aproveitem as oportunidades da economia dos aplicativos. O software está no cerne de todas as empresas, em todos os setores. Do planejamento ao desenvolvimento e do gerenciamento à segurança, a CA está trabalhando com empresas de todo o mundo para mudar a maneira como vivemos, fazemos negócios e nos comunicamos – usando dispositivos móveis, as nuvens privada e pública e os ambientes distribuídos e de mainframe. Para saber mais sobre os programas de sucesso de nossos clientes, visite ca.com/customer-success. Obtenha mais informações em ca.com/br.

- 1 Ponemon Institute: 2014 Custo da violação de dados
- 2 UE propôs regulamentação para a proteção de dados
- 3 Relatório de investigações de violações de dados de 2013
- 4 Relatório Bender: Testes com base em requisitos
- 5 Métricas obtidas a partir da experiência de implementação da Grid-Tools