

WIARYGODNOŚĆ BAZ DANYCH W MIGRACYJNYCH PROJEKTACH ERP

Jarosław BADUREK

Streszczenie: Artykuł dyskutuje strategie migracyjne ERP w kontekście rosnącej złożoności tych systemów. Pokazano znaczenie jakości danych referencyjnych przedsiębiorstwa, wskazując na warunki powodzenia projektów migracyjnych z uwzględnieniem aspektów pozatechnicznych. Wywód zilustrowano szeregiem przykładów z praktyki przemysłowej.

Słowa kluczowe: ERP, jakość danych, bazy danych, strategie migracji, modele referencyjne, SOA, BPM, MDM.

1. Złożoność organizacyjno-techniczna systemów gospodarczych

Tworzenie złożonego systemu informatycznego jest poważnym wyzwaniem dla jego projektantów. Jeszcze większą trudność stanowi wprowadzanie złożonych zmian w już istniejącym (złożonym) systemie. Poziom złożoności zadania ulega tutaj systemowemu spotęgowaniu, tzn. jest wyższy niż odrębne konstruowanie systemów niezależnych. Kolejny, trzeci poziom organizacyjnej i technicznej złożoności mogą stanowić migracje złożonych systemów IT, np. ERP. Tryb warunkowy tak sformułowanej tezy wiąże się z różnymi rodzajami migracji:

1. Konwersja (*convert*) - zmiany w kodzie aplikacji przy zachowaniu jej funkcjonalności i poprawie parametrów eksploatacyjnych.
2. Emulacja (*emulate*) - podobna organizacja i funkcjonalność w nowym środowisku.
3. Przepisanie (*rewrite*) - całkowita zmiana kodu aplikacji przy zachowaniu jej funkcjonalności.
4. Zastąpienie (*replace*) - nowa aplikacja, nowa technologia, nowa funkcjonalność.
5. Integracja (*integrate*) - rozszerzona wersja zastępowania dla grup aplikacji (ERP).
6. Przeprogramowanie (*reprogramming*) - nowe aplikacje, nowe procesy organizacyjne, nowe modele danych.

Widzimy bowiem, że w pierwszym przypadku mamy możliwość redukcji złożoności projektowej przy pomocy automatyzacji konwersji np. przy przechodzeniu na inną wersję oprogramowania. Z kolei droga emulacyjna może być trudniejsza jeśli mówimy o jej uniwersalnym wariantcie tj. wirtualizacji. Kolejne warianty migracyjne mogą też być bardzo kosztowne a nawet wymuszać skoki technologiczne (aplikacyjne). Mogą być również realizowane w wersji ograniczonej dla pojedynczych podsystemów o charakterze wyspowym (*stand alone*), niemniej i tu możemy mieć do czynienia z kombinacją wcześniej wymienionych trzech poziomów złożoności. W szczególności **projekt migracyjny (ERP) może zawierać elementy konstruowania nowego, złożonego systemu z jednoczesnym wprowadzaniem złożonych zmian do rozwiązań już istniejących i jednoczesnym transformowaniem znacznych obszarów istniejącego środowiska organizacyjno-technicznego przedsiębiorstwa.**

Zasygnalizowana cecha systemów ERP, jaką jest ich wysoka złożoność, wynika z roli jaką pełnią w przedsiębiorstwie, którą można porównać do roli podstawowego programu każdej konfiguracji komputerowej, jakim jest system operacyjny. Jednocześnie systemy te nadal stanowią centrum IT przedsiębiorstw mimo, że istnieją na rynku co najmniej od dwóch dekad (pakiety zintegrowane, MRP) i same stają się fragmentem jeszcze bardziej złożonych struktur (Tab. 1).

Tab. 1. Wzrost złożoności organizacyjnej i technicznej systemów gospodarczych.

Rodzaje złożoności organizacyjnej	Czas powstania (zgrubnie dekadowo)	Cechy gospodarczych systemów IT	Przykłady zastosowań (modele)
funkcyjna	lata 1960-te	wspomaganie pojedynczych funkcji	drukowanie statystyk sprzedaży
grup funkcji	lata 1970-te	wspomaganie wydziałów	zarządzanie zapasami w gospodarce magazynowej
procesowa	lata 1980-te	przejście do organizacji procesowej	całościowe pakiety ERP
wielosystemowa	lata 1990-te	integracja wielu przedsiębiorstw	standardy EDI, <i>e-commerce</i> , CRM, <i>customer replenishment</i>
sieciowa	lata zerowe XXI w.	globalna ekonomia sieciowa	rozproszone usługi webowe, ontologiczne bazy danych, SOA, <i>cloud computing</i>

2. Minimalizacja celów migracji jako warunek jej powodzenia

Z faktu, że informatyka funkcjonuje w sferze gospodarczej od dziesięcioleci wynika, iż coraz rzadsze są sytuacje instalowania komputerów i oprogramowania dla procesów biznesowych, które wcześniej obsługiwano „ręcznie“, bez ich elektronicznego wspomagania. Coraz częściej projekt informatyczny oznacza wymianę starszego sprzętu na nowszy i zastępowanie już istniejącego oprogramowania nowocześniejszym pakietem. Oznacza to, że strategie migracyjne nabierają coraz większej wagi. Jednocześnie ***architekci IT nie mają już tak wielu komfortowych możliwości konstruowania systemów od podstaw, bez kłopotliwych obciążeń historycznych.*** Raczej muszą projektować wymianę starszej infrastruktury IT na nowszą i to w taki sposób aby nie zakłócać bieżącego funkcjonowania przedsiębiorstwa.

Tymczasem otwarcie dużego projektu informatycznego w firmie prowokuje jego uczestników z różnych działów do prób załatwienia, niejako przy okazji, wszelkich problemów, które wiążą się z komputerami, mimo że można było to zrobić wcześniej. Można to także zrobić później ale niekoniecznie w trakcie migracji, zwiększając w ten sposób związane z nią ryzyka.

Przykład praktyczny. Dział kontroli jakości ma swoją bazę danych surowców i dostawców, zasilaną interfejsami ERP. Pielęguje przy tym dłuższe, 60-znakowe, nazwy

wyrobów, bo materiałowy moduł starego ERP ma tylko 30 znaków w stosownym polu. W trakcie migracji do nowego ERP gdzie pole jest 40-znakowe, dla uniknięcia podwójnej pielęgnacji danych, w dziale powstaje pomysł kompromisowego rozwiązania kwestii nazw przez sprowadzenie ich do tej właśnie długości. Jednocześnie planuje się ich nową klasyfikację, dyskutowaną od kilku lat. Podobnie dzieje się z wielokrotnie przesuwaną kwestią ich synchronizacji (te same surowce kupowane od różnych dostawców). Inne działy również uaktywniają się organizacyjnie, wykazując inwencję projektową, która może zagrozić całości przedsięwzięcia. Zamiast sprawnie przeprowadzonej migracji ERP grozi nam tu, skazany na niepowodzenie, nadmiernie rozbudowany projekt gwałtownej zmiany niemal całego krajobrazu IT firmy. Wyjściem z sytuacji jest *jasne zdefiniowanie celu migracyjnego: gwarancja tylko już posiadanej funkcjonalności aplikacji. Dopiero w drugiej fazie stopniowe dochodzenie do pełnej efektywności nowego pakietu ERP.*

Konsekwentne zastosowanie takiej strategii wymaga zdecydowanego wsparcia zespołu projektowego przez kierownictwo firmy, świadomego zasad rządzących rozwojem organizacji:

1. *Zasada ustawicznej zmiany*: organizacja musi nieustannie się zmieniać, inaczej nieustannie będzie postępować spadek jej użyteczności.
2. *Zasada rosnącej złożoności*: zmiany organizacji powodują wzrost jej złożoności a konieczność jej redukcji wymaga dodatkowych zasobów.

Zasada pierwsza wynika z nieustannej presji zmienego rynku (wymagania klientów, konkurencja) na przedsiębiorstwo natomiast druga wiąże się z interpretacją zarządzania jako przetwarzania informacji. W sytuacji kiedy owo przetwarzanie jest w daleko idącym stopniu wspomagane komputerowo *informatyka odgrywa w przedsiębiorstwie rolę technologii dyfuzyjnej* tj., bezpośrednio lub pośrednio, przenikającej wszystkie jego obszary. Mamy wówczas do czynienia z wszechzwiązkami informatycznych modułów wspomagających organizację, połączonych „każdy z każdym“. Tymczasem w obszarze struktury softwarowej (organizacyjnej) przyjmuje się, że modyfikacja K modułów, spośród ich ogólnej liczby M, powoduje konieczność zweryfikowania

$$I = K(M-K) + K(K-1)/2$$

interfejsów tj. nie tylko wszelkich połączeń między zweryfikowanymi modułami (drugi człon równania) ale także interfejsów między „nowymi” a „starymi” modułami. Bez szczegółowego badania przebiegu zmienności tej funkcji widać szybki przyrost jej wartości w zależności od K, niezależnie od poziomu parametru M.

Tak sformułowane reguły są więc organizacyjnymi wersjami praw Lehmana [1], znanych w obszarze inżynierii systemów informacyjnych (*software engineering*) i korespondują z wcześniejszymi uwagami dotyczącymi fenomenu wzrostu złożoności organizacyjno-technicznej systemów gospodarczych. W szerszym wymiarze owo zjawisko wynika także z praw termodynamiki i nieuchronnych zmian *entropii*. Przy czym znaczenie samej funkcji stanu wykracza poza przyrodę nieożywioną i dotyczy także funkcjonowania człowieka oraz tworzonych przez niego systemów technicznych (entropiczna definicja informacji [2]).

3. Przykłady bazodanowych strategii migracyjnych

Pokazany w p. 2 przykład ilustruje wagę danych referencyjnych w przedsiębiorstwie. Ich wiarygodność odgrywa kluczową rolę dla normalnego funkcjonowania firmy. W projektach migracyjnych dane te podlegają dodatkowym narażeniom, co zilustrujemy dalej fragmentem migracji ERP dotyczącym bazy danych materiałowych.

Przykład: Prawidłowe znakowanie wyrobów (opakowania pierwotne, wtórne i kolejnych stopni) stanowi podstawę prawidłowego funkcjonowania automatyki produkcyjnej (np. księgowanie ilości wyprodukowanych wyrobów gotowych) i magazynowej (wejście/wyjście palet). W migrowanym systemie zapamiętano w specjalnym polu bazy danych materiałowych warianty produktu dla elementu aplikacyjnego (AI, Application Identifier) nr 20 standardu EAN. System docelowy wadliwie przejął wartość do innego pola generując wariant podstawowy (00) dla wszystkich palet dostarczanych klientom, którzy naliczali kary umowne od każdej błędnie oznakowanej palety. Poważniejsze skutki wywołałyby błędy migracji kodów EAN wyrobów, list alergenów (przemysł spożywczy), składników (przemysł farmaceutyczny, chemiczny, kosmetyczny) czy relatywnych dat przydatności do użytku (skojarzone z materiałem okresy liczone od produkcji względnie od zakończenia dystrybucji).

Fragment migracji ERP dotyczący bazy danych materiałowych stanowi poważne wyzwanie dla zespołu projektowego, zwłaszcza jeśli dodatkowo łączy się z fuzją przedsiębiorstw (projekt migracyjno-integracyjny). Pytamy zatem o mapę drogową (*road map*) migracji, dotyczącej sposobów przejścia do nowych rozwiązań w wymiarze informatycznym, gdzie mamy szereg alternatyw pokazanych w tabeli 2.

Tab. 2 Migracja bazy danych ERP na przykładzie danych materiałowych.

Strategie migracyjne							
procesowe (techniczne)			projektowe (organizacyjne)				
<i>update</i>	<i>upgrade</i>	konsolidacja	klastry	cele ustalone		cele zmienne	
<i>off-line/on-line</i>		modele (wirtualizacja)		dane referencyjne (stałe)	dane transakcyjne (zmiennie)	podsystemy	sposoby kodowania

Działania migracyjne można podzielić na procesowe (w tym przypadku o charakterze technicznym, bardziej standaryzowane) i projektowe (ukierunkowane organizacyjnie, bardziej unikatowe). Możemy powiedzieć, że te pierwsze zorientowane są na cele bieżące a z uwagi na większą rutynowość i stabilniejsze relacje międzyludzkie, wiążą się z mniejszymi ryzykami i słabszym zaangażowaniem kierownictwa. W trybie projektowym (konsultingowym) mamy z kolei do czynienia ze strategicznymi celami a zatem większym ryzykiem i częstszymi konfliktami, co wymaga zwiększonej uwagi kierownictwa.

Podane przykłady procesowe można odnieść np. do sfery bazodanowej:

- uaktualnienie wersji oprogramowania (*update*),
- nowa wersja oprogramowania (*upgrade*),
- konsolidacja (migracja bez zmian wersji oprogramowania),
- klastry (wymagają specjalnych modeli migracyjno/konsolidacyjnych).

Odrębne zagadnienia techniczne to możliwości migrowania w trybie *on-line* (podczas pracy aplikacji) i *off-line* (z jej zatrzymaniem), budowanie środowisk testowych (dobre środowisko testowe to jedno z większych wyzwań stających przed zespołem migracyjnym) i strategie wirtualizacyjne: od typowo technicznych, jak partycjonowanie czy emulacje aż do zorientowanych biznesowo, np. komodytyzacja (*commodity*, infrastruktura standaryzowana dostępowo/rozliczeniowo) [3].

W sferze działań w typowym trybie projektowym można zauważyć te o niższym lub wyższym stopniu niepewności. Podział jest relatywny i w każdym przypadku musimy liczyć się ze znacznym ryzykiem projektowym. Może się też zdarzyć, że bardziej standaryzowane migracje danych okażą się trudniejsze niż integrowanie podsystemów satelitarnych skojarzonych z ERP (może być ich wiele i bardzo zróżnicowanych, jeśli chodzi o cechy rzutujące na powodzenie migracji).

Zdefiniowane cele strategii migracyjnej wymagają z kolei przełożenia ich na metody (np. zwine, *agile*), modele (np. COBIT, ISO, PRINCE2) i narzędzia (np. kwestionariusz Thomsetta, lista zagrożeń McConella) wdrażania ładu korporacyjnego (IT Governance) [4]. Jeśli wskażemy przy tym dodatkowo na trendy rozwojowe webowych platform programowych to nieuchronnie nasuwają się tu skojarzenia z modelem SOA (Service Oriented Architecture) [5]. Model ten łączy w sobie dwa, pozornie sprzeczne, elementy organizacyjne: centralizację i decentralizację czyli architekturę integrującą rozproszone usługi. ***Charakterystyczne dla SOA jest zdecydowane wyjście poza czysto narzędziowy wymiar inżynierii softwarowej i uwzględnienie pozatechnicznych aspektów gospodarczych aplikacji IT.*** W szczególności zakłada się, że ze specyfiki modelu biznesowego można wyprowadzić architekturę wspomagającego go systemu softwarowego.

Odpowiedź na pytanie jak kojarzyć platformy SOA z przedsięwzięciami migracyjnymi mogłaby być podobnie rozbudowana jak w przypadku wyboru samego systemu ERP. Pokażmy zatem punktowo przykład strategii SOA według kategorii Gartner Group J2EZ czyli Easy SOA w oparciu o platformę J2EE (Java Enterprise). Jak nazwa wskazuje, jest to podejście, które obiecuje szybką implementację, także firmom, które dotąd przykładały mniejszą wagę do systemowego rozwoju IT. Można je realizować używając narzędzi takich firm jak ClearNova lub Unify. W tym ostatnim przypadku mamy do czynienia z pakietem NXJ Developer, który spełnia wymagania X-RAD (Extreme Rapid Application Development) i może być wykorzystany w fazach:

- projektowania,
- migracji,
- implementacji.

W pierwszej z nich modelujemy procesy biznesowe generując diagramy z wykorzystaniem modułu graficznej wizualizacji BPM (Business Process Management). Z kolei tworzymy zintegrowany system usług webowych na poziomie bazodanowym – NXJ generuje usługi (Web Service) odpowiedzialne za operacje SQL-owe dla wyspecyfikowanych tabeli. Również i tu nie ma potrzeby klasycznego programowania „liniowego”; aplikacja jest generowana graficznie, podobnie jak w szeroko znanych od kilkunastu lat pakietach 4GL. W fazie implementacji, także przy użyciu NXJ, generujemy maski dla użytkowników. Również i ten przykład potwierdza, że ***kluczowym elementem metodyki SOA jest prawidłowe modelowanie środowiska biznesowego a więc semantyczna integracja sfery technicznej i użytkowej.***

4. Problemy migracyjne i ich przyczyny

Ostatnia uwaga koresponduje z pozytywnymi efektami stosowania podejścia referencyjnego SOA/BPM w projektach migracyjnych:

- zmniejszenie złożoności problemów projektowych dzięki podejściu systemowemu;
- przejrzystość procesów w całości i w rozbiciu na fragmenty żądanej wielkości;

- jasne zdefiniowanie celów i zakresów odpowiedzialności;
- spełnienie wymagań prawnych i rewizyjnych;
- małowredundantna dokumentacja procesowa, dostępna także dla partnerów;
- ustalenie miar pozwalających na optymalizacje procesów;
- jednolita terminologia dla wykonawców, użytkowników, klientów i dostawców.

Rozważmy bliżej ostatni z wymienionych czynników: jednolite nazewnictwo. Konwencje terminologiczne są doskonale znane każdemu programiście wraz z konsekwencjami ich niestosowania. W trywialnym przypadku prowadzi to do niekompilowalności programu a na poziomie wykonawczym do jego nieczytelności, nawet dla samego autora. **Jednolite nazewnictwo wymuszane jest tu głównie przez słowniki i gramatyki sztucznych języków a ponadto w oparciu o normatywy zespołowe.** W obszarze dużych projektów migracyjnych, na styku IT/użytkownicy, problem jest poważniejszy a jego ignorowanie obniża efektywność komunikowania się w grupie, rzutując negatywnie na całość przedsięwzięcia.

Trudno jest precyzyjnie modelować, optymalizować i mierzyć coś, czego nie można precyzyjnie opisać. Przykładowo: jeśli w przedsiębiorstwie używamy słowa „partia“ i w związku z tym „śledzenie partii“ (*tracing*) to można pod tym pojęciem rozumieć zarówno partię produkcyjną jak i dostawczą. Z kolei w samym obszarze wytwórczym ktoś może żargonowo posługiwać się oryginalnie angielskim terminem ERP „shop order“ czyli „zleceniem produkcyjnym“, z którym wiąże się wyprodukowanie partii towaru ale związki między tymi pojęciami nie zawsze są jednoznaczne i muszą zostać sprecyzowane w dokumentacji procesowej. W szczególności jedno zlecenie produkcyjne może dotyczyć wielu partii produkcyjnych, jeśli to ostatnie pojęcie używane jest w znaczeniu technologicznym, związanym z parametrami linii wytwórczej (wsad). I dalej: wiele „partii produkcyjnych“ może otrzymać ten sam symbol partii (Lot Code) dla wielu palet wyrobu gotowego. Owszem, standardy EAN nakazują jednoznaczne nadawanie środkom transportowym numeru SSCC (Serial Shipping Container Code) ale trudniej znaleźć ich powiązania z „partiami surowców“ użytych do produkcji. Widzimy zatem, że precyzyjne opisanie wieloaspektowości jednego tylko zagadnienia procesowego jakim jest śledzenie partii wymaga precyzji terminologicznej.

Możemy wprawdzie założyć, że kontekst dokumentacji wystarczy aby w każdej sytuacji zidentyfikować znaczenie terminu, można także poprosić określone osoby o komentarz, ale to wiąże się z dodatkowym czasem, z drugiej strony chcemy uzyskać wspomniany efekt zmniejszenia nadmiarowości opisów procesów. Warto więc zdefiniować określone konwencje terminologiczne, pozwalające na oszczędność słów bez strat precyzji przekazu. Jedną z nich jest stosowanie warstwowości w modelowaniu procesu biznesowego. Przykładowo: na najwyższym poziomie modelowania używamy liczby mnogiej używając konstrukcji typu „procesy produkcji“; na kolejnym, niższym poziomie, stosujemy liczbę pojedynczą w zwrotach postaci „proces pakowania“. Wreszcie na najniższej warstwie modelowania (tutaj w modelu trójpoziomowym) posługujemy się złoženiami czasownik+ rzeczownik, np. „konfekcjonowanie wyrobów“, przy czym w idealnym wariacie występujące tu pojęcia pochodzą z ich zdefiniowanego katalogu.

Jak dotąd nie definiowaliśmy wprost przedmiotowego pojęcia „wiarygodności (baz) danych“ i związanych z nim rozbudowanych list atrybutów jakościowych. Wskazano jednak na znaczenie stosowania modeli referencyjnych ładu korporacyjnego (IT Governance) w projektach migracyjnych, w szczególności dla danych referencyjnych. Można zatem skrótkowo stwierdzić, że migrowane bazodanowo dane są wiarygodne jeśli

spełniają wymagania systemów MDM (Master Data Management) [6]. Pozornie nie mówimy tu o standardach normowanych ale *de facto* w praktyce przemysłowej. Faktycznie, na rynku istnieje szereg rozwiązań w sferze systemów zarządzania danymi referencyjnym. Niemniej w aspekcie bazodanowym można sprowadzić je do systemów FDM (Federated Database Management) gdzie z kolei mamy do czynienia z gremiami normującymi jakość danych.

Sfederowana (szerzej: wirtualna) baza danych oznacza bowiem jednolity system bazodanowy dla jego składowych, w szczególności zarządzanie metadanymi, co ma kluczowe znaczenie dla sukcesu projektu migracyjnego. Z kolei celem migracji (z fuzją firm lub bez) ERP jest właśnie uzyskanie jednolitego rozwiązania aplikacyjnego na drodze transformacji systemu migrowanego do docelowego, przy czym (jak pokazano w p. 1) niekoniciecznie oznacza to zastąpienie jednego systemu drugim. Więcej: nowy system może integrować szereg wcześniej istniejących podsystemów, formalnie działających poza klasycznie rozumianym ERP. Tak postrzegana całość migracyjnego środowiska IT przedsiębiorstwa może być również interpretowana jako rodzaj sfederowanej bazy danych.

5. Pozatechniczne aspekty migracji

Według badań Gartner Group trzy czwarte wszelkich fuzji kończy się niepowodzeniem a istotną tego przyczyną są problemy migracji systemów informatycznych. Mamy tu bowiem do czynienia z projektami o charakterze unikalnym. Operując terminologią informatyczną można stwierdzić, że każdorazowo konstruowana jest przy tym wersja złożonego oprogramowania o numerze 1.0.

Z kolei *migrowanie systemów ERP, właśnie z uwagi na ich kompleksowość, wymaga podejścia interdyscyplinarnego*, co nadal bywa problemem dla kierowników takich projektów. Często kładą oni przesadny nacisk na czysto techniczne aspekty przedsięwzięcia, natomiast przy ich nieznanym nierealistycznie szacują nakłady inwestycyjne (błędy planowania zasobów).

Płynność celów i brak konsekwencji w ich realizacji to kolejny problem z tego obszaru. Jeśli zakładamy, że migrowane oprogramowanie zmniejszy zatrudnienie to efekt ekonomiczny zostanie tylko wtedy osiągnięty gdy faktycznie dokonamy redukcji personelu. Tymczasem zmiana standardów informatycznych wymusza często nowe formy organizacji zadań rzutując również na sferę osobową przedsiębiorstwa (hierarchie, podział pracy). Rozwój nowego systemu może przybierać także formy niekontrolowane: nadmiar informacji potrafi być tak samo szkodliwy jak ich brak a większa ilość danych często prowadzi do trudności w realizacji rutynowych zadań, miast je upraszczać.

Koresponduje to z tezą, że *do istotnych przyczyn niepowodzeń projektów migracyjnych należą pozatechniczne*. Potrzeba gwarancji wiarygodności baz danych, co w zakresie informacji referencyjnych wiąże się z konsekwentnym wdrażaniem rozwiązań poziomu MDM, może prowadzić do utraty suwerenności organizacyjnej przez niektórych pracowników. W przykładzie z p. 2 oznacza to zależność nazw produktów w systemie laboratoryjnym od nazw pielęgnowanych centralnie w systemie ERP, co jest oczywistą zaletą techniczną i organizacyjną ale może rodzić trudności akceptacji nowego systemu przez pracowników.

Kolejną kwestią pozatechniczną jest pytanie o metodę ostatecznego uruchomienia migrowanego systemu. Można to robić stopniowo (modułowo) bądź zdecydować się na taktykę "big-bangu". Z przyczyn, polityczno-asekuracyjnych krytykuje się często ten ostatni wariant, zalecając koncepcję wdrożenia stopniowego, jako technicznie

bezpieczniejszego (o ile to możliwe). Jest to często wygodniejsze dla informatyków jak i towarzyszących przedsięwzięciu firm doradczych (dłuższy czas ich obecności w projekcie oznacza z reguły większe dochody). Z drugiej strony praktyka projektowa pokazuje, że *czas migracji (fuzji) nie powinien być dłuższy niż kilkanaście miesięcy, co wiąże się z unikaniem podwójnego obciążania pracowników koniecznością równoległej pielęgnacji danych referencyjnych w dwóch systemach.*

Literatura

1. Lehman M.M.: Metrics and laws of software evolution—the nineties view. Proceedings of the 4th International Software Metrics Symposium (METRICS '97), IEEE, Albuquerque, NM, USA, November 5-7, 1997 , 20-32.
2. Muller S.J.: Asymmetry: The Foundation of Information. Springer, New York, 2007.
3. Smith J., Nair R.: Virtual Machines: Versatile Platforms for Systems and Processes. Elsevier Inc., San Francisco CA USA, 2005.
4. Brand K.: IT Governance based on Cobit 4.1 - A Management Guide. Van Haren Publishing, Amersfoort, NL, 2007.
5. Berkem B.: From The Business Motivation Model (BMM) To Service Oriented Architecture (SOA). Journal of Object Technology Vol. 7 Nr. 8, ETH Zurich, 2008, 57-70.
6. Loshin D.: Master Data Management. Morgan Kaufmann Publishers, Burlington MA USA, 2009.

Dr inż. Jarosław BADUREK
Katedra Inżynierii Systemów Produkcji
Wydział Zarządzania i Ekonomii
Politechnika Gdańska
ul. G. Narutowicza 11/12
80-952 Gdańsk
email: j_badurek@yahoo.com