Krok 11: Rozpakuj/przekształć/załaduj rozwój

Tutaj omówiono następujące tematy:

* Rzeczy do rozważenia przy opracowywaniu ekstrakcji/transformacji/ładowania (ETL)

* Typowe działania związane z transformacją danych i ciągłe niedocenianie wysiłków związanych z transformacją danych

* Trzy rodzaje sum uzgadniania, które powinny zostać wygenerowane w procesie ETL: liczba rekordów, liczba domen i liczba kwot

* Ocena współpracownika

* Sześć odpowiednich procedur testowania ETL: testy jednostkowe, testy integracyjne, testy regresyjne, testy wydajnościowe, testy zapewniania jakości (QA) i testy akceptacyjne

* Jak stworzyć formalny plan testów?

* Krótkie opisy działań związanych z rozwojem ETL, rezultatów wynikających z tych działań oraz zaangażowanych ról

* Ryzyko niewykonania kroku 11

Rzeczy do rozważenia

Ekstrakty danych źródłowych

* Kto napisze programy ETL? Czy ci programiści pisali wcześniej programy ETL? Czy rozumieją proces ETL?

* Czy programy ETL istnieją już w poprzedniej wersji lub innej aplikacji BI? Ile z nich trzeba rozbudować?

* Czy możemy poprosić programistów systemów operacyjnych o udostępnienie nam plików z rozpakowaniem, czy też musimy sami uzyskać dane źródłowe?

* Co musimy wiedzieć o systemach operacyjnych, zanim będziemy mogli uzyskać dane? Jakie programy operacyjne muszą zakończyć działanie, zanim będziemy mogli wyodrębnić dane z plików źródłowych i źródłowych baz danych?

Narzędzie ETL

* Czy pracowaliśmy wcześniej z tym narzędziem ETL, czy jest ono dla nas nowe?

* Czy zespół ETL został wystarczająco przeszkolony w zakresie narzędzia ETL?

* Czy narzędzie ETL może wykonać wszystkie wymagane przekształcenia, czy też będziemy musieli napisać własny kod? W jakim języku (C++, COBOL)?

Zależności procesu ETL

* Jakie są zależności między modułami programu? W jakiej kolejności musimy uruchamiać nasze programy ETL (lub narzędzie ETL)

moduły)?

* Ile modułów programu możemy uruchomić równolegle?

* Jakie są zależności między tabelami? Czy niektóre tabele muszą być ładowane przed innymi?

* Ile tabel możemy załadować równolegle?

Testowanie

* Czy będziemy przeprowadzać wzajemne oceny? Czy używamy ekstremalnych technik programowania (XP)?

* Ilu testerów będziemy mieli w projekcie?

* Czy ekspert merytoryczny i przedstawiciel biznesu będą uczestniczyć w testach?

* Kto będzie koordynatorem testów? Kto będzie rejestrować wyniki testów i prowadzić dziennik testów?

* Jakiego rodzaju testy musimy wykonać? Testy integracyjne czy regresyjne? Test wydajności? Testowanie jakości? Testy akceptacyjne?

* Którzy ludzie biznesu będą uczestniczyć w testach akceptacyjnych? Tylko przedstawiciel handlowy? Ekspert merytoryczny? Inni ludzie biznesu?

Rozważania techniczne

* Jakie problemy techniczne z platformą musimy wziąć pod uwagę?

* Jak jest skonfigurowany obszar pomostowy? Na serwerze dedykowanym?

* Czy proces ETL zostanie podzielony między komputer mainframe i jeden lub więcej serwerów?

* W jakich środowiskach działa narzędzie ETL?

* Jakiego typu oprogramowania pośredniczącego potrzebujemy?

Korzystanie z narzędzi ETL stało się bardzo rozpowszechnione, ale organizacje, które z nich korzystają, bardzo szybko odkrywają, że narzędzia te mają swoje ograniczenia. W zależności od złożoności wymaganych przekształceń danych źródłowych oraz wieku i stanu plików źródłowych często trzeba napisać niestandardowy kod, aby rozszerzyć funkcjonalność narzędzia ETL.

Transformacja danych źródłowych

Reguły techniczne i biznesowe dla wymaganych przekształceń danych źródłowych zostały zgromadzone i zdefiniowane na wszystkich etapach planowania projektu, definiowania wymagań projektowych, analizy danych, prototypowania aplikacji i analizy repozytorium metadanych. Podczas tych kroków reguły zostały prawdopodobnie wydobyte ze starych podręczników, starych notatek, wiadomości e-mail, programów (wspomagania operacyjnego i podejmowania decyzji) oraz narzędzi inżynierii oprogramowania wspomaganego komputerowo (CASE) i dostarczone przez ludzi, którzy pamiętają, kiedy i dlaczego reguła biznesowa powstał. Reguły te są teraz odzwierciedlane jako działania transformacji danych w procesie ETL.

Działania związane z transformacją danych

Projekty BI stanowią najlepszą okazję do wyeliminowania martwych i bezużytecznych danych, ponieważ pozwalają ludziom biznesu spojrzeć na ich wymagania informacyjne w innym świetle. Po prawidłowym wdrożeniu czynności przekształcania danych, takie jak czyszczenie, podsumowywanie, wyprowadzanie, agregacja i integracja, dadzą odpowiednio czyste, skondensowane, nowe, kompletne i ustandaryzowane dane

* Oczyszczanie: Z definicji czyszczenie jest procesem transformacji BI, w którym dane źródłowe, które naruszają reguły biznesowe, są zmieniane w celu dostosowania do tych reguł. Oczyszczanie jest zwykle realizowane poprzez edycję w programach ETL, które korzystają z wyszukiwań tabel i logiki programu w celu określenia lub uzyskania poprawnych wartości danych, a następnie zapisania tych wartości danych w plikach ładowania używanych do zapełniania docelowych baz danych BI.

* Podsumowanie: wartości liczbowe są sumowane w celu uzyskania całkowitych liczb (kwoty lub liczebności), które następnie mogą być przechowywane jako fakty biznesowe w wielowymiarowych tabelach faktów. Sumy podsumowujące można obliczać i przechowywać na wielu poziomach (np. podsumowanie sprzedaży wydziału, regionalne podsumowanie sprzedaży i łączna sprzedaż według kraju).

* Wyprowadzanie: podczas tego procesu tworzone są nowe dane z istniejących niepodzielnych (szczegółowych) danych źródłowych. Wyprowadzanie jest zazwyczaj realizowane przez obliczenia, przeglądanie tabel lub logikę programu. Przykłady obejmują:

- Generowanie nowego kodu do klasyfikacji klientów na podstawie określonej kombinacji istniejących wartości danych

- Obliczanie zysku z pozycji przychodów i kosztów

- Dołączanie ostatnich czterech cyfr kodu pocztowego na podstawie adresu w tabeli wyszukiwania pocztowego

- Obliczanie wieku klienta na podstawie jego daty urodzenia i bieżącego roku

* Agregacja: wszystkie dane dotyczące obiektu biznesowego są gromadzone. Na przykład elementy danych dla klienta mogą być agregowane z wielu plików źródłowych i źródłowych baz danych, takich jak plik Customer Master, plik Prospect, plik Sales i dane demograficzne zakupione od dostawcy. (W wielowymiarowym żargonie projektowania baz danych termin agregacja odnosi się również do zestawienia wartości danych).

* Integracja: Integracja danych oparta na regułach normalizacji wymusza konieczność uzgadniania różnych nazw danych i różnych wartości danych dla tego samego elementu danych. Pożądanym rezultatem jest posiadanie każdego unikalnego elementu danych znanego pod jedną standardową nazwą, z jedną standardową definicją i zatwierdzoną domeną. Każdy element danych powinien być również powiązany z jego plikami źródłowymi i źródłowymi bazami danych oraz docelowymi bazami danych BI. Standaryzacja danych powinna być celem biznesowym.

Niedocenianie wysiłków związanych z transformacją danych

Transformacja danych źródłowych jest podobna do otwierania rosyjskiej lalki - otwierasz jedną, a w środku jest druga. To może być niekończący się proces. Dlatego czas potrzebny na proces ETL jest chronicznie niedoszacowany. Pierwotne oszacowania są zwykle oparte na liczbie konwersji danych technicznych wymaganych do przekształcenia typów i długości danych i często nie uwzględniają przytłaczającej liczby przekształceń wymaganych do wymuszenia reguł domeny danych biznesowych i

reguł integralności danych biznesowych. Specyfikacje transformacji podane programiście ETL nigdy nie powinny ograniczać się tylko do technicznych reguł konwersji danych. W przypadku niektórych dużych organizacji z wieloma starymi strukturami plików, stosunek poszczególnych wysiłków związanych z transformacją danych może wynosić nawet 80% wysiłku związanego z wymuszeniem reguł domeny danych biznesowych i reguł integralności danych biznesowych i tylko 20% wysiłku związanego z wymuszeniem technicznych reguł konwersji danych. Dlatego spodziewaj się pomnożenia oryginalnych szacunków związanych z transformacją danych ETL przez cztery. Nawet jeśli uważasz, że masz bardzo realistyczny harmonogram procesu ETL, nie zdziw się, jeśli nadal nie dotrzymujesz terminów z powodu brudnych danych. Jeśli nie dotrzymujesz terminów, nie zdziw się, gdy odkryjesz, że nie oczyściłeś wystarczająco danych. Nalegaj na pełnoetatowe zaangażowanie przedstawiciela biznesowego i nalegaj na znalezienie odpowiedniego przedstawiciela biznesowego — kogoś, kto ma wiedzę na temat firmy i ma uprawnienia do podejmowania decyzji dotyczących zasad biznesowych. Te postanowienia są niezbędne do przyspieszenia procesu ETL. Ponadto nakłaniaj sponsora biznesowego i przedstawiciela biznesowego do uruchomienia inicjatywy w zakresie jakości danych w organizacji lub przynajmniej w działach pod ich kontrolą lub pod ich wpływem. Kiedy ludzie biznesu prowadzą inicjatywę dotyczącą jakości danych, są bardziej skłonni do pomocy w procesie transformacji ETL. Przypomnij im, że podczas gdy technicy IT mogą znać semantykę procesu, ludzie biznesu znają zawartość danych i semantykę biznesową. Rozumieją, co naprawdę oznaczają dane.

Pojednanie

Jedną z najczęstszych skarg dotyczących aplikacji BI jest to, że dane w docelowych bazach danych BI nie są zgodne z danymi w systemach źródłowych. W rezultacie ludzie biznesu często nie ufają danym BI. Jak na ironię, przez większość czasu dane w docelowych bazach danych BI są dokładniejsze niż dane w źródle operacyjnym plików lub źródłowych baz danych, ponieważ dane zostały ponownie sformatowane, ustandaryzowane i oczyszczone. Jednak bez dowodu tego zaufania nie można przywrócić. Sumy uzgadniania stanowią dowód i muszą być dostępne jako metadane w repozytorium metadanych. Sumy rozliczenia ETL to sumy kontroli procesu ETL, a nie sumy rozliczenia operacyjnego z powrotem do sprawozdania finansowego lub księgi głównej organizacji. Celem sum uzgadniania ETL jest zapewnienie, że wszystkie wartości danych przekazywane do procesu ETL mogą być uzgodnione ze wszystkimi wartościami danych wychodzącymi z procesu ETL. Przechowywanie sum uzgadniania ETL, a także statystyk jakości i obciążenia danych jako metadanych podkreśla znaczenie udostępnienia repozytorium metadanych. Te metadane są kluczową informacją dla ludzi biznesu, którzy chcą zobaczyć jakie dane zostały załadowane, które dane zostały odrzucone iz jakich powodów oraz jaka jest wiarygodność danych (czystość) współczynnik dotyczy danych BI po każdym cyklu ładowania. Na przykład dane źródłowe z wielu plików źródłowych i źródłowych baz danych mogły nie zostać prawidłowo zsynchronizowane (błąd terminowości), co mogło spowodować niespójności w wynikach analizy. Dzięki statystykom obciążenia dostępnym jako metadane analitycy biznesowi mogą szybko rozpoznać i rozwiązać problem.

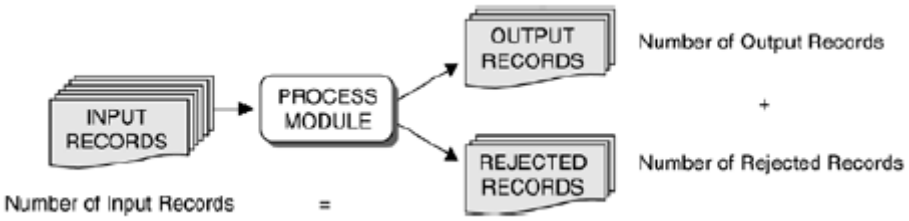
Obliczanie sum rozliczenia

Zdecydowanie zbyt wiele projektów BI nie wykorzystuje dobrych praktyk informatycznych powszechnie stosowanych w systemach operacyjnych. Poważnym błędem jest przekonanie, że aplikacje BI służą „tylko” do wspomagania decyzji i dlatego są mniej krytyczne niż systemy operacyjne. Aplikacje BI są tak samo krytyczne jak systemy operacyjne, ponieważ decyzje podejmowane na podstawie danych w środowisku wspomagania decyzji BI mogą wpływać na kierunek strategiczny i żywotność organizacji. Jedną ze sprawdzonych w czasie dyscypliny podczas manipulowania danymi, niezależnie od tego, czy w jakiś sposób zmieniasz dane, czy kopiujesz je lub przenosisz z jednego miejsca do drugiego, jest uzgadnianie każdego nowego wyjścia ze starym wejściem. Każdy program lub

moduł programu, który odczytuje dane, a następnie zapisuje je w nowym pliku, nawet jeśli tylko w pliku tymczasowym, musi generować sumy uzgodnienia. Istnieją trzy typy sum uzgodnienia: liczba rekordów, liczba domen i liczba kwot.

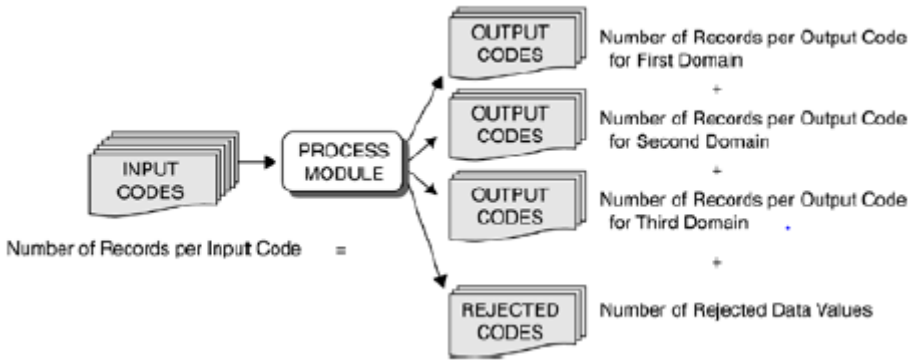
Liczba rekordów

Jedną z najbardziej podstawowych sum uzgadniających jest prosta liczba odczytanych i zapisanych rekordów. Jeśli rekordy są odrzucane, ponieważ nie przeszły kontroli edycji w procesie ETL, należy również policzyć liczbę odrzuconych rekordów. Całkowita liczba rekordów zapisanych i rekordów odrzuconych musi być równa liczbie rekordów odczytanych



Liczba domen

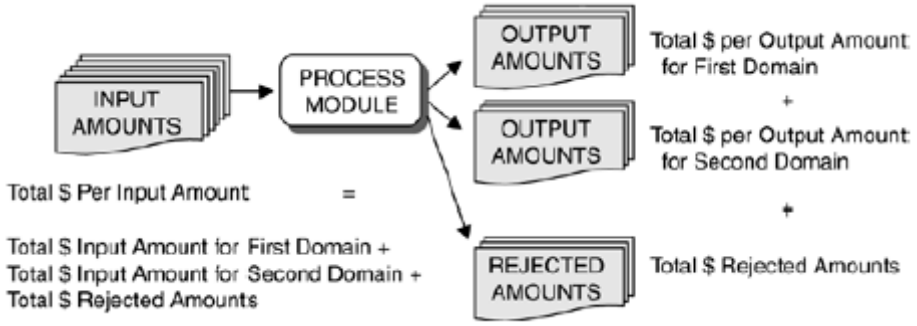
Liczenie domen obejmuje policzenie liczby rekordów dla każdej unikatowej domeny (wartości danych) pola wejściowego i zliczenie liczby rekordów dla tej samej unikalnej domeny w pliku wyjściowym. Komplikacja z liczbą domen pojawia się, gdy „przeciążony" element danych źródłowych musi zostać podzielony na wiele kolumn. Przeciążony źródłowy element danych to element danych używany do wielu celów, ponieważ wiele jedno- i dwubajtowych pól kodu znajduje się w systemach operacyjnych. Domena przeciążonego elementu danych opisuje nie jeden obiekt biznesowy, ale wiele obiektów biznesowych i dlatego musi być podzielona na różne kolumny dla różnych tabel. (Obiekty biznesowe są zwykle implementowane jako tabele wymiarów). W takim przypadku łączna liczba wielu domen po stronie wyjściowej musi być równa liczbie jednej domeny po stronie wejściowej. Jeśli wartości danych zostały odrzucone, ponieważ nie przeszły edycji jakości danych, pojawiłaby się dodatkowa liczba wszystkich odrzuconych rekordów. Zatem suma wielokrotnych zliczeń domen wyjściowych i zliczeń odrzuconych wartości danych musi być równa liczbie domen wejściowych



Liczba się liczy

Jeszcze ważniejsze od liczby domen są liczby. Są podstawowym mechanizmem uzgadniania plików źródłowych i źródłowych baz danych z docelowymi bazami danych BI. Uzgadnianie kwot odbywa się na dwa sposoby. Jednym z nich jest podsumowanie każdego pola kwoty w każdym pliku wejściowym i każdego odpowiadającego mu pola kwoty w każdym pliku wyjściowym. Jeśli rekordy zostałyby odrzucone, pojawiłoby się trzecie podsumowanie dla odrzuconej kwoty całkowitej. Łączna suma

całkowitej kwoty wyjściowej i całkowitej kwoty odrzuconej musi być równa całkowitej kwocie wejściowej. Bardziej skomplikowany algorytm uzgadniania kwot musi zostać opracowany, jeśli pole kwoty przychodzącej jest przeciążonym elementem danych źródłowych, który należy podzielić na kilka różnych kolumn kwot dla docelowych baz danych BI. W takim przypadku kryteria wyboru i edycji w specyfikacjach przekształcenia muszą być użyte do utworzenia wielu sum wyjściowych. Ponadto te same kryteria wyboru i edycji muszą zostać uruchomione w pliku wejściowym, aby wygenerować te same wielokrotne sumy kwot do weryfikacji



Przechowywanie statystyk uzgodnień

Ze względu na dużą ilość transformacji i czyszczenia, które zwykle występują w procesie ETL, ludzie biznesu powinni oczekiwać, że dane w docelowych bazach danych BI będą inne niż w oryginalnych plikach źródłowych i źródłowych bazach danych. Ich chęć lub potrzeba uzgodnienia danych między źródłem a celem jest słuszna. Jednak należy to osiągnąć za pomocą sum uzgodnienia, a nie porównując elementy danych źródłowych z docelowymi kolumnami dolar za dolar. W związku z tym wszystkie sumy uzgodnień wygenerowane przez każdy moduł programu dla każdego cyklu ładowania aplikacji BI muszą być przechowywane jako metadane w repozytorium metadanych. Jeśli to możliwe, należy rozważyć dodatkową kategoryzację odrzuconych rekordów z krótkimi opisami powodu odrzucenia. Opisy te mogą być tworzone na podstawie kryteriów edycji ETL.

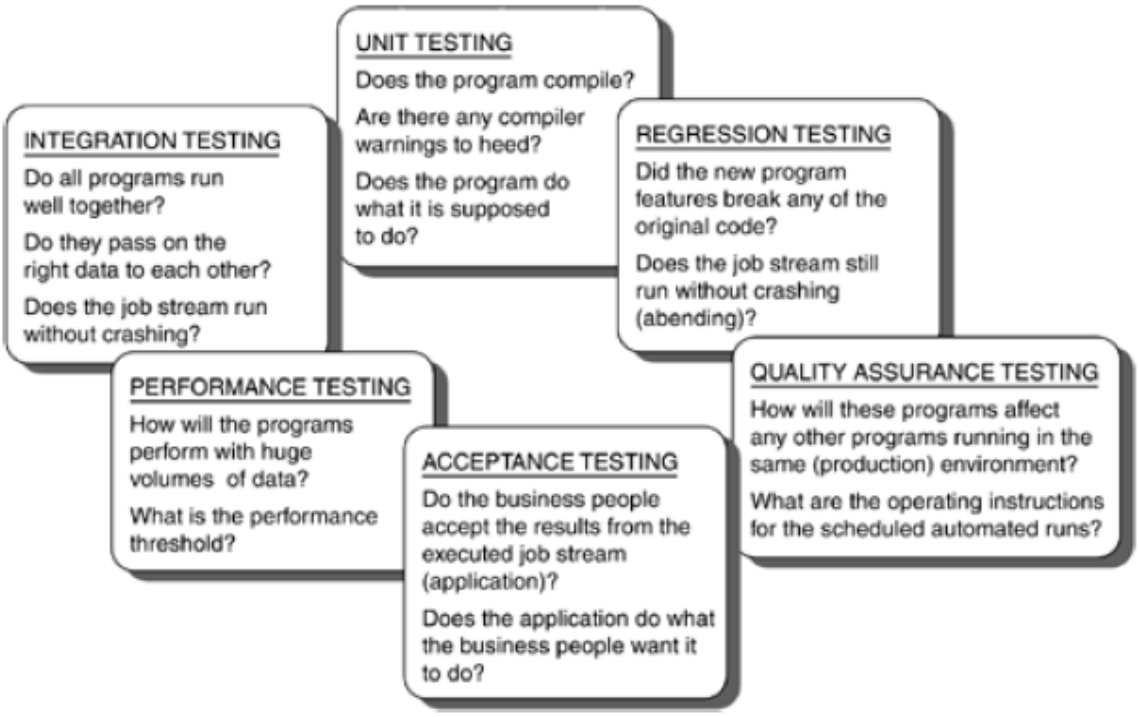
Ocena współpracownika

Przeglądy partnerskie są podobne do koncepcji programowania ekstremalnego (XP) „programowania w parach", z wyjątkiem tego, że samo kodowanie nie występuje w parach, spotkania często obejmują więcej niż dwie osoby, a przeglądy dotyczą wszystkich rodzajów rezultatów zadań poza programami . Przeglądy partnerskie to nieformalne spotkania, które łączą przeglądanie, testowanie i ograniczoną ilość burzy mózgów na temat dostarczanych zadań projektowych. Celem przeglądów partnerskich jest umożliwienie twórcy danej części pracy nad projektem przedstawienia zadania, które można dostarczyć jego współpracownikom do walidacji lub dyskusji. Najlepiej jest to osiągnąć w środowisku podstawowego zespołu, ale w ocenie wzajemnej mogą również uczestniczyć członkowie zespołu rozszerzonego, którzy są żywotnie zainteresowani tym zadaniem projektowym. Po zakończeniu zadania projektowego twórca pracy (lub jakikolwiek inny członek głównego zespołu) może zdecydować się poprosić o wzajemną recenzję, aby uzyskać wkład w złożony problem, uzyskać informację zwrotną na temat kreatywnego rozwiązania lub poprosić o opinie na temat element, który jest niepewny lub słabo zdefiniowany. Następnie określa, którzy członkowie zespołu powinni uczestniczyć w przeglądzie, ustala harmonogram nieformalnego spotkania przeglądowego i dystrybuuje kopie dokumentów (np. specyfikacji, programów, modeli danych, raportów) do zaproszonych współpracowników przed spotkaniem. Spotkanie prowadzi deweloper pracy. Kierownik projektu, który zawsze powinien uczestniczyć w wzajemnych ocenach, zapewnia, że spotkanie nie ugrzęźnie w sporach ani nie oddali się od omawianych zagadnień. Każdy uczestnik powinien zapoznać się z przekazanymi dokumentami i

powinien być przygotowany do komentowania i burzy mózgów na ten temat. Uczestnicy wspólnie odpowiadają za wykrycie błędów lub nieporozumień w dostarczanym zadaniu. Rozmowy są inicjowane, aby pomóc deweloperowi zrozumieć wykryte problemy lub umożliwić deweloperowi uzasadnienie swojej pracy w sposób zadowalający wszystkich obecnych. Burza mózgów zwykle ogranicza się do znalezienia błędów lub nieporozumień; nie obejmuje ich rozwiązania, chyba że rozwiązanie jest oczywiste i można je przedstawić lub omówić w ciągu kilku minut. Po zakończeniu przeglądu partnerskiego wykryte błędy lub nieporozumienia powinny zostać udokumentowane, a twórca produktu ma za zadanie ich poprawienie lub stworzenie nowego rozwiązania zadania projektowego. Recenzje recenzentów brzmią bardziej formalnie niż w rzeczywistości. Można je najlepiej porównać do ustrukturyzowanych sesji burzy mózgów o dużej mocy. Spotkania recenzyjne powinny być ograniczone do około godziny.

Testowanie ETL

Niestety, testowanie, takie jak uzgadnianie, jest często wykonywane bardzo słabo w projektach BI, jeśli w ogóle. To jest nie do przyjęcia – i nie jest też wymówką, że „można to naprawić w następnym wydaniu". Następna wersja będzie jeszcze większa i bardziej skomplikowana, a jej przetestowanie będzie wymagało więcej czasu. Innymi słowy, jeśli testowanie teraz trwa zbyt długo, testowanie trwa jeszcze dłużej, co zwykle oznacza, że nigdy nie przeprowadza się odpowiednich testów. Pędzenie do wdrożenia kosztem testowania jest niewybaczalne, zwłaszcza jeśli odbywa się to w celu dotrzymania sztucznego terminu. Te same typy testów, które dotyczą systemów operacyjnych, dotyczą również aplikacji BI. Poniższe sekcje krótko opisują każdy typ.



Testów jednostkowych

Testowanie jednostkowe odnosi się do testowania dyskretnych modułów programu i skryptów, a nie do testowania przepływu lub powiązań między programami. Każdy programista musi testować swoje

moduły programu i skrypty indywidualnie (jeśli wykorzystywane są techniki XP, odbywa się to w parach). Testy jednostkowe składają się z trzech komponentów: kompilacji, funkcjonalności i edycji.

Kompilacja: Oczywiście każdy moduł programu musi się pomyślnie skompilować lub nie można go zaimplementować. Na rynku istnieje wiele narzędzi testujących dla każdego możliwego języka programowania. Narzędzia te umożliwiają programiście śledzenie każdego kroku kodu podczas jego wykonywania, wyświetlając obrazy przed i po danych dla każdego wiersza kodu.

Funkcjonalność: Każdy moduł programu musi wykonywać funkcje, dla których został zaprojektowany i musi dawać oczekiwane wyniki testów. Dlatego każdy programista musi utworzyć mały plik testowy z prawidłowymi i nieprawidłowymi danymi, aby przetestować każdą funkcję swoich modułów programu. Musi z góry wiedzieć, co moduły programu powinny zrobić z prawidłowymi i nieważnymi danymi testowymi (oczekiwanymi wynikami testów). Testy jednostkowe nie są zakończone, dopóki wszystkie moduły programu nie przyniosą wszystkich oczekiwanych wyników.

Edycje: każdy moduł programu musi wyłapać błędy i, w zależności od wagi błędu, wygenerować komunikat o błędzie lub łagodnie zakończyć program. Żaden program nigdy nie powinien się nagle zatrzymywać ("awaria" lub awaria) z tajemniczym komunikatem o błędzie systemu. Ze względu na wysoki stopień niskiej jakości danych w plikach źródłowych dość często zdarza się, że więcej linii kodu jest pisanych do edycji niż do funkcji.

Rozważ zezwolenie na testowanie dowolnego fragmentu kodu przez kogoś innego niż programista. To, wraz z recenzją, powinno wyłapać większość błędów, których nie można wychwycić z powodu dumy z własności i zbytniego zbliżenia się do własnego kodu.

Jeśli używane jest narzędzie ETL, testowanie jednostkowe nadal dotyczy poszczególnych modułów narzędzi ETL. W takim przypadku testujesz ważność swoich instrukcji ETL, czyli technicznych metadanych ETL. Ważne jest, aby pamiętać, że jeśli narzędzie ETL nie spełnia standardów Twojej organizacji, będziesz musiał uzupełnić narzędzie o własny, napisany na zamówienie kod. W takim przypadku testowanie jednostkowe będzie połączeniem testowania metadanych technicznych ETL, testowania kodu napisanego na zamówienie i testowania „uścisku dłoni” między nimi.

Testy integracyjne

Testowanie integracyjne, znane również jako testowanie systemowe, jest pierwszym kompletnym uruchomieniem procesu ETL. Obejmuje to wszystkie trzy zestawy procesów ETL: ładowanie początkowe, ładowanie historyczne i ładowanie przyrostowe. Tylko dlatego, że wszystkie moduły programu przechodzą indywidualne testy jednostkowe, nie można zakładać, że cały proces ETL będzie przebiegał płynnie. Interakcje i przepływ wszystkich programów, jak określono w diagramie przebiegu procesu ETL, muszą być obserwowane i sprawdzane. Interakcje: Moduły programu odbierają dane, manipulują nimi i przekazują je innym modułom programu. Ta interakcja między modułami musi zostać przetestowana. Dane testowe używane do testowania integracyjnego różnią się od danych używanych do testowania jednostkowego. Do testowania integracji używana jest kopia stosunkowo dużego podzbioru reprezentatywnych operacyjnych danych źródłowych.

Przepływ: Diagram przepływu procesu ETL powinien wskazywać, które programy muszą być uruchamiane w jakiej kolejności, które programy mogą działać równolegle oraz gdzie wtrącone są narzędzia sortowania i scalania. Ten przepływ należy przetestować pod kątem funkcjonalności i wydajności. Testowanie funkcjonalności zapewnia, że właściwy proces jest wykonywany na właściwych danych we właściwym czasie, to znaczy, że programy działają we właściwej kolejności. Testowanie

wydajności zapewnia, że cały proces ETL może zakończyć się w oczekiwanym czasie. Jeśli nie, należy przeprojektować przepływ i ponownie przetestować cały proces ETL.

Jeśli używane jest narzędzie ETL, cały proces ETL musi zostać przetestowany od początku do końca, z wyjątkiem tego, że uruchamiane są procesy narzędzia ETL, a nie programy napisane przez użytkownika. Przedstawiciel biznesowy i ekspert merytoryczny powinni być zaangażowani w testy integracyjne. Jako pierwsi dowiadują się, czy dany bieg był udany, czy nie. Jest to również doskonała okazja do przeprowadzenia dla nich zaawansowanych szkoleń, które pozwolą rozwiać wszelkie przyszłe podejrzenia dotyczące dokładności i jakości danych w docelowych bazach danych BI. Testowanie integracyjne, podobnie jak testowanie jednostkowe, wymaga wielu przebiegów testowych w celu usunięcia wszystkich defektów i dostrojenia przepływu. Za każdym razem, gdy rzeczywiste wyniki testu nie odpowiadają oczekiwanym wynikom testu, program powodujący błąd musi zostać poprawiony, a wszystkie programy muszą zostać ponownie uruchomione. Jednak w przeciwieństwie do testów jednostkowych, testowanie integracyjne jest zbyt skomplikowane, aby można je było przeprowadzić bez formalnego planu testów, który powinien zawierać opis przypadków testowych i kolejność wykonywania programów.

Testowanie regresji

Najbardziej skomplikowanym i najbardziej czasochłonnym ze wszystkich rodzajów testowania jest testowanie regresyjne. Przypomina to testowanie integracyjne, ale tym razem testowane programy nie są nowe. Ponieważ środowisko wspomagania decyzji BI ewoluuje, a nowe aplikacje BI są stale dodawane do procesu ETL, zespół projektowy będzie musiał przeprowadzić obszerne testy regresji we wszystkich wydaniach z wyjątkiem pierwszego. Z każdym nowym wydaniem BI proces ETL musi być modyfikowany (ulepszany) w celu wydobycia większej ilości danych z systemów operacyjnych dla nowej aplikacji BI. Nowa aplikacja BI może mieć osobny zestaw programów dostępu do danych i analizy, ale proces ETL jest współdzielony. Głównym celem testowania regresji jest upewnienie się, że modyfikacje istniejących programów ETL nie spowodowały przypadkowo pewnych błędów, które wcześniej nie istniały. Jeśli nowe programy zostały dodane do przebiegu procesu ETL, nowe interakcje między programami muszą zostać przetestowane, a wszystkie kolejne stare programy, których dotyczy problem, muszą zostać ponownie przetestowane. Jeśli przepływ procesu ETL musiał zostać ulepszony lub przeprojektowany w celu zwiększenia wydajności, cały proces ETL musi zostać ponownie przetestowany.

Opracowanie nowego planu testów dla każdego testu regresji byłoby zbyt czasochłonne. Dlatego wskazane jest zapisanie oryginalnego planu testów oraz oryginalnych danych testowych stworzonych na potrzeby testów integracyjnych pierwszego wydania. Następnie ulepsz oryginalny plan testów dla kolejnych testów regresji o nowe programy, nowe dane i nowe przypadki testowe.

Test wydajności

Testy wydajności, znane również jako testy warunków skrajnych, są przeprowadzane w celu przewidywania zachowania i wydajności systemu. Jest to bardzo podobne do tradycyjnego testowania wydajności w systemach operacyjnych, ale testowanie wydajności BI jest bardziej skomplikowane ze względu na ogromne ilości danych w docelowych bazach danych BI. Ponieważ większość organizacji nie ma więcej niż trzy do czterech godzin w swoich nocnych oknach wsadowych, celem jest ustalenie, ile danych można przetworzyć w tym czasie i ile nocy zajmie ukończenie całego procesu ETL. W przeciwieństwie do testów integracyjnych lub testów regresyjnych, testowanie wydajności nie musi być wykonywane na każdym module programu. Testowanie wydajności może być ograniczone tylko do najbardziej krytycznych modułów programu o największej ilości danych i najdłuższym czasie wykonywania. Oprócz przeprowadzania testów fizycznych można korzystać z narzędzi do symulacji

testów warunków skrajnych. Te narzędzia symulacyjne pozwalają opisać platformę produkcyjną, w tym inne programy działające na tym samym serwerze i współdzielące tę samą przestrzeń. Na podstawie Twoich danych narzędzia obliczają i projektują szacunkowe dane dotyczące wydajności. Zdecydowanie zaleca się przeprowadzenie symulacji przed rzeczywistym testowaniem wydajności na rzeczywistych danych.

Testy zapewnienia jakości

Większość dużych organizacji ma ścisłe procedury przenoszenia aplikacji do środowiska produkcyjnego. Procedury te zazwyczaj obejmują testy QA, aw większości przypadków dla takich testów tworzone jest oddzielne środowisko QA. Personel operacyjny kieruje programistami w zakresie przenoszenia baz danych i programów do środowiska QA. Następnie wszystkie instrukcje obsługi i zaplanowane prace należy przekazać personelowi operacyjnemu w celu przetestowania. Przejdą symulowany przebieg produkcyjny, zanim umożliwią przeniesienie komponentów aplikacji do środowiska produkcyjnego.

Testy akceptacyjne

Testy akceptacyjne można przeprowadzić na dwa sposoby, w zależności od konfiguracji testowania jako całości. Najlepiej byłoby, gdyby istniało oddzielne środowisko testów akceptacyjnych, które można by również wykorzystać do testowania regresji przyszłych wydań. Dzięki oddzielnemu środowisku testów akceptacyjnych, testy QA i testy akceptacyjne mogą być wykonywane w tym samym czasie. Jednak utrzymywanie oddzielnego środowiska testów akceptacyjnych może nie być wykonalne lub uzasadnione. Prostszą alternatywą jest wykonanie testów akceptacyjnych po testach QA w tym samym środowisku QA. Jeśli przedstawiciel biznesowy aktywnie uczestniczył w testach integracyjnych lub testach regresyjnych, podczas testów akceptacyjnych powinno być bardzo niewiele niespodzianek. W rzeczywistości, jeśli przedstawiciel biznesowy czuje się komfortowo z wynikami testów integracyjnych lub regresyjnych i z wyjątkiem nieprzewidzianych problemów wykrytych podczas testów QA, oddzielne testy akceptacyjne mogą w ogóle nie być konieczne. Jeśli jednak zastosowano tradycyjne podejście, w którym przedstawiciel biznesowy nie był zaangażowany w żadne czynności związane z projektowaniem lub testowaniem, z wyjątkiem sporadycznych przeglądów, testowanie akceptacyjne jest najważniejszym testem ze wszystkich. Niektóre zespoły projektowe ograniczają testy akceptacyjne do części dostępowej i analitycznej aplikacji BI i wykluczają przedstawiciela biznesowego z testów ETL. To duży błąd. Kiedy analitycy biznesowi i menedżerowie biznesowi skarżą się na nieprawidłowe dane w docelowych bazach danych BI, przyczyną może nie być to, że programy raportujące nie działają poprawnie, ale że proces ETL jest wadliwy. Dlatego testowanie, jak poprawnie wprowadzić dane do docelowych baz danych BI, jest ważniejsze niż testowanie, jak je poprawnie pobrać, ponieważ błąd w programie raportującym jest znacznie łatwiejszy do znalezienia i naprawienia niż błąd w procesie ETL. Ponadto, ponieważ przedstawiciel biznesowy jest zaangażowany w analizę danych źródłowych i dostarczanie reguł biznesowych do czyszczenia danych, logiczne jest, że powinien przetestować proces ETL, który implementuje te reguły. Przedstawiciel handlowy powinien zadać niektóre z poniższych pytań.

* Czy pobierane są odpowiednie dane?

* Jeśli element danych źródłowych jest podzielony na wiele kolumn, czy jest to wykonywane prawidłowo podczas procesu transformacji?

* Jeśli niektóre elementy danych zostaną połączone, czy w wyniku tego procesu transformacji wynikły jakiekolwiek problemy z integralnością?

* Czy dane są poprawnie ładowane do odpowiednich docelowych baz danych BI i odpowiednich tabel BI?

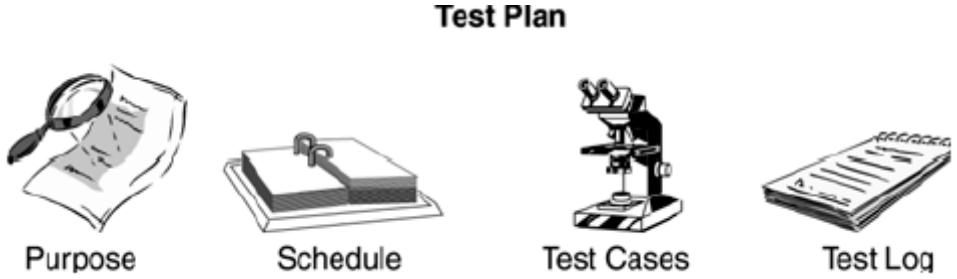
* Czy dane w docelowych bazach danych BI można pogodzić z plikami źródłowymi i źródłowymi bazami danych? Gdzie są przechowywane sumy uzgodnień?

* Czy wartości danych są poprawnie przekształcone i oczyszczone? Czy złe dane prześlizgują się bez powiadomienia?

* Czy wydajność obciążenia jest odpowiednia? I czy dane BI są dostępne dla ludzi biznesu, kiedy tego oczekują?

Formalny plan testów

Z możliwymi wyjątkami testów jednostkowych i wydajnościowych, sesje testowe ETL są zorganizowanymi wydarzeniami, które są prowadzone i kontrolowane przez program zwany planem testów. Każdy plan testów powinien określać informacje przedstawione na rysunku.



* Cel: ogólny opis tego, co jest testowane. Na przykład: „Dane są wyodrębniane z pliku głównego klienta, pliku transakcji konta, bazy danych historii sprzedaży i bazy danych produktu głównego. Dane wyodrębnione z pliku głównego klienta, transakcji konta i historii sprzedaży muszą zostać scalone w ramach numeru klienta, a wyodrębnione dane z Product Master i Sales History muszą być połączone w Product Number. Specyfikacje programowania ETL obejmują 28 transformacji i 46 algorytmów czyszczących. Przeprowadzimy 20 przypadków testowych, aby przetestować transformacje i 42 przypadki testowe, aby przetestować algorytmy czyszczące."

* Harmonogram: Po zapoznaniu się ze schematem przebiegu procesu ETL i ustaleniu, które programy muszą być uruchamiane w jakiej kolejności, a które mogą działać równolegle, każdy program w strumieniu zadań musi być zaplanowany tak, aby uruchamiał się dokładnie w tej kolejności w określonych terminach w określonych godzinach.

* Przypadki testowe: większość planu testów będzie stanowić lista przypadków testowych. Ważne jest, aby przedstawiciel biznesowy uczestniczył w pisaniu przypadków testowych. Każdy przypadek testowy określa kryteria wejściowe i oczekiwane wyniki wyjściowe dla każdego przebiegu. Opisuje również logikę programu do wykonania i wygląd danych wynikowych. Na przykład: „Prześlij moduł ETL3.3 przy użyciu tymczasowego pliku VSAM T11Customer i tymczasowego pliku VSAM T11Product. Oba pliki tymczasowe są sortowane w kolejności malejącej. Moduł ETL3.3 łączy dwa pliki i odrzuca rekordy, które nie pasują do Sale-Tran -Cd i Cust-Num. Wszystkie odrzucone rekordy muszą wywołać komunikat o błędzie: 'ETL3.3.e7 Nie znaleziono dopasowania w <wydrukuj Cust-Num> i <wydrukuj Prd-Nbr> <wydrukuj systemową sygnaturę daty i czasu> ”.

* Dziennik testów: Szczegółowa ścieżka audytu musi być przechowywana dla wszystkich przebiegów testów, z wyszczególnieniem daty i godziny uruchomienia programów, numerów programów lub modułów programu, kto je zatwierdził, oczekiwanych wyników testów, rzeczywistych wyników testów, czy test był akceptowane lub nie, oraz wszelkie dodatkowe uwagi.

Pamiętaj, że wszystkie programy w procesie ETL są testowane i ponownie testowane, aż cały proces ETL zostanie uruchomiony zgodnie z oczekiwaniami od początku do końca.

Działania rozwojowe ETL

Działania na rzecz rozwoju ETL nie muszą być wykonywane liniowo. Poniższa lista krótko opisuje czynności związane z Krokiem 11, Rozwój ETL.

1. Budowanie i testowanie jednostkowe procesu ETL. Pod kierunkiem głównego programisty ETL programy ETL muszą być opracowywane dla trzech zestawów procesów ładowania: ładowania początkowego, ładowania historycznego i ładowania przyrostowego. Jeśli planujesz używać narzędzia do ładowania systemu zarządzania bazami danych (DBMS) do zapełniania docelowych baz danych BI, wówczas należy napisać tylko programy wyodrębniania i transformacji, w tym programy, które tworzą końcowe pliki ładowania. Jeśli planujesz używać narzędzia ETL, musisz utworzyć instrukcje (metadane techniczne) dla narzędzia ETL. Wszystkie napisane na zamówienie programy ETL i wszystkie moduły narzędzi ETL muszą być testowane jednostkowo pod kątem kompilacji, funkcjonalności i edycji.

2. Test integracji lub regresji procesu ETL. Po przetestowaniu jednostkowym wszystkich indywidualnych programów ETL lub modułów programu, należy przetestować cały przebieg procesu ETL. Odbywa się to za pomocą testów integracyjnych w pierwszym wydaniu i testów regresji w kolejnych wydaniach. Oba typy testowania muszą być wykonywane w ramach formalnego planu testów z przypadkami testowymi, oczekiwanymi wynikami testów, rzeczywistymi wynikami testów oraz dziennikiem przebiegów testów.

3. Test wydajności procesu ETL. Ponieważ wiele docelowych baz danych BI to bardzo duże bazy danych (VLDB), ważne jest, aby przetestować wybrane programy lub moduły narzędzi ETL. Przeprowadzaj testy warunków skrajnych z danymi o pełnej objętości w tych programach lub modułach narzędzi ETL, które odczytują lub zapisują tabele o dużej objętości i wykonują skomplikowane operacje, zwłaszcza podczas równoległej pracy z tabelami o dużej objętości. Testy wydajności można również symulować za pomocą narzędzi do symulacji testów warunków skrajnych.

4. Test zapewnienia jakości procesu ETL. Większość organizacji nie zezwala na przenoszenie programów do środowiska produkcyjnego, dopóki nie przejdą one procesu testowania kontroli jakości. Ten test jest zwykle przeprowadzany w ramach nadzoru nad personelem operacyjnym w odrębnym środowisku QA.

5. Test akceptacyjny procesu ETL. Jeśli przedstawiciel biznesowy i ekspert w danej dziedzinie byli aktywnie zaangażowani w czynności związane z testowaniem integracyjnym lub regresyjnym, wtedy testowanie akceptacyjne powinno być niewiele więcej niż ostateczną, formalną certyfikacją od przedstawiciela biznesowego. Jeśli nie byli zaangażowani, wszystkie funkcje procesu ETL muszą zostać zweryfikowane pod kątem kompletności i poprawności, w szczególności proces uzgadniania.

Rezultaty wynikające z tych działań

1.Plan testów ETL. Plan testów powinien określać cel każdego testu i przedstawiać harmonogram uruchamiania testów we wstępnie zdefiniowanej kolejności. Powinien również opisywać przypadki testowe, w tym kryteria wejściowe i oczekiwane wyniki wyjściowe. Dziennik testów powinien

towarzyszyć planowi testów, dokumentując, kiedy testy zostały przeprowadzone, kto je przeprowadził i jakie były wyniki testów.

2. Programy ETL . Wszystkie programy i skrypty wyodrębniające, przekształcające i ładujące dla całego procesu ETL powinny być zakodowane i przetestowane. Jeśli używane jest narzędzie ETL, należy napisać instrukcje dotyczące modułów narzędzi ETL i przetestować moduły narzędzi ETL.

3. Biblioteka programów ETL. Wszystkie programy, skrypty i moduły narzędzi ETL ETL powinny znajdować się w bibliotece programów ETL z kontrolą wersji lub w bibliotece narzędzi ETL. Te programy, skrypty i moduły narzędzi ETL powinny być poddane testom integracji lub regresji, wydajności, kontroli jakości i akceptacji dla całego ET

Role zaangażowane w te działania

* Przedstawiciel firmy . Przedstawiciel biznesowy powinien brać udział w testach integracyjnych lub regresyjnych oraz testach akceptacyjnych. Z pomocą głównego programisty ETL wraz z ekspertem merytorycznym piszą przypadki testowe.

* Administrator bazy danych. Administrator bazy danych może być bardzo pomocny w procesie rozwoju ETL. Administrator bazy danych pomaga głównemu programiście ETL w przepływie procesu ETL, a także przegląda wszystkie wywołania bazy danych napisane przez programistów ETL. Niejednokrotnie administrator bazy danych

był w stanie usprawnić proces ETL, wywołując mało znane narzędzia DBMS w odpowiednim punkcie przepływu procesu ETL.

* Deweloperzy ETL. Jednym z głównych zadań przypisanych programistom ETL jest kodowanie lub ulepszanie programów ETL oraz ich testowanie jednostkowe. Jeśli organizacja korzysta z narzędzia ETL, programiści ETL muszą napisać instrukcje ETL (metadane techniczne) dla procesów narzędzia ETL.

* Główny programista ETL. Główny programista ETL zarządza całym procesem ETL. Razem z innymi programistami ETL przegląda schemat przebiegu procesu ETL oraz dokument projektu programu ETL i przypisuje im moduły programistyczne. Tworzy plan testów i współpracuje z przedstawicielem biznesowym oraz ekspertem merytorycznym przy tworzeniu przypadków testowych. Jest również odpowiedzialny za koordynację przebiegów testów i utrzymywanie aktualnego dziennika testów.

* Ekspert merytoryczny. Ekspert w danej dziedzinie, samodzielnie lub z przedstawicielem biznesowym, pisze lub ulepsza przypadki testowe dla planu testów. Sugeruje również, jakie sumy uzgadniania muszą zostać wytworzone. Ekspert w danej dziedzinie powinien uczestniczyć jako tester podczas testów integracyjnych lub regresyjnych oraz podczas testów akceptacyjnych.

* Testerzy. Testerami mogą być programiści, analitycy systemów, „zaawansowani użytkownicy", eksperci merytoryczni i każdy, kto ma pewne umiejętności techniczne i jest dostępny do udziału w testowaniu. Programiści nie powinni testować własnego kodu, ale mogą testować kod napisany przez innych programistów. Testowanie to czynność, którą można łatwo podzielić na moduły.

Wskazane jest zaangażowanie jak największej liczby testerów, aby przyspieszyć proces testowania.

Ryzyko niewykonania kroku 11

Dobrze zaprojektowany i przetestowany proces ETL jest podstawą środowiska wspomagania decyzji BI. Ten krok jest bardzo czasochłonny, ale bez niego nie masz aplikacji BI. Koniec opowieści.