

**WYZWANIA W ROZWOJU
PODSTAW METODYCZNYCH
PROJEKTOWANIA SYSTEMÓW
INFORMATYCZNYCH
ZARZĄDZANIA**

Studia Ekonomiczne

ZESZYTY NAUKOWE

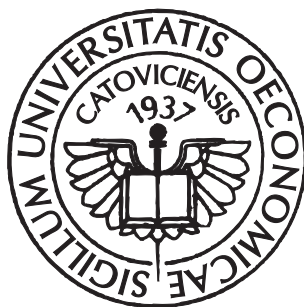
WYDZIAŁOWE

UNIWERSYTETU EKONOMICZNEGO

W KATOWICACH

**WYZWANIA W ROZWOJU
PODSTAW METODYCZNYCH
PROJEKTOWANIA SYSTEMÓW
INFORMATYCZNYCH
ZARZĄDZANIA**

**Redaktorzy naukowi
Małgorzata Pańkowska
Stanisław Stanek**



Katowice 2013

Komitet Redakcyjny

Krystyna Lisiecka (przewodnicząca), Anna Lebda-Wyborna (sekretarz),
Halina Henzel, Anna Kostur, Maria Michałowska, Grażyna Musiał, Irena Pyka,
Stanisław Stanek, Stanisław Swadźba, Janusz Wywiół, Teresa Żabińska

Komitet Redakcyjny Wydziału Informatyki i Komunikacji

Tadeusz Trzaskalik (redaktor naczelny), Mariusz Żytniewski (sekretarz),
Grażyna Trzpiot, Małgorzata Pańkowska, Andrzej Bajdak

Rada Programowa

Lorenzo Fattorini, Mario Glowik, Gwo-Hsiung Tzeng,
Zdeněk Mikoláš, Marian Noga, Bronisław Micherda, Miloš Král

Recenzenci

Stanisław Wrycza

Redaktor

Beata Kwiecień

Skład tekstu

Urszula Grendys

© Copyright by Wydawnictwo Uniwersytetu Ekonomicznego
w Katowicach 2013

ISBN 978-83-7875-038-3

ISSN 2083-8611

Wszelkie prawa zastrzeżone. Każda reprodukcja lub adaptacja całości
bądź części niniejszej publikacji, niezależnie od zastosowanej
techniki reprodukcji, wymaga pisemnej zgody Wydawcy

WYDAWNICTWO UNIwersytetu Ekonomicznego w Katowicach

ul. 1 Maja 50, 40-287 Katowice, tel.: +48 32 257-76-35, faks: +48 32 257-76-43
www.wydawnictwo.ue.katowice.pl e-mail: wydawnictwo@ue.katowice.pl

SPIS TREŚCI

WSTĘP	7
Krystian Gembala	
PROJEKTOWANIE ORGANIZACJI	9
Summary	22
Sebastian Kostrubała	
KOMUNIKACJA CZŁOWIEK-KOMPUTER	23
Summary	40
Radosław Kowal	
ZARZĄDZANIE CYKLEM ŻYCIA APLIKACJI – NA PRZYKŁADZIE SAP SOLUTION MANAGER	41
Summary	55
Artur Machura	
ANALIZA BIZNESOWA W PRZEDSIĘWZIĘCIACH IT	57
Summary	62
Joanna Palonka	
ZARZĄDZANIE UMOWAMI Z WYKONAWCAMI SYSTEMÓW IT	63
Summary	80
Małgorzata Pańkowska	
CELE ROZWOJU ARCHITEKTURY PRZEDSIĘBIORSTWA	81
Summary	89
Małgorzata Pańkowska	
JĘZYK SYSML W INŻYNIERII WYMAGAŃ	91
Summary	102
Teresa Porębska-Miąc	
PROJEKTOWANIE I WDRAŻANIE SYSTEMÓW CRM	103
Summary	113

Anna Sołtysik-Piorunkiewicz	
PROJEKTOWANIE LOGISTYCZNYCH SYSTEMÓW INFORMATYCZNYCH.....	115
Summary	135
Stanisław Stanek	
ANALIZA WYBRANYCH KONCEPCJI W OBSZARZE PROJEKTOWANIA	
WYMAGAŃ	137
Summary	162
Stanisław Stanek, Edyta Abramek, Radosław Kowal	
PROJEKTOWANIE SYSTEMÓW INTERORGANIZACYJNYCH.....	163
Summary	176
Rafał Tyrała	
ELEKTRONICZNA WYMIANA DANYCH NA PRZYKŁADZIE	
SYSTEMU BPSC IMPULS 5.....	177
Summary	193
Mariusz Żytniewski, Piotr Zadora	
MODELOWANIE PROCESÓW BIZNESOWYCH Z UŻYCIEM NOTACJI BPMN.....	195
Summary	210

WSTĘP

Pomimo ogromnego postępu w obszarze doskonalenia technologii informacji, wciąż trwają poszukiwania racjonalnych rozwiązań metodologicznych tworzących fundament doskonalszych, niż wykorzystywane obecnie, systemów informatycznych wspierających zarządzanie organizacjami.

Studia stanowią monograficzny przegląd wybranych prac nad rozwojem metod projektowania i ich zastosowań oraz omawiają zasadnicze kierunki dalszego ich rozwoju. Obejmują 13 artykułów, prezentujących kolejno najważniejsze zagadnienia tworzenia i zastosowań systemów informatycznych zarządzania, skupiając głównie uwagę na dorobku polskich ośrodków naukowych.

Publikacja ta jest pierwszą pracą naukową na Uniwersytecie Ekonomicznym w Katowicach poświęconą analizie i projektowaniu systemów informatycznych zarządzania. Zagadnienia przedstawione w artykułach znacznie wybiegają poza zakres przedstawiony na wykładach z przedmiotu Analiza i projektowanie systemów informatycznych.

Autorzy artykułów podjęli się zadania wskazania kierunków badań i przedstawienia ich rezultatów w dziedzinie analizy i projektowania systemów informatycznych zarządzania. Poza tym, podjęli się prezentacji tematów związanych z projektowaniem systemów i uzupełniających w sposób istotny tę dziedzinę wiedzy.

Studia te powstały z myślą o praktykach rozwoju systemów informatycznych – menedżerach, analitykach, projektantach, wdrożeniowcach, użytkownikach aplikacji programowych, a także nauczycielach akademickich i studentach, którzy są zainteresowani problematyką projektowania, wdrażania i doskonalenia systemów informatycznych biznesu.

Celem nadrzędnym prezentowanej pracy jest ustalenie, na co należy zwrócić uwagę, przygotowując projekt systemu informatycznego w zróżnicowanym według kompetencji środowisku, jakim jest przedsiębiorstwo informatyczne, by zderzenie odmiennych poglądów oraz różnorodnej wiedzy, różnych perspektyw postrzegania, schematów, a także narzędzi projektowania i wnioskowania nie doprowadziło do istotnych zaburzeń w procesie budowy systemów informatycznych zarządzania.

Autorzy mają nadzieję, że opracowanie będzie stanowić użyteczne źródło wiedzy oraz inspiracji nie tylko dla studentów i pracowników naukowych, ale także przedstawicieli praktyki gospodarczej zainteresowanych wzbogacaniem wiedzy na temat rozwoju i projektowania systemów informatycznych zarządzania.

Autorzy składają podziękowania recenzentowi prof. dr hab. Stanisławowi Wryczy za przygotowanie wnikliwej i inspirującej do dalszych prac badawczych recenzji.

Małgorzata Pańkowska

Stanisław Stanek

Krystian Gembala

PROJEKTOWANIE ORGANIZACJI

1. Cechy organizacji

Organizacja (przedsiębiorstwo) jest jednostką wyodrębnioną z otoczenia pod względem ekonomicznym, prawnym, organizacyjno-technicznym i terytorialnym [GlHe06].

Przyjmuje się, że głównym celem organizacji jest osiągnięcie zysku ze swojej działalności produkcyjnej, usługowej lub handlowej. Przedsiębiorstwo dysponuje ściśle określonymi zasobami materialnymi i niematerialnymi.

Poprzez wyodrębnienie ekonomiczne rozumie się sytuację, gdy:

- organizacja posiada odrębność majątkową,
- gospodarowanie jest związane z ryzykiem ekonomicznym,
- pojawia się określona relacja między dochodami pracowników przedsiębiorstwa a jego wynikami produkcyjnymi i ekonomicznymi, co jest powodem odczucia i motywacji wspólnego interesu materialnego,
- rozwój przedsiębiorstwa, polegający na powiększeniu swojego majątku, jest w dużym stopniu uzależniony od możliwości pokrycia nakładów inwestycyjnych z własnych dochodów,
- organizacja reprodukuje, we własnym zakresie, zużywane zasoby majątkowe poprzez zbywanie swych świadczeń, lecz nie łamiąc przy tym zasad rentowności [Mach 98].

Wyodrębnienie prawne oznacza zdolność organizacji do działań prawnych. Dzięki temu otrzymuje ona „osobowość prawną”, która pozwala jej na zawieranie umów, zaciąganie kredytów bankowych, występowanie jako odrębny podmiot w obrocie towarowym oraz samodzielne funkcjonowanie na rynku. Organizacja uzyskuje osobowość prawną poprzez wpis do rejestru przedsiębiorstw. Wszystkie wątpliwości w tej kwestii regulują odpowiednie przepisy. Pojęcie rejestracji przedsiębiorstwa dotyczy tylko tych podmiotów, których działalność jest wykonywana w formie przedsiębiorstwa opisanego w odpowiednim akcie prawnym.

Wyodrębnienie organizacyjno-techniczne polega na tym, że dana jednostka tworzy wydzieloną całość pod względem terytorialnym i organizacyjnym. Posiada własną strukturę organizacyjną, zobrazowaną za pomocą graficznego

schematu organizacyjnego. Stanowi on układ relacji zachodzących pomiędzy elementami organizacji, do których zalicza się szczebel strategiczny, taktyczny, operacyjny, pomocniczy oraz technostrukturę.

Wyodrębnienie terytorialne określa swobodę działania na rynku bez ingerencji z zewnątrz, a szczególnie bez wpływu organów państwowych. Oznacza również wyłączne prawo do decydowania w wyborze celów organizacji, technologiach i sposobach ich realizacji. Czynnikiem ten jest także odpowiedzialny za błędnie podejmowane decyzje.

1.1. Funkcje i cele organizacji

Organizacje są trwałym elementem kultur i cywilizacji. Pełnią różnorodne funkcje, m.in.:

- przewyżniają granice naszych możliwości jako jednostek,
- umożliwiają realizację celów, których osiągnięcie bez nich byłoby trudniejsze lub nawet niemożliwe,
- gromadzą i chronią przeważającą część naszej wiedzy, skumulowanej i zarejestrowanej w formie aktualnych technik cywilizacyjnych,
- umożliwiają opracowywanie nowych i skuteczniejszych sposobów działania,
- zapewniają swoim pracownikom źródło utrzymania, a także stają się miejscem samorealizacji i osobistego zadowolenia [Grif02].

Jednym z zasadniczych elementów organizacji jest cel. Trzeba bardzo dokładnie określić: co ma być zrobione?, gdzie? i kiedy? Istnienie organizacji bez celu nie ma żadnego rzeczowo uzasadnionego sensu [GIHe06]. W organizacjach można wyróżnić kilka rodzajów celów:

- **strategiczne** – związane z najwyższym szczeblem organizacji, którego rolą jest znajdowanie, tworzenie i opracowywanie kierunków rozwoju;
- **taktyczne** – dotyczą najczęściej tego wszystkiego, co przyczynia się do zrealizowania celów strategicznych. Ustalają je głównie menedżerowie średniego szczebla organizacji;
- **operacyjne** – ustalone przez kierowników najniższego szczebla organizacji i są tylko z tym szczeblem związane.

Cele przedsiębiorstwa określa się hierarchicznie. Najpierw są opracowywane cele globalne, począwszy od najbardziej ogólnych, a następnie cele szczegółowe dla każdego szczebla organizacji, kończąc na najniższym jej poziomie – operacyjnym. Cele powinny jasno określać czasy realizacji, ponadto ważne jest to, aby ich ilość była dla każdego szczebla jednoznacznie wyznaczona [Mach99].

Dla przykładu celami dla szczebla:

- strategicznego mogą być: wysokie zyski, zdobycie znaczącego rynku, wejście na zagraniczny rynek;
- taktycznego: odnowa parku maszynowego, wdrożenie nowych technologii;
- operacyjnego natomiast: poprawa jakości wyrobu, skrócenie czasu wytwarzania produktu, usprawnienie techniki wykonania itd.

1.2. Plany organizacji

Do realizacji założonych celów niezbędny jest plan. Żadna organizacja nie będzie skuteczna bez wcześniejszego opracowania planu. Pozwala on na kontrolowanie wytyczonych zamierzeń, wskazanie postępów w realizacji osiągnięcia celu oraz odnalezienie odstępstw od wytyczonego celu [GIHe06].

Wyróżnia się dwa zasadnicze rodzaje planów:

- strategiczny – wieloletni, perspektywiczny, opracowywany na 5 i więcej lat,
- operacyjny – bieżący, krótkookresowy opracowywany na okres do 1 roku, stanowiący część realizacji wdrożeń planu strategicznego.

Prawidłowo skonstruowany plan powinien zawierać:

- wyszczególnione zasoby,
- wykaz zastosowanych procedur wraz z ich opisem,
- zestaw działań,
- spis zadań z opisem szczegółowym,
- skład grupy ludzi odpowiedzialnych za wykonanie i kontrolę realizacji celu,
- szczegółowe i skonkretyzowane wyszczególnienie celów, dla których plan został opracowany,
- normy czasowe (okresy realizacji, terminy),
- normy zużycia zasobów (pracochłonność, materiałochłonność),
- normy kosztów,
- normy ryzyka,
- rozkłady prac,
- miejsca realizacji celów,
- miejsca, punkty, czasy, w których będzie sprawdzany przebieg postępu wykonywania planu,
- sposoby dokonywania oceny.

Organizację tworzy się po to, aby poprzez wspólne działanie można było osiągnąć znacznie więcej niż w pojedynkę.

Dla każdej organizacji ustala się cele i opracowuje plany ich realizacji. Aby było to możliwe, należy organizację odpowiednio zaprojektować, a następnie stworzyć.

Dostosowywanie struktury do jej celów i zasobów nazywamy projektowaniem organizacji.

Przystępując do projektowania organizacji, należy rozważyć jakie elementy i czynniki uwzględnić oraz jakimi zasadami się kierować, aby poprawnie opracować strukturę organizacji.

1.3. Czynniki projektowe organizacji

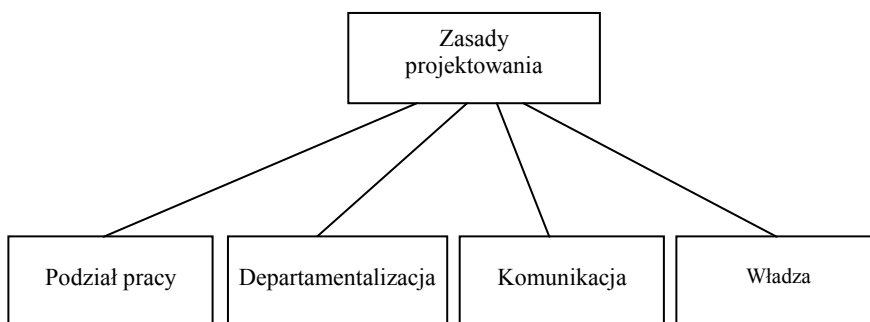
Do podstawowych czynników projektowych organizacji zalicza się:

- zasoby ludzkie,
- zadania,
- hierarchię władzy.

Czynniki te mają wielki wpływ na skuteczność zarządzania przedsiębiorstwem.

Projektowanie struktur organizacyjnych oraz sam proces organizowania można scharakteryzować za pomocą czterech kluczowych zasad przedstawionych na rys. 1. Są nimi:

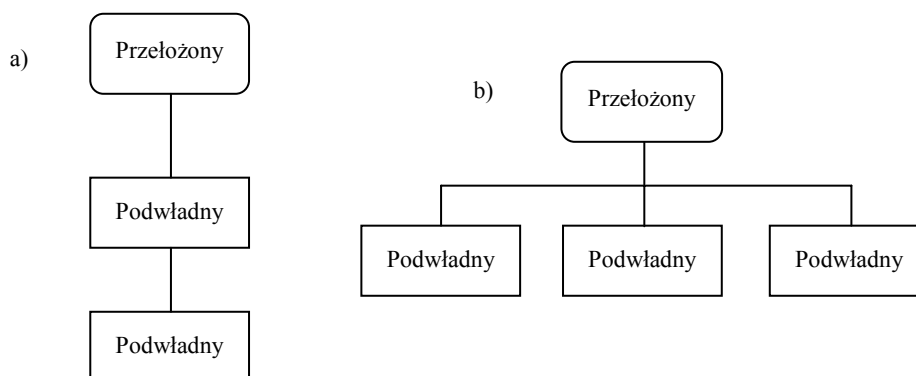
1. Podział pracy w organizacji musi być dokonany w taki sposób, aby określone zadania wykonywały osoby jak najbardziej kompetentne, co pozwoli na podniesienie efektywności organizacji.
2. Zadaniem menedżera jest pogrupowanie podobnych do siebie zadań w większe jednostki, odpowiadające różnym rodzajom struktur organizacyjnych (departamentalizacja).
3. Komunikacja w organizacji jest hierarchiczna – oznacza to, że wszelkie uprawnienia są delegowane z góry do dołu, uwzględniając więc służbową.
4. Dozwolona jest centralizacja lub decentralizacja kontroli i władzy [GIHe06].



Rys. 1. Kluczowe zasady projektowania struktur organizacyjnych

Struktura organizacyjna – to graficzne przedstawienie relacji między poszczególnymi elementami (stanowiska organizacyjne, komórki, jednostki) organizacji. Istotą struktury według Stabryły [Stab98] jest odpowiednie spojenie celów i zadań z ludźmi oraz sposobami oddziaływania na nich w procesach pracy. Inaczej mówiąc, to utworzenie w organizacji takich elementów, a następnie powiązań, aby efektywniej nią kierować (zarządzać).

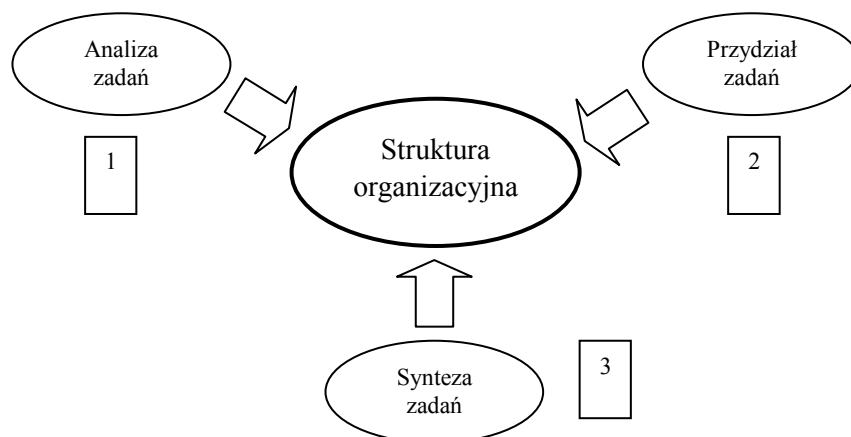
Podział pracy – w przedsiębiorstwie jest podstawą struktur organizacyjnych. Polega on na tym, iż ogół zadań dzieli się na odpowiednie części, czyli zadania cząstkowe, a te z kolei na dalsze elementy, doprowadzając do czynności elementarnych, dalej już niepodzielnych. W praktyce i w teorii zarządzania stosuje się dwa rodzaje podziału zadań: pionowy i poziomy, co przedstawiono na rys. 2.



Rys. 2. Rodzaje podziału zadań: a) pionowy, b) poziomy (rodzajowy, przedmiotowy, terytorialny)

Głębokość tych podziałów zależy od wielu czynników, ale przede wszystkim od rozpiętości kierowania.

Struktura organizacyjna przedsiębiorstwa powstaje w wyniku trzech działań przedstawionych na rys. 3:



Rys. 3. Zadania wpływające na powstanie struktury organizacji

Strukturę organizacyjną można rozpatrywać z punktu widzenia następujących czynników:

- specjalizacji czynności (dotyczy ona podziału pracy na poszczególne zadania i łączenia tych zadań w jednostki robocze),
- normalizacji czynności (odnosi się ona do procedur stosowanych przez organizację w celu zwiększenia możliwości przewidywania jej funkcjonowania oraz doprowadzenia do jednorodności i zwartości pracy),
- koordynacji czynności (dotyczy procedur, które integrują funkcje działów w danej organizacji),
- centralizacji i decentralizacji podejmowania decyzji (czynnik dotyczący lokalizacji uprawnień decyzyjnych),
- wielkości jednostki roboczej (liczba pracowników w grupie roboczej [GrHe04]).

Nie ma jednej stałej, najlepszej struktury dla danej organizacji. Będzie się ona zmieniać w czasie w miarę rozwoju organizacji.

Do najważniejszych funkcji struktur organizacyjnych zalicza się przede wszystkim to, że:

- struktury organizacyjne stanowią ramy działań organizacyjnych,
- regulują działania poszczególnych pracowników i zespołów,
- umożliwiają osiągnięcie określonego poziomu realizacji potrzeb pracowników,
- zapewniają efektywną realizację celów organizacji,
- kształtują zależności hierarchiczne i funkcjonalne,
- dzielą uprawnienia decyzyjne i odpowiedzialność [GIHe06].

Projektując organizację, ma się przede wszystkim na uwadze określenie elementów struktury oraz łączących je więzi. Wszystkim częściom organizacji przydziela się odpowiednie cele i zadania. Rozmieszcza się uprawnienia decyzyjne i odpowiedzialność, a następnie formalizuje procesy wykonawcze.

Do projektowania struktur organizacyjnych można wybrać jeden ze sposobów:

- diagnostyczny,
- prognostyczny.

Podejście diagnostyczne polega na rejestracji faktycznego sposobu organizowania, jego analizie i ocenie. Wykorzystywane jest też do wyszukiwania niewłaściwych rozwiązań, występujących w organizacji, a także znajdowania elementów, których modyfikacja wpłynęłaby na poprawę działania.

Podejście prognostyczne polega na poszukiwaniu rozwiązania idealnego, które w danych warunkach pozwoliłoby jak najsprawniej zrealizować cel.

Elementami składowymi każdej struktury organizacyjnej są komórki (stanowiska pracy, biura, działy, wydziały, zakłady, dywizje itp.) i powiązania zwane więziami. Podstawowymi więziami są więzi liniowe i funkcjonalne, a więziami wspomagającymi są więzi techniczne i informacyjne. Opis więzi przedstawiono w tab. 1.

Tabela 1

Rodzaje więzi stosowanych w projektowaniu struktur organizacji

Więzi podstawowe		Więzi wspomagające	
liniowe	funkcjonalne	techniczne	informacyjne
1	2	3	4
(hierarchiczne, służbowe) – wyrażają formalne podporządkowanie: przełożony – podwładny. Ich najważniejszą cechą jest uprawnienie przełożonego do decydowania o zadaniach, które powinien wykonać podwładny	(doradcze) – obowiązkowe lub nieobowiązkowe służeńie radą innym członkom organizacji. Zaliczamy do nich: – więzi służbowe między funkcjonalnym przełożonym (czyli takim, który wydaje tylko decyzje w ramach określonych funkcji) a podwładnymi,	zależność jednostek biorących udział w jednym procesie technologicznym (np. robotnicy przy taśmie). Polega na wzajemnym uzależnieniu członków zespołu w działalności związanej z wymianą zasileń	przepływ informacji pomiędzy jednostkami. Polega na jednostronnym lub wzajemnym informowaniu się o wszelkich stanach rzeczy i ich zmianach

cd. tabeli 1

1	2	3	4
	– więzi wspomaganie między członem funkcjonalnie uzależniającym a członem funkcjonalnie uzależnionym. Polega na doradzaniu lub pomaganiu – nie dysponowaniu		

1.4. Rodzaje struktur

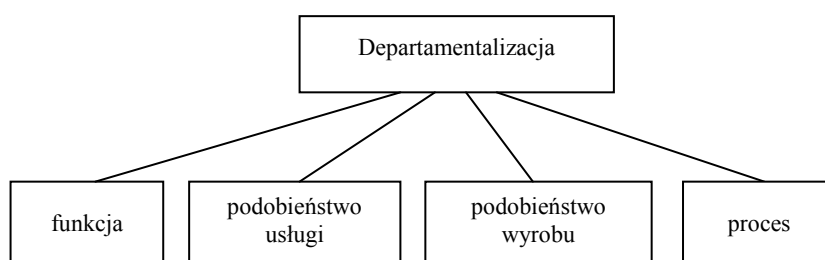
Do projektowania struktur można użyć różnego rodzaju form graficznych. Ze względu na rodzaj więzi organizacyjnych można wyróżnić struktury:

- liniowe,
- liniowe wzbogacone doradcami funkcjonalnymi,
- funkcjonalne,
- sztabowe (sztabowo-liniowe).

Ze względu na rozpiętość kierowania można wyróżnić struktury:

- smukłe – duża ilość szczebli kierowania, mała rozpiętość kierowania;
- płaskie – duża rozpiętość kierowania – wiele działów podlega jednemu kierownikowi, niewiele szczebli pośrednich, szybkie procesy decyzyjne.

Istotnym czynnikiem przy projektowaniu organizacji jest zasada departamentalizacji, czyli grupowania zadań. Może się ona odbywać według różnych kryteriów (rys. 4).



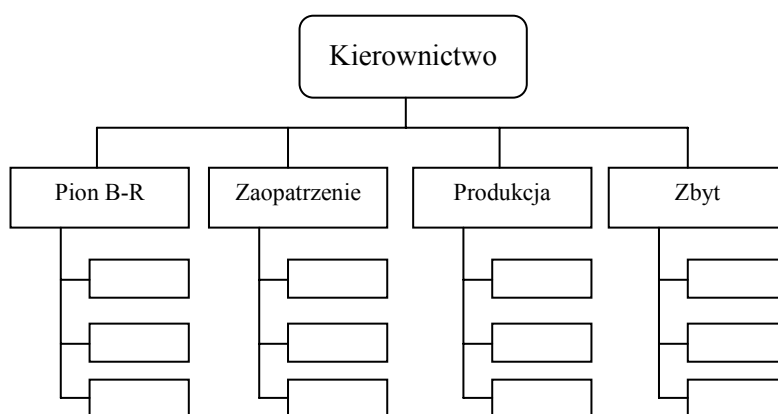
Rys. 4. Podział grupowania zadań według kryteriów

Projekt struktury organizacji, opartej na zasadzie grupowania zadań (departamentalizacji), stosuje się wówczas, gdy przedsiębiorstwo wytwarza dość jednolity i standardowy towar, stabilny pod względem technologicznym. Pozwala to utrzymać na zadowalającym poziomie komunikację pomiędzy specjalistami.

Wadą tego rozwiązania jest brak wyspecjalizowanego monitoringu linii produkcyjnych zaliczanych do szczebla operacyjnego, stąd też przy czynniku departamentalizacji najczęściej jest tworzona struktura liniowa (rys. 5).

1.4.1. Struktura liniowa

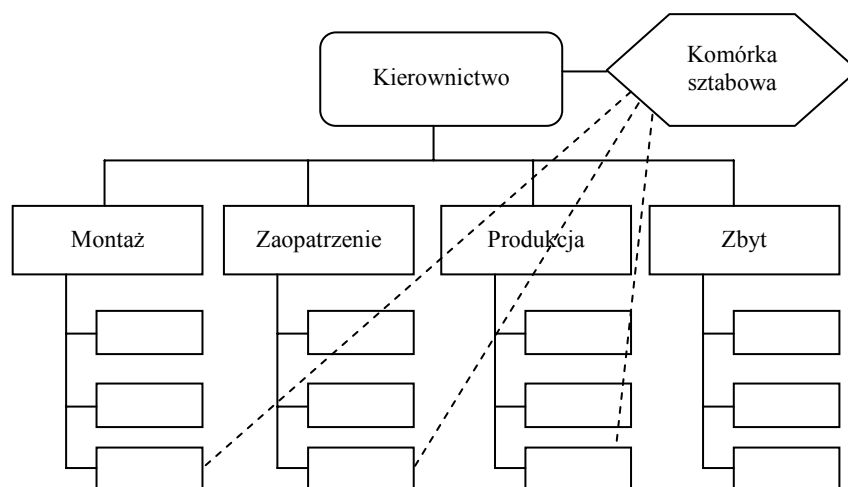
Struktura liniowa charakteryzuje się dużą stabilnością, niedającą się łatwo przystosować do zmiennych warunków, skutecznie tłumiącą inicjatywy pracowników. Ponadto w strukturze tej kierownicy muszą posiadać szeroki zakres wiedzy ze względu na potrzebę udzielania kompetentnych porad.



Rys. 5. Struktura liniowa organizacji

1.4.2. Struktura liniowo-sztabowa

Innym rodzajem struktury jest struktura liniowo-sztabowa (rys. 6). Jej powstanie jest spowodowane nieustannym wzrostem i rozwojem organizacji. W związku z tymi procesami narasta potrzeba pionowej koncentracji w przedsiębiorstwach. Taka sytuacja zdarza się wówczas, gdy istnieje wiele działów wspomagania i doradztwa.



Rys. 6. Struktura liniowo-sztabowa organizacji

W strukturach liniowo-sztabowych wykonywane funkcje, będące bezpośrednim wypełnianiem misji organizacji, są traktowane jako działania liniowe. Współpraca pomiędzy linią a sztabem nie może być konfliktowa, ale z uwagi na konieczność współpracy działów często dochodzi do rozdzwieńków, co staje się poważnym zakłóceniem prac firmy.

1.4.3. Struktura macierzowa

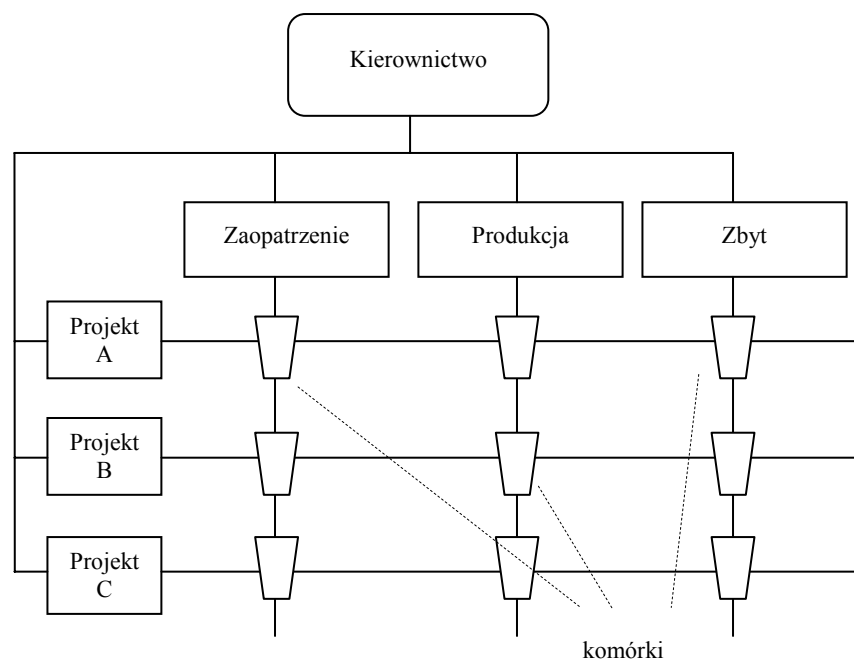
Odrębnym, nowym typem struktury organizacji jest struktura macierzowa (rys. 7). Ma ona zastosowanie w przypadku wystąpienia potrzeby realizacji pewnych przedsięwzięć specjalnych, które wymagają powołania zespołów projektowych. Jej zasadniczym przeznaczeniem jest praca nad skomplikowanymi problemami, które realizują organizacje zajmujące się głównie wykonywaniem jednorazowych projektów. Menedżerowie projektów i kierownicy działów dzielą odpowiedzialność za ustalanie priorytetów i zarządzanie pracą pracowników przydzielonych do projektów. Menedżer projektu odpowiada za jego sukces, natomiast kierownicy działów za dostarczenie mu zasobów niezbędnych do jego realizacji.

Członkami zespołów projektowych są pracownicy stałych komórek organizacyjnych. Komórki, które są najczęściej doraźnymi zespołami roboczymi, stanowią elementy macierzy. Skupiają się one na wykonywaniu jednego zadania. To umożliwi lepsze wykorzystanie energii twórczej i umiejętności ich członków oraz właściwe przydzielanie specjalistów do realizowanych zadań. Do-

bieranie zespołów może polegać na przyporządkowaniu kierownikom funkcjonalnym osób należących do konkretnych zespołów albo na tym, że każdy z kierowników funkcjonalnych, biorących udział w realizacji projektu, może być odpowiedzialny za wykonanie części zadania, jak gdyby pełnił rolę podwykonawcy. Zazwyczaj kolumny macierzy są odpowiednikami przetwarzanych funkcji, wiersze natomiast określają zadania, które są wykonywane zgodnie z doraźnymi projektami czy nowymi typami produktów. Również kierownicy funkcjonalni mają równorzędne i równoprawne pozycje co kierownicy zadań i projektów. Towarzyszy temu zjawisko dualizmu uprawnień decyzyjnych. Pracownicy otrzymują polecenia z dwóch różnych źródeł.

Struktura macierzowa występuje najczęściej w organizacjach skupiających wysoko kwalifikowanych specjalistów wdrażających projekty lub w przypadku wystąpienia potrzeby realizacji pewnych przedsięwzięć specjalnych, które wymagają powołania zespołów projektowych.

Dominującą rolę odgrywa tutaj ograniczony czas. Menedżer może w tym okresie korzystać z usług pracowników o rozmaitych specjalnościach funkcjonalnych.



Rys. 7. Struktura macierzowa organizacji

Struktura ta nosi nazwę macierzy z tego względu, że kierownicy w płaszczyźnie pionowej nadzorują specjalistów danych dziedzin, a w płaszczyźnie poziomej członkowie zespołów, będących podstawowymi komórkami organizacji, są podwładnymi też innych kierowników projektu. W strukturze tej jest naruszona zasada jedności kierownictwa, bo specjaliści-pracownicy mają dwóch przełożonych.

Aby zaprojektować najlepszy system dla organizacji, należy wziąć pod uwagę równocześnie technikę, otoczenie i cykl życia organizacji.

Na organizację można również spojrzeć z punktu widzenia przepływu informacji i pod tym kątem projektować jej strukturę.

1.4.4. Struktura oparta na modelu przetwarzania informacji

Organizacja to sieć przetwarzania informacji. Im większa jest niepewność związana z działaniem organizacji, tym więcej danych musi być przetworzonych i przesyłanych. Struktura organizacji, tworzona według tego kryterium, powinna być tak zaprojektowana, aby możliwa była redukcja niepewności. Temu celowi służą cztery rozwiązania projektowe:

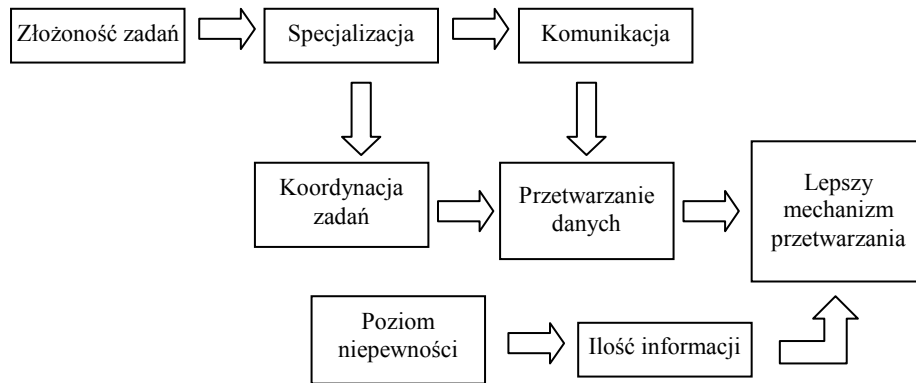
1. Zastosowanie zasobów rezerwowych.
2. Utworzenie samowystarczalnych komórek organizacyjnych.
3. Zastosowanie systemów informacyjnych.
4. Zastosowanie powiązań poziomych.

Ilość informacji potrzebnych do wykonania każdego zadania zależy od:

- różnorodności wyjść (różnorodność produktów, różnorodność usług, liczba klientów),
- różnorodności wejść (liczba różnych technicznych specjalności zaangażowanych w projekt, liczba maszyn, narzędzi itp.),
- stopnia trudności celu (mierzonego przy użyciu różnorodnych kryteriów efektywności).

Im większa różnorodność wejść i wyjść oraz trudności zadań, tym większą liczbę czynników i interakcji trzeba brać pod uwagę podejmując decyzję, a z tym jest ściśle związana potrzeba dostępu do (posiadania) informacji.

Na rys. 8 zaprezentowano model przetwarzania informacji, wpływający na strukturę organizacji pod kątem informatycznym. W modelu przedstawiono czynniki przyczyniające się do zwiększania potrzeb informacyjnych, które znacząco wpływają na projektowanie.



Rys. 8. Projektowanie organizacji poprzez model przetwarzania informacji

Zwiększająca się ilość informacji dla potrzeb decyzyjnych jest związana z następującymi zdarzeniami:

1. Gdy realizowane zadania w firmie są skomplikowane, to wymagana jest wysoka specjalizacja.
2. Specjalizacja narzuca konieczność bezpośredniego porozumiewania się ze sobą (komunikacja).
3. Duża ilość specjalistów generuje koordynację zadań.
4. Koordynacja pracy w organizacji wymaga przetwarzania dużej ilości danych.
5. Gdy rośnie poziom niepewności, to musi narastać ilość informacji dla jego redukcji.
6. Wzrost informacji wpływa na wypracowanie lepszych mechanizmów przetwarzania.

Podsumowanie

Omówione sposoby projektowania struktury powinny również uwzględniać hierarchię władzy w organizacji. Praktycznie wygląda to tak, że dla struktur liniowych władza jest czerpana z pozycji formalnej. Stanowiska liniowe są zawarte w rdzeniu operacyjnym, szczeblu średnim (taktycznym) i wierzchołku strategicznym. Za osiągnięcie celów jest odpowiedzialne kierownictwo zaliczane do szczebla taktycznego.

W strukturach sztabowych specjaliści są zgrupowani w sztabie. Realizują działalność pomocniczą: służą radą, doświadczeniem i wspierają stanowiska liniowe. W modelu organizacji stanowiska sztabowe wchodzi w skład technostruktury i personelu pomocniczego. Stanowiska sztabowe czerpią swą władzę z wiedzy.

Literatura

- [GIHe06] Glinka B., Hensel P.: *Projektowanie organizacji*. Wydawnictwo Naukowe Uniwersytetu Warszawskiego, Warszawa 2006.
- [GrHe04] Grudzewski W.M., Hejduk J.K.: *Metody projektowania systemów zarządzania*. Difin, Warszawa 2004.
- [Grif02] Griffin R.W.: *Podstawy zarządzania organizacjami*. Wydawnictwo Naukowe PWN, Warszawa 2002.
- [Mach99] Machaczka J.: *Podstawy zarządzania*. Wydawnictwo Akademii Ekonomicznej, Kraków 1999.
- [Stab98] Stabryła A.: *Podstawy zarządzania firmą. Modele, metody, praktyka*. Antykwa, Kraków-Kluczbork 1998.

DESIGNING ORGANIZATION

Summary

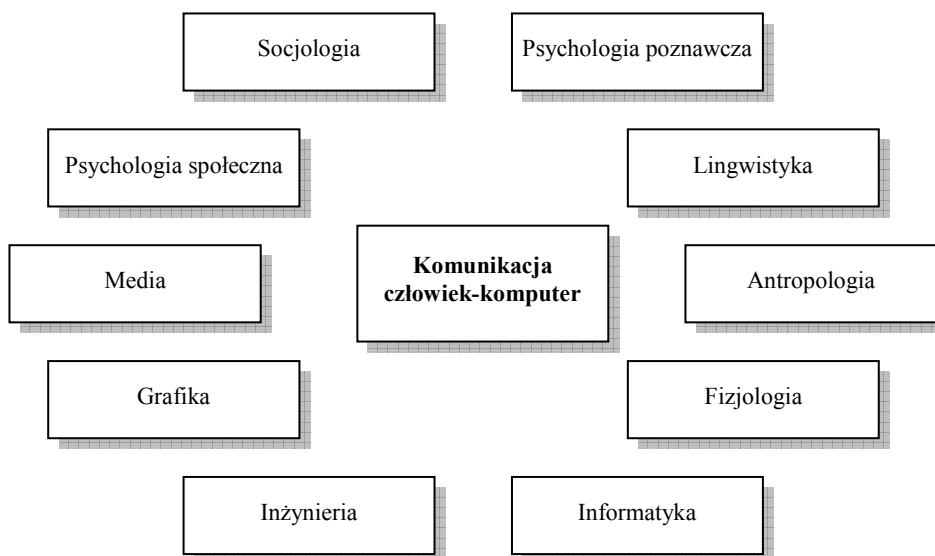
The chapter includes consideration on basic principles of designing organization. At first, author explains what is organization and what are its goals and functionalities for business. Next, author defines the designing organization and presents organizational structure development principles. The main part of the chapter covers presentation of different approaches to the designing organization, applicable for different organizational structures.

Sebastian Kostrubała

KOMUNIKACJA CZŁOWIEK-KOMPUTER

Wprowadzenie

Komunikacja człowiek-komputer (*human computer interaction*) jako termin naukowy pojawił się w literaturze we wczesnych latach 80. W tym okresie następowała zmiana w definiowaniu i podejściu do użytkownika komputera. Projektanci systemów komputerowych zauważyli i zaakceptowali zmianę głównego odbiorcy systemów z profesjonalistów na ludzi niezwiązanych zawodowo z informatyką. Oznaką takiego podejścia było zdefiniowanie terminu „naiwnego użytkownika” przez Easona w 1976 roku [Easo76, s. 3-7] oraz wypowiedź z 1983 roku wiceprezesa firmy IBM prof. Branscomba, odpowiedzialnego za badania naukowe, który stwierdził, że komputery przestają być ekskluzywnymi narzędziami dla naukowców, a stają się powszechne i niezbędne dla wszystkich. W tym okresie firma IBM poświęciła większość swoich badań zagadnieniom dotyczącym łatwego użycia komputera. Lata 80. to rozkwit komputerów osobistych, stąd nie może dziwić ilość prowadzonych w tym czasie badań. Sama problematyka komunikacji pomiędzy człowiekiem a maszyną pojawiła się jednak w badaniach naukowych znacznie wcześniej. Przykładem może być założone w 1949 roku towarzystwo naukowe Ergonomics Research Society. Zostało one powołane po to, by zająć się m.in. badaniami nad „dopasowaniem maszyny do człowieka”. Członkowie tego towarzystwa już wtedy zauważali, że problem komunikacji człowiek-maszyna jest problemem interdyscyplinarnym, łączącym budowę maszyn z takimi dziedzinami, jak np. anatomia, czy psychologia [WWW8]. Wraz z rozwojem badań nad interakcją użytkownika z komputerem, wyróżniono znacznie więcej powiązanych ze sobą dyscyplin. Przykładowe powiązania pokazano na poniższym rysunku.



Rys. 1. Interdyscyplinarność problemu komunikacji człowiek-komputer

Źródło: [SaJB08, s. 535].

Przyglądając się rysunkowi, można dojść do wniosku, że idealny projektant interaktywnych systemów będzie dysponował wiedzą i doświadczeniem z zakresu: psychologii, nauk poznawczych, ergonomii i informatyki. Oprócz zrozumienia zasad i wymagań dotyczących samego procesu projektowania, będzie on musiał posiadać także umiejętności związane z inżynierią oprogramowania, potrzebne do zbudowania niezbędnych narzędzi. Naturalne w procesie projektowania i budowy interfejsów wydają się być umiejętności graficzne. Dobrze byłoby, gdyby projektant, oprócz umiejętności inżynierskich, potrafił zrozumieć szerszy kontekst interakcji użytkownika, czyli posiadał wiedzę z socjologii. Oczywiście znalezienie tak opisanego, idealnego projektanta systemów interaktywnych jest w rzeczywistości niemożliwe. Gorzej, że również stworzenie takiego zespołu projektowego, który posiadałby specjalistów z każdej z wymienionych dyscyplin, należy do rzadkości. W związku z tym, w praktyce obserwuje się tendencję do projektowania interakcji pomiędzy człowiekiem a komputerem z punktu widzenia określonej dyscypliny lub kilku głównych dziedzin, najczęściej: informatyki, psychologii i nauk poznawczych [DFAB04, s. 5]. Próbę opisaną jaką rolę odgrywają poszczególne dziedziny w procesie projektowania można odnaleźć m.in. w pracy Paula Booth'a [Boot89, s. 7-12].

1. Modele projektowania systemów interaktywnych

Tradycyjne podejście do projektowania systemów komputerowych nie sprawdza się podczas projektowania interakcji pomiędzy człowiekiem a komputerem. Model kaskadowy oraz późniejsze modele uwzględniające iterację pomiędzy poszczególnymi działaniami są niewystarczające, głównie ze względu na to, że koncentrują się na technicznym aspekcie działania systemu [SmAt06, s. 48]. Potrzebę innego podejścia do projektowania systemów interaktywnych dostrzeżono już w latach 80., gdy rozpoczęto na szeroką skalę badania nad komunikacją człowiek-komputer. Podstawy postępowania przy projektowaniu tego rodzaju systemów opracowali Gould i Lewis, formułując 3 zasady:

- zbliżenie do użytkownika i zadań, które powinny być realizowane (zrozumienie użytkownika i jego zachowań),
- empiryczny pomiar wydajności użytkownika (poprzez analizę jego pracy z prototypami),
- projektowanie iteracyjne [GoLe85].

Zmiana podejścia projektantów z ukierunkowania się na aspekty techniczne systemu na zwrot ku problemom użytkownika stała się podstawą nowego podejścia zaproponowanego przez profesora Normana – projektowaniu zorientowanym na użytkownika (*user-centered design*) – [Norm88]. W ramach tego nurtu powstało wiele modeli i sposobów projektowania systemów opisanych przez naukowców zajmujących się tą problematyką (m.in. Nielsen – znany ze swych wieloletnich badań nad użytecznością, Kritzberg, Dix, Preece, Beyer, Cooper). Warto przypomnieć model profesora Nielsena, ponieważ był on jednym z pierwszych i stał się inspiracją dla kolejnych autorów. Zaproponował on jedenaście etapów tworzenia interaktywnych systemów:

1. Poznanie użytkownika:
 - a) indywidualne cechy użytkownika,
 - b) bieżące i przyszłe zadania użytkownika,
 - c) analiza funkcjonalna,
 - d) ewolucja użytkownika i pracy.
2. Analiza konkurencyjności.
3. Określenie celów użyteczności.
4. Projektowanie równoległe.
5. Współprojektowanie.
6. Skoordynowane projektowanie całego interfejsu.
7. Stosowanie wytycznych i analizy heurystycznej.
8. Prototypowanie.

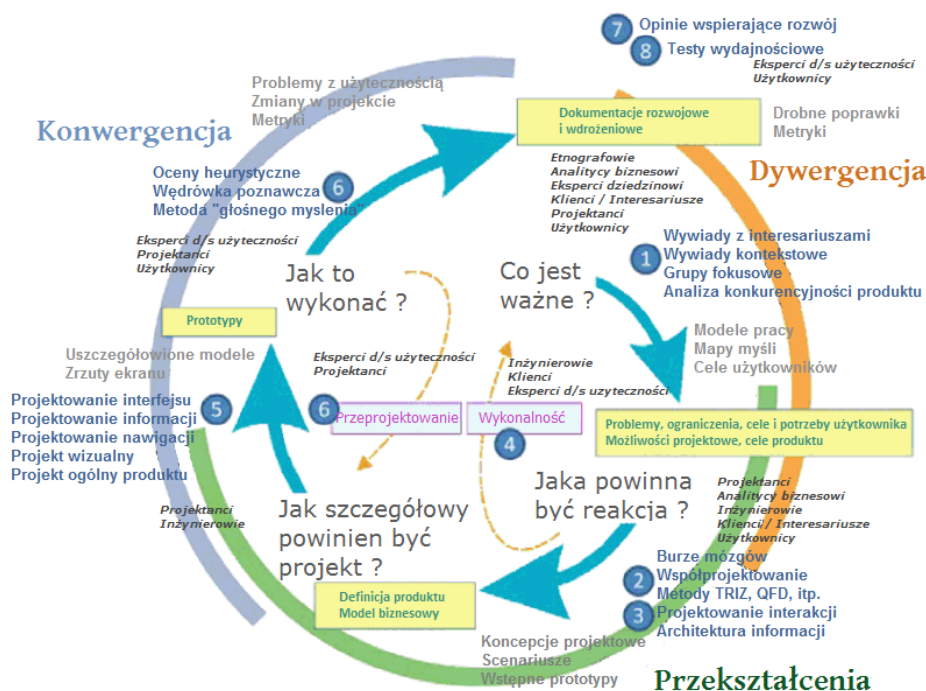
9. Testy empiryczne.
10. Projektowanie iteracyjne
11. Zbieranie informacji zwrotnych z pola użycia [Niel93, s. 71-110].

Należy przy tym zaznaczyć, że Nielsen opracował zestaw kroków i ograniczył się do stwierdzenia, że przejście przez nie wszystkie nie jest wymagane. Pozostali specjaliści dziedziny, rozwijając temat i opierając się na swoim doświadczeniu zasugerowali, które z tych kroków raczej powinny być wykonane, a które można pominąć.

Kolejnym znanym podejściem jest projektowanie kontekstowe zaproponowane przez Beyera i Holtzblatt (dość powszechnie stosowane przez zespoły projektowe stosujące metody zwinne). Podejście to integruje kilka technik projektowania zorientowanego na użytkownika w jeden proces projektowy, którego kluczowym problemem jest określenie tego jak użytkownik będzie pracować w przyszłości. To jak system ma działać, jaka ma być jego struktura i jakie zostaną użyte do jego stworzenia technologie jest zdeterminowane zebranymi danymi od użytkowników [BeHo98, s. 3].

W większości przypadków wymienieni wcześniej autorzy tworzyli swe podejścia projektowe kilkanaście lat temu. Nie oznacza to jednak, że są one nieaktualne lub też zamykają możliwości tworzenia rozwiązań opartych na komunikacji człowiek-komputer. Aktualność podejścia zorientowanego na użytkownika potwierdzają swoimi zaleceniami wiodące firmy informatyczne, takie jak np. IBM [WWW12]. Na podstawie tego podejścia został również opracowany w 1999 roku międzynarodowy standard ISO 13407, opisujący proces projektowania komputerowych systemów interaktywnych (*Human-centred design for interactive systems*) zorientowany na człowieka (szerszy kontekst niż projektowanie zorientowanie na użytkownika czy konsumenta). Dostarcza on wymagania i rekomendacje zasad oraz czynności poprzez cały cykl życia systemu interaktywnego. Standard ten został zastąpiony w 2010 roku przez ISO 9241-210. Na temat problematyki projektowania systemów interaktywnych powstaje nadal duża ilość publikacji naukowych, a o wadze problemu mogą świadczyć liczne konferencje naukowe poświęcone tej tematyce, m.in. cykl konferencji Human-Computer Interaction, czy Human-Centred System Engineering. Nadal trwają poszukiwania jak najlepszego modelu projektowania oraz najlepszych wzorców postępowania dla zadań wchodzących w skład tych modeli. Przykładem może być publikacja z 2010 roku, w której autorzy próbują po raz kolejny stworzyć model oparty na najlepszych do tej pory praktykach projektowych [JoSa10, s. 166-181]. Zaproponowany model składa się nie tylko z opisu

8 konkretnych działań, ale zawiera również inne użyteczne informacje projektowe, takie jak: przynależność do określonych faz, uczestnicy biorący udział w kolejnych czynnościach projektowych, metody jakie powinny być stosowane podczas tych czynności i efekty działań, którymi poszczególne działania powinny się zakończyć.



Rys. 2. Proces projektowania systemów interaktywnych

Źródło: [JoSa10, s. 169].

Duża różnorodność dostępnych modeli projektowania systemów interaktywnych nie zmienia faktu, że istnieje kilka podstawowych zasad, które w przypadku tych systemów powinny być spełnione. Udane systemy interaktywne to takie, które są:

- użyteczne: robią to do czego zostały zaprojektowane,
- możliwe do użycia: pozwalają na naturalną interakcję, bez zagrożeń lub nie spodziewanych błędów,
- używalne: sprawiają, że ludzie chcą ich używać.

Istnieją prawne regulacje określające jak powinna wyglądać interakcja z użytkownikiem, zabezpieczające potencjalnych użytkowników przed oprogramowaniem niespełniającym powyższych założeń. Przykładem może tu być dyrektywa Unii Europejskiej dotycząca minimalnego bezpieczeństwa i zdrowia przy pracy z ekranami monitorów (90/270/EEC), nieograniczająca się jedynie do określania norm i standardów zdrowotnych, a wymuszająca także projektowanie systemów zgodnych z wytycznymi:

- a) oprogramowanie musi odpowiadać zdefiniowanym zadaniom,
- b) oprogramowanie musi być łatwe w użyciu, a tam gdzie to możliwe, dostosowane do poziomu wiedzy lub doświadczenia operatora; nie można używać bez wiedzy pracownika żadnych urządzeń do kontroli jakościowej lub ilościowej,
- c) systemy muszą zapewniać przekazywanie pracownikom informacji zwrotnej o ich działaniu,
- d) systemy muszą wyświetlać informacje w formacie i tempie dostosowanym do operatorów,
- e) muszą być stosowane zasady ergonomii oprogramowania, w szczególności przy przetwarzaniu danych dotyczących ludzi [WWW9].

Wymienione powyżej zasady są jednymi z wielu dostępnych zasad projektowania przeznaczonych dla projektantów systemów interaktywnych. Niektóre z tych zasad zostały opisane w dalszej części tekstu.

2. Projektowanie interfejsu

Projektowanie interfejsu użytkownika jest jednym z wielu zadań składających się na proces projektowania systemów interaktywnych (rys. 2). Przyjmując jednak podejście zorientowane na użytkownika jest to ta część projektowanego systemu, która jest najbliższa użytkownikowi. Metoda prototypowania systemów interaktywnych w głównej mierze opiera się na generowaniu kolejnych modyfikacji interfejsu użytkownika. Przyglądając się cyklowi życia systemu interaktywnego, można zauważyć, że już w początkowej fazie projektowania powstają wstępne prototypy interfejsu w postaci szkiców formatek oraz scenariuszy kilku lub kilkunastu formatek, ułożonych w sekwencje mające imitować nawigację użytkownika po systemie. W tej wczesnej fazie opracowywania interfejsu niezwykle ważny jest aktywny udział użytkowników, którzy na tym interfejsie mają pracować. Wspólne projektowanie interfejsu realizuje koncepcję zorientowania na użytkownika, niesie jednak ze sobą zagrożenie wydłużenia procesu projektowego lub skupienia się na rzeczach mało istotnych w tej fazie

projektowej (kolory, kształty itp.). W związku z tym warto wykorzystać praktyczne metody wspólnej pracy z użytkownikiem nad projektem interfejsu. Jedną z takich metod jest przeprowadzenie sesji JAD (Joint Application Development) z udziałem użytkowników, analityków, projektantów i programistów. W przypadku gdy sesja JAD jest poświęcona działaniu całego systemu, wśród uczestników powinni również znaleźć się menedżerowie. Kluczem do powodzenia sesji jest wyodrębnienie lidera z dużymi umiejętnościami organizacyjnymi i interpersonalnymi [ShCR09, s. 141]. Jego zadaniem jest sterowanie procesem projektowania oraz ucinanie rozważań nad kwestiami nieistotnymi. Jeżeli projektowanie systemu odbywa się według jednego z wcześniej wymienionych modeli, na tym etapie powinny istnieć już zebrane wymagania użytkowników dotyczące systemu oraz być wyodrębnieni aktorzy i ich przypadki użycia. W takiej sytuacji jedną z metod jest wychodzenie od wyodrębnionych przypadków użycia do konkretnych projektów interfejsów realizujących wymagania użytkownika związanego z omawianym problemem. W celu silniejszego aktywowania użytkownika w działaniach projektowych, wstępne projekty są opracowywane wspólnie na papierze albo w postaci rysunków, albo w postaci dużego arkusza papieru, gdzie są naklejane karteczki z oznaczeniami komponentów, które powinny zostać użyte do realizacji konkretnego interfejsu. Prototypy papierowe pozwalają na opisanie ogólnych cech interfejsu, na ich podstawie projektanci mogą opracować bardziej złożone projekty interfejsu już z użyciem narzędzi komputerowych pozwalających na tworzenie projektów interfejsu, takich jak: Microsoft Visio, czy Enterprise Architect. Dopuszczalne jest także użycie środowisk programistycznych pozwalających na projektowanie interfejsów (takich jak Microsoft Visual Studio, czy Eclipse). Na tym etapie tworzenia systemu interaktywnego nie zawsze jest to jednak wskazane, ponieważ efektem działania takich programów są działające prototypy, które w przypadku niedoświadczonych użytkowników sugerują finalny wygląd i działanie interfejsu, co prowadzi do niepotrzebnych nieporozumień na linii użytkownik-projektant. Dodatkowym atutem użycia narzędzi „czysto” projektowych jest możliwość tworzenia scenariuszy, czyli sekwencji projektów interfejsów wizualizujących realizację określonego przypadku użycia. Scenariusze w przeciwieństwie do projektów pojedynczych ekranów interfejsu pozwalają zaprojektować wstępną nawigację po systemie. Tworzenie scenariuszy i wstępnych prototypów odbywa się na w miarę ogólnym poziomie, dlatego projektanci mogą zadbać o całą strukturę interfejsu systemu. Jest to dość ważne, by na każdym etapie projektowania pamiętać również o całym rozwiązaniu, a nie skupiać się tylko i wyłącznie na pojedynczych ekranach systemu. Takie podejście zapewnia spójną nawigację, sposób działania i wygląd w obrębie całego systemu.

Kolejnym etapem jest poddanie szkicowych projektów analizie i ocenie, a następnie cały zespół projektowy wraz z użytkownikami musi podjąć decyzję o wstępnej akceptacji opracowanych interfejsów lub ponownym ich przeprojektowaniu. Zaakceptowane prototypy powinny zostać przekształcone w prototypy funkcjonalne, które już w sposób bezpośredni imitują docelowy wygląd interfejsu. W tym celu należy wykorzystać wymienione wcześniej środowiska programistyczne, w przypadku projektowania aplikacji internetowych mogą to być wstępne, opracowane strony w języku HTML. Na tym etapie tworzeniem prototypów zajmują się specjaliści, których zadaniem jest zaproponowanie takich interfejsów, które po pierwsze są rozwinięciem prototypów powstałych we wcześniejszych krokach, a po drugie są zrealizowane zgodnie z obowiązującymi regułami i wskazówkami projektowymi.

Działające prototypy powinny zostać poddane operacji testowania i ocenia. Testowanie interfejsu może odbywać się za pomocą różnych metod i technik, których wspólnym celem jest sprawdzenie i dalsze doskonalenie projektu interfejsu. Najczęściej spotykanymi podejściami do testowania i ocenia interfejsu są: oceny heurystyczne, metoda przejścia się po interfejsie, oceny interaktywne i formalne testowanie użyteczności interfejsu [DeWR08, s. 323]. Ocena heurystyczna jest zazwyczaj realizowana przez jedną z osób zespołu projektowego, który sprawdza czy zaprojektowany interfejs spełnia określone zasady i wzorce projektowe. Również w metodzie przejścia przez interfejs obecny jest członek zespołu, który prezentuje końcowemu użytkownikowi prototypy interfejsu, a ten zgłasza swoje uwagi. Metody oceny interaktywnej i testów użyteczności są realizowane już bezpośrednio przez użytkowników pracujących na systemie. W pierwszej z nich testowanie przez użytkownika odbywa się na zasadzie podoba/nie podoba się, w drugiej testy odbywają się w sposób formalny, najczęściej w specjalistycznych laboratoriach mierzących sprawność pracy użytkownika z danym interfejsem. Formalne testy użyteczności są uznawane za najdroższe i są stosowane zazwyczaj przy większych projektach. Do prawidłowego przeprowadzenia testów użyteczności wystarczy 5-10 użytkowników [WWW11].

Proces projektowania interfejsu jest procesem iteracyjnym, dlatego po wykonaniu fazy ocen i testów następuje powrót do fazy tworzenia prototypów, gdzie następuje poprawa lub opracowanie nowych prototypów interfejsu.

3. Zasady i wzorce projektowe

Podobnie jak w przypadku modeli całego cyklu życia systemów interaktywnych, istnieje także ogromna ilość proponowanych technik: wskazówek, wzorców, reguł, konwencji, czy standardów projektowania interfejsu. Techniki te dodatkowo można podzielić na zależne i niezależne od platformy, dla której

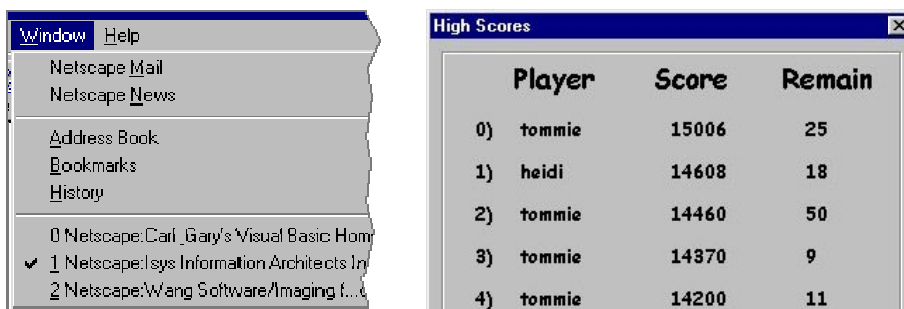
jest przeznaczony projektowany system. Przykładowo, inne wskazówki stosuje się do projektowania interfejsów aplikacji przeznaczonych dla systemu Mac OS X firmy Apple [WWW4], a inne dla aplikacji działających pod systemem Windows firmy Microsoft [Micr95].

Proces projektowania interfejsu jest zawsze procesem kreatywnym, każdy projekt jest inny i w końcowym efekcie otrzymuje się różnorodne interfejsy użytkownika. Dodatkowym problemem jest różnorodność użytkowników oraz ich nieograniczony potencjał używania interfejsu do celów zupełnie nieprzewidzianych przez projektanta. W związku z tym przyjmuje się, że nawet najlepszy zaprojektowany samodzielnie przez projektanta interfejs (z użyciem wszystkich dostępnych wskazówek i zasad projektowania), będzie nieodpowiedni dla użytkownika. Jest to jeden z 11 sloganów użyteczności zdefiniowanych przez Nielsen, które w zwężonej formie oddają problematykę tworzenia użytecznych interfejsów:

1. Twoje najlepsze pomysły nie będą wystarczająco dobre.
2. Użytkownik zawsze ma rację.
3. Użytkownik nie zawsze ma rację.
4. Projektanci nie są użytkownikami.
5. Użytkownicy nie są projektantami.
6. Wiceprezesi nie są użytkownikami.
7. Mniej znaczy więcej.
8. Szczegóły mają znaczenie.
9. System pomocy nie zawsze jest konieczny.
10. Inżynieria użyteczności jest procesem [Niel93, s. 10-16].

Na pierwszy rzut oka widać, że część wymienionych sloganów jest wzajemnie sprzeczna. Użytkownik ma zawsze rację, bo to on będzie pracował na systemie, ale z drugiej strony często nie zna swoich potrzeb i nie zawsze jest w stanie wyobrazić sobie najlepszego rozwiązania. Podobnie jak projektanci, którzy przez to, że również są użytkownikami systemów komputerowych przenoszą swoje doświadczenia na projektowany interfejs, zapominając, że są zupełnie innym typem użytkownika niż docelowy odbiorca. Wymienione przez Nielsena slogany doczekały się rozwinięcia i można obecnie znaleźć w literaturze, jak i w Internecie ich znacznie większą ilość [WWW3].

Nie tylko projektanci próbują przenosić swoje doświadczenia, robią to również programiści (choć często w sposób nieświadomy). Czasami efekty takich działań wymykają się testerom i można je zaobserwować w finalnych wersjach oprogramowania. Poniżej znajdują się dwa przykłady, które nie są specjalnie uciążliwe dla użytkowników, ale pokazują inny sposób postrzegania świata przez programistów. Zwykli użytkownicy numerują elementy od wartości 1, natomiast programiści w swoich językach programowania odliczają zazwyczaj od 0, co czasami przenoszą na programowane interfejsy.



Rys. 3. Numerowanie od 0 – przenoszenie doświadczeń na interfejs

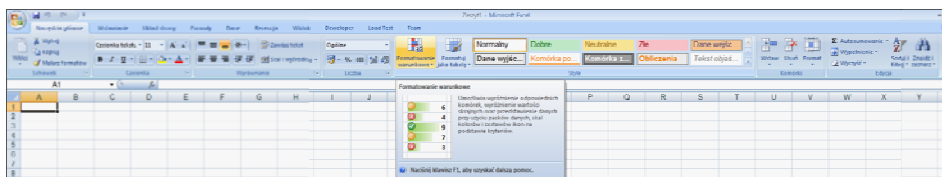
Źródło: Zrzuty ekranów z systemów Netscape Navigator i gry xBlock, za stronę [WWW5].

Większość ze sloganów ma swoje odzwierciedlenie we wskazówkach i zasadach projektowych. Przykładowo jedną ze wskazówek dotyczącą projektowania interfejsu jest zasada czystego i prostego interfejsu, niepozwalającego użytkownikowi na zbyt dużą swobodę, za to dopasowanego do niego i do jego stylu pracy. Wskazówka ta odzwierciedla slogan „mniej znaczy więcej”, czyli jeżeli interfejs ma mniej elementów, za to bardziej dopasowanych do stylu pracy użytkownika, to użyteczność takiego interfejsu będzie większa, niż interfejsu z dużą ilością elementów, które mogą użytkownika wprowadzać w błąd.

Wskazówki projektowe są niezwykle cenne dla projektantów i są jednymi z najczęściej wykorzystywanych narzędzi w procesie projektowania interfejsu. Szczególnie przydatne są przy tworzeniu pierwszych funkcjonalnych prototypów interfejsu. Przykładowymi wskazówkami projektowymi może być lista podstawowych zasad i najlepszych praktyk z zakresu projektowania interfejsu proponowanych przez firmę IBM:

1. Łatwy dostęp do funkcji, które użytkownicy potrzebują przez większość czasu pracy, funkcje mniej potrzebne powinny być umieszczone w mniej widocznym miejscu.
2. Optymalizacja interfejsu pod kątem najczęstszych i najważniejszych z punktu widzenia użytkownika zadań.
3. Obiekty i informacje powinny być widoczne na interfejsie, a dostęp do nich powinien być prosty [WWW13].

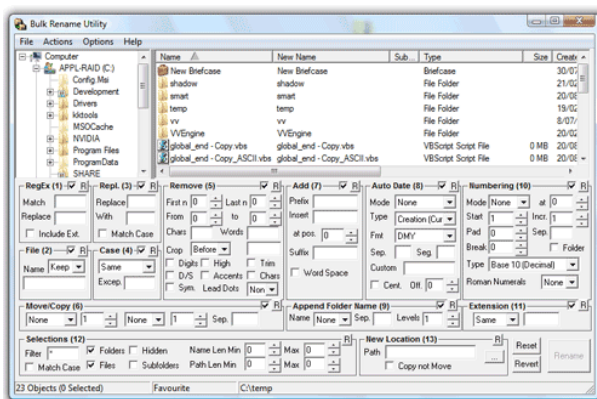
Koncepcje zawarte w pierwszych 3 wskazówkach projektowych są realizowane przykładowo przez obiekt wstążki w systemach Office.



Rys. 4. Wstążka systemu Microsoft Office

Źródło: Microsoft Office 2007.

4. Użycie właściwych wartości domyślnych powoduje kończenie zadań przez użytkownika relatywnie prosto i szybko.
 5. Interfejs powinien być elastyczny, tak by umożliwić użytkownikowi dostosowanie go do własnych potrzeb.
 6. Użytkownicy powinni być informowani o tym, co w danej chwili robi system (paski postępu oraz stanu).
 7. Elementy interfejsu, które wyglądają identycznie, powinny zachowywać się identycznie i dawać takie same rezultaty działania.
 8. System powinien umożliwiać użytkownikowi swobodę działania (która umożliwia popełnianie błędów) bez wywoływania strachu, że może on doprowadzić do uszkodzenia danych (przykładowo zastosowanie opcji: cofnij – ponów operację).
 9. Aplikacja powinna być przewidywalna dla użytkownika, dlatego powinna używać standardowych konwencji, gdzie to tylko możliwe (np. obsługa skrótów klawiszowych Ctrl-C, Ctrl-V, operacje drag & drop). Użycie powszechnie stosowanych wzorców projektowych sprawi, że użytkownik nie będzie musiał uczyć się interfejsu.
 10. Podczas projektowania systemu zawsze należy mieć na uwadze użytkownika końcowego.
 11. Należy unikać dodawania zbyt dużej ilości funkcji, ponieważ zwiększają one zakres możliwych działań, co może paraliżować użytkowników.
- Poniżej znajduje się zrzut ekranu z systemu pozwalającego na definicję reguł i zmianę nazw wielu plików równocześnie. Hasło reklamowe tego oprogramowania brzmi: „Zmieniaj nazwy wielu plików jednym przyciskiem. Wsadowa zmiana nazw jest prosta.”, jednak interfejs systemu jest tak przeładowany funkcjami, że nie wydaje się być prosty.



Rys. 5. Ekran główny systemu Bulk Rename Utility

Źródło: [WWW2].

12. Interfejs powinien być tak zaprojektowany, by mógł być bez problemu lokalizowany dla innych krajów bez potrzeby jego ponownego projektowania.
13. Podczas projektowania interfejsu należy pamiętać o użytkownikach niepełnosprawnych, którzy być może nie mogą posługiwać się klawiaturą, myszą, czy joystickiem.



Rys. 6. Klawiatura wirtualna

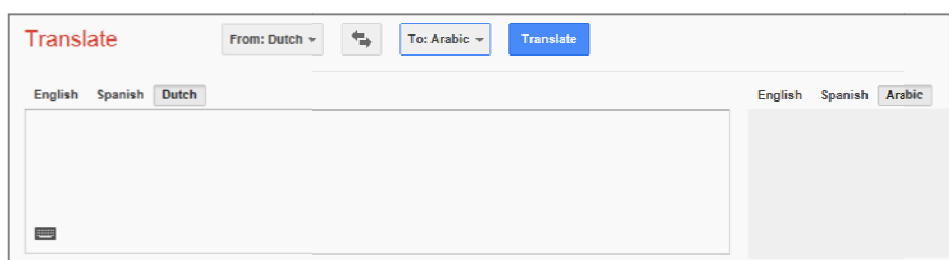
Źródło: System Microsoft Windows.

14. Pomoc kontekstowa powinna być dostępna dla użytkownika wszędzie tam gdzie może jej potrzebować.
15. Interfejs wizualny powinien być intuicyjny i przyjazny.
16. Interfejs powinien być przejrzysty i wizualnie prosty.
Poniżej została przedstawiona funkcjonalność wyboru par języków, dla których będzie dokonywane tłumaczenie zrealizowane w odmienny sposób. Pierwszy, pomimo że czytelny, zabiera mnóstwo miejsca, drugi jest wizualnie prostszy.



Rys. 7. Parametryzacja par języków, dla których ma być dokonywane tłumaczenie

Źródło: Zrzut ekranu z systemu firmy LEC za stronę: [WWW6].



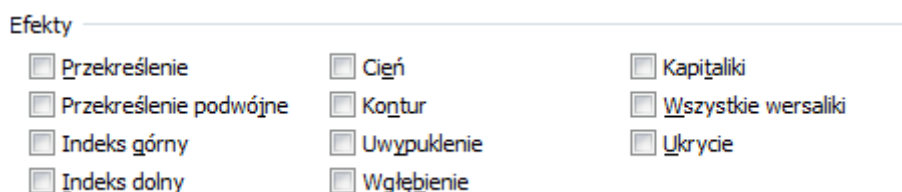
Rys. 8. Wybór języków do tłumaczenia w translatorze Google

Źródło: Zrzut ekranu ze strony.

17. Zasady projektowania interfejsu wizualnego:

- a) projektowanie subtraktywne – redukcja szumu informacyjnego poprzez eliminowanie elementów wizualnych niezwiązanych bezpośrednio z wizualną komunikacją z użytkownikiem;
- b) hierarchia wizualna – zrozumienie ważności zadań użytkownika i ustalenie wizualnej hierarchii tych zadań. Zadania ważniejsze powinny być bardziej widoczne, np. poprzez zmianę położenia, kontrastu, rozmiaru;
- c) afordancja – użytkownik powinien być w stanie w prosty sposób określić jaka czynność może być podjęta z danym obiektem wizualnym. Zasada ta dotyczy nie tylko niestandardowych elementów wizualnych, które mogą być niezrozumiałe dla użytkownika, ale również tradycyjnych elementów, które są już użytkownikowi znane.

Nawet największym firmom tworzącym oprogramowanie, takim jak Microsoft, zdarza się zaprojektować interfejs tak, by łamał zasadę afordancji. Przykładem jest Microsoft Word i okno dialogowe ustawień czcionki, gdzie w efektach użyto pól wyboru sugerujących na możliwość dowolnej kombinacji efektów. W praktyce jednak pola te działają jak pola opcjonalne uniemożliwiając wykonanie nieprawidłowych wyborów typu: kapitaliki i wersaliki równocześnie;



Rys. 9. Okno dialogowe ustawień czcionki

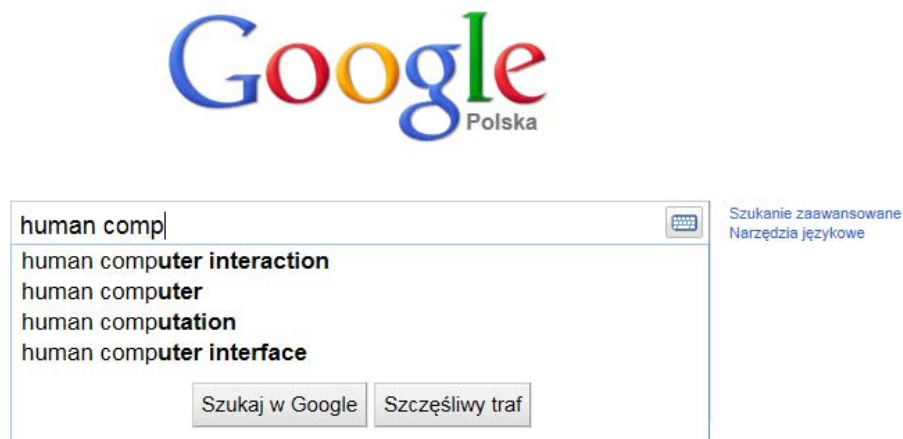
Źródło: System Microsoft Office Word.

- d) przestrzeń – brak oszczędzania na miejscu poprzez eliminowanie dodatkowych przestrzeni. Używanie „białych znaków” dających dodatkowe pole manewru dla użytkownika.

Zasad i wskazówek projektowania interfejsu można w literaturze znaleźć o wiele więcej. Dwaj wymienieni wcześniej autorzy – Nielsen i Norman również proponują swoje wskazówki projektowe (jako firma doradcza Nielsen Norman Group) – [WWW1], które w większości pokrywają się z tymi wymienionymi wcześniej. Wśród ich wskazówek projektowych można odnaleźć zasadę mówiącą o nienadużywaniu ilości kolorów. Autorzy zwracają uwagę na zjawisko zwane „ślepotą kolorów”. Pewna część populacji (ok. 10% mężczyzn) nie rozróżnia kolorów – różne odmiany daltonizmu, stąd też sugestia o nienadużywaniu kolorystyki. W Internecie można odnaleźć również artykuły obfitujące w praktyczne rady odpowiedniego doboru kolorów, tak by interfejs był czytelny dla większości użytkowników [WWW7].

Wskazówki projektowe są ciągle rozwijane i chętnie wykorzystywane przez projektantów systemów interaktywnych. Innym równie popularnym narzędziem są wzorce projektowe, które w przeciwieństwie do zasad projektowych nie operują na ogólnym poziomie, lecz oferują gotowe rozwiązania (zgodne z wskazówkami projektowymi). Wzorce projektowe powinno się zastosować w okreś-

lonych sytuacjach do realizacji z góry założonych celów projektowych. Przykładem dość powszechnie stosowanego wzorca jest autowypełnianie (*autocomplete*). Stosuje się go w sytuacji, gdy projektant interfejsu wie, że użytkownik będzie musiał wypełniać pewne pole swojego interfejsu wielokrotnie, często wpisując te same wartości, lub wartości te będzie musiał uzyskać w inny sposób. System automatycznego podpowiadania możliwych wartości przy zadanym kontekście w znaczący sposób ułatwia i przyspiesza pracę użytkownika.



Rys. 10. Przykład wykorzystania wzorca projektowego autowypełnianie

Źródło: Wyszukiwarka internetowa firmy Google.

Wzorce projektowe, oprócz samego rozwiązania, opisują również kontekst, w którym nie powinny być stosowane. W przypadku opisywanego tu autowypełniania, nie wolno go stosować w polach wrażliwych (służących np. do wpisywania haseł), w przypadkach gdy ilość dostępnych wartości w danym kontekście jest zbyt duża lub odwrotnie gdy lista wartości jest stała. W tym ostatnim przypadku należy po prostu stosować proste pole wyboru. Innymi znanymi i często spotykanymi wzorcami projektowymi są informacja na żądanie (*details on demand*), czy dynamiczne formularze, których wygląd zmienia się w trakcie wprowadzania informacji przez użytkownika, w zależności od kontekstu.

Podsumowanie

Komunikacja pomiędzy człowiekiem a komputerem od początku odbywała się w głównej mierze poprzez wizualny interfejs. Wraz z rozwojem technologii interfejs stawał się coraz bogatszy, a autorzy systemów zaczęli dysponować szerokim wachlarzem form multimedialnych. Wydajność dzisiejszych urządzeń umożliwia wykorzystywanie chociażby takich technik, jak wirtualna, czy poszerzona rzeczywistość, które stwarzają dodatkowe możliwości dla projektantów i twórców systemów interaktywnych. Również znaczący postęp dokonał się w zakresie sterowania komputerem, gdzie oprócz tradycyjnych kontrolerów, takich jak klawiatura czy mysz, w powszechnym użyciu znalazły się ekrany dotykowe, sterowanie głosem, gestami, czy nawet całym ciałem.

Wydawać by się mogło, że całe to bogactwo technik i narzędzi, którymi dysponują współcześni twórcy oprogramowania wymusza stosowanie równie nowoczesnych technik i zasad projektowania. Okazuje się jednak, że metody opracowane oraz stosowane kilka, czy kilkanaście lat temu są nadal aktualne i doskonale sprawdzają się również dzisiaj. Wynika to z podejścia zorientowanego na użytkownika, a nie na technologię. Wydaje się nawet, że współcześni projektanci jeszcze bardziej skupiają się na problemach użytkowników, dopasowując systemy do dedykowanych, mniejszych grup odbiorców. Przykładowo można spotkać rozwiązania nie tylko przeznaczone dla osób niepełnosprawnych, ale również specjalnie dedykowane dla dzieci czy osób starszych. W szerszym ujęciu podejście zorientowane na człowieka widać w projektach informatycznych stających przed wyzwaniami związanymi z powszechną globalizacją. Zasady projektowe dotyczące lokalizowania aplikacji na potrzeby różnych języków są rozbudowywane o kwestie typowo ludzkie, takie jak zasady kulturowe, religijne, historyczne, czy językowe, często pomijające fizyczne granice państw.

Zorientowanie na użytkownika, wspólne projektowanie, prototypowanie, testy i mierzenie wydajności interfejsu oraz podejście iteracyjne nadal pozostają najlepszymi metodami projektowania systemów interaktywnych.

Literatura

- [BeHo98] Beyer H., Holtzblatt K.: *Contextual Design: Defining Customer-centered Systems*. Academic Press, San Diego 1998.
- [Boot89] Booth P.A.: *An Introduction to Human-computer Interaction*. Psychology Press, Harlow 1989.

- [DeWR08] Dennis A., Wixom B.H., Roth R.M.: *System Analysis and Design*. Wiley, Hoboken 2008.
- [DFAB04] Dix A., Finlay J., Abowd G.D., Beale R.: *Human-computer Interaction*. Pearson Education, Harlow 2004.
- [Easo76] Eason K.D.: *Understanding the Naive Computer User*. „The Computer Journal” 1976, Vol. 19(1).
- [GoLe85] Gould J., Lewis C.: *Designing for Usability: Key Principles and What Designers Think*. „Communications of the ACM” 1985, No. 28(3).
- [JoSa10] Joshi A., Sarda N.: *Evaluating Relative Contributions of Various HCI Activities to Usability*. W materiałach konferencyjnych Human-Centred Software Engineering: Third International Conference, HCSE 2010.
- [Micr95] *Microsoft Corporation: The Windows Guidelines for Software Designers*. Microsoft Press, Redmond WA 1995.
- [Niel93] Nielsen J.: *Usability Engineering*. Academic Press, Boston 1993.
- [Norm88] Norman D.: *The Psychology of Everyday Things*. Basic Books, New York 1988.
- [SaJB08] Satzinger J.W., Jackson R.B., Burd S.D.: *Systems Analysis and Design in a Changing World*. Cengage Learning EMEA, 2008.
- [ShCR09] Shelly G.B., Cashman T.J., Rosenblatt H.S.: *System Analysis and Design*. Cengage Learning, Hampshire 2009.
- [SmAt06] Smith-Atakan S.: *Human-computer Interaction*. Cengage Learning EMEA, Hampshire 2006.

Źródła internetowe

- [WWW1] <http://developer.apple.com/library/mac/#documentation/UserExperience/Conceptual/AppleHIGuidelines/XHIGIntro/XHIGIntro.html> [dostęp: 22.11.2012].
- [WWW1] <http://www.asktog.com/basics/firstPrinciples.html#humanInterfaceObjects> [dostęp: 22.11.2012].
- [WWW2] <http://www.bulkrenamentility.co.uk/screenshots.php> [dostęp: 22.11.2012].
- [WWW3] <http://www.cs.cmu.edu/~bam/uicourse/special/> [dostęp: 22.11.2012].
- [WWW5] <http://www.homepage.mac.com/bradster/iarchitect/clarity.htm> [dostęp: 22.11.2012].
- [WWW6] <http://www.lec.com> [dostęp: 22.11.2012].
- [WWW7] <http://www.lighthouse.org/accessibility/design/accessible-print-design/effective-color-contrast> [dostęp: 22.11.2012].
- [WWW8] <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2037509/> [dostęp: 22.11.2012].
- [WWW9] <http://osha.europa.eu/en/legislation/directives/provisions-on-workload-ergonomical-and-psychosocial-risks/osh-directives/5> [dostęp: 22.11.2012].
- [WWW10] <http://www.translate.google.com> [dostęp: 22.11.2012].
- [WWW11] <http://www.useit.com/alertbox/20000319.html> [dostęp: 22.11.2012].
- [WWW12] <http://www-01.ibm.com/software/ucd/ucd.html> [dostęp: 22.11.2012].
- [WWW13] <http://www-01.ibm.com/software/ucd/designconcepts/designbasics.html> [dostęp: 22.11.2012].

HUMAN-COMPUTER INTERACTION

Summary

The paper covers deliberations on methods of designing interactive systems. The first part of paper presents the issue of human-computer interaction. Then describes a user-oriented approach and design models of the interactive systems. The last two parts of the paper focuses on the design of user interface. Starting from the description of the design process, and ending with the principles and design patterns.

Radosław Kowal

ZARZĄDZANIE CYKLEM ŻYCIA APLIKACJI – NA PRZYKŁADZIE SAP SOLUTION MANAGER

Wprowadzenie

Niemal każde duże przedsiębiorstwo do usprawnienia zarządzania korzysta z wielu narzędzi informatycznych. W odniesieniu do produktów informatycznych, tak jak do wszystkich innych, można stosować koncepcję cyklu życia. Aplikacje zmieniają się stosunkowo często, ze względu na konieczność dostosowywania oprogramowania do zmiennych potrzeb użytkowników oraz z powodu konieczności eliminowania błędów. Wraz ze wzrostem stopnia skomplikowania rozwiązań informatycznych, wykorzystywanych do wspierania procesów biznesowych w przedsiębiorstwach, pojawiła się potrzeba sprawnego zarządzania używanymi rozwiązaniami, w tym w szczególności opracowanie metod i technik, pozwalających na szybkie i sprawne przeprowadzanie zmian.

Jako przykładowym rozwiązaniem tego problemu posłużono się narzędziem o nazwie Solution Manager, produkowanym przez renomowanego dostawcę oprogramowania dla biznesu – firmę SAP AG. Ze względu na duży stopień skomplikowania zagadnień związanych z implementacją i utrzymaniem rozwiązań SAP w przedsiębiorstwach, opracowano i stworzono oprogramowanie wspierające zarządzanie cyklem życia aplikacji. SAP Solution Manager realizuje wiele funkcji z zakresu zarządzania cyklem życia aplikacji, a dodatkowo jego niewątpliwą zaletą jest cena – jest dostarczany nieodpłatnie organizacjom korzystającym z rozwiązań z grupy Business Suite.

1. Koncepcja cyklu życia aplikacji – Application Lifecycle Management (ALM)

W dzisiejszych czasach duża część operacji gospodarczych odbywa się przy użyciu narzędzi informatycznych. Można powiedzieć, że praktyka biznesowa w dużej mierze opiera się na infrastrukturze informatycznej i usługach informa-

tycznych. Pojęcie usługi informatycznej obejmuje: infrastrukturę informatyczną, potrzebną do świadczenia usługi, dane i aplikacje. Infrastruktura informatyczna składa się z kolei z trzech podstawowych części: sprzętu (hardware), oprogramowania (software) oraz procesów organizacyjnych dotyczących zarządzania rozwiązaniami informatycznymi w firmie (IT processes). Biorąc pod uwagę ogromne uzależnienie współczesnej gospodarki od technologii informatycznych należy zapewnić odpowiednie ich wykorzystanie. W celu wskazania sposobów optymalnego wykorzystania zasobów informatycznych i świadczenia usług informatycznych w przedsiębiorstwach zostało opracowanych wiele standardów. Jednym z najpopularniejszych jest ITIL – IT Infrastructure Library.

ITIL to zestaw najlepszych praktyk w zakresie zarządzania infrastrukturą informatyczną i usługami informatycznymi w firmie. ITIL opisuje cykl życia aplikacji z punktu widzenia zarządzania usługami informatycznymi. ITIL koncentruje się na procesach, funkcjach organizacyjnych oraz dobrych praktykach, poprzez zastosowanie których możliwe jest lepsze, tj. bardziej wydajne i stabilne, zapewnienie usług informatycznych dla realizacji procesów biznesowych. Stanowi szczegółowy opis wielu istotnych praktyk w zakresie technologii informacyjnych, zawierający obszerne listy zadań do wykonania, procedur i obowiązków, które mogą być dostosowane do potrzeb konkretnej organizacji [Bret05]. Punktem wyjścia dla dobrych praktyk, rekomendowanych w ramach ITIL są wymagania biznesowe, określane z punktu widzenia realizacji procesów biznesowych w danej firmie. Te wymagania determinują z kolei wykorzystanie określonych usług informatycznych. Wiedząc z jakich usług informatycznych i w jaki sposób firma zamierza korzystać, można określić sposoby, działania i procesy, dzięki którym wykorzystanie usług informatycznych będzie bardziej optymalne.

2. Definicja zarządzania cyklem życia aplikacji

Chappell w swoim artykule dotyczącym zarządzania cyklem życia aplikacji prowadzi dyskurs z poglądem utożsamiającym zarządzanie cyklem życia aplikacji z cyklem życia rozwoju oprogramowania. Zwraca przy tym uwagę na wymiar biznesowy ALM, który decyduje o rozszerzeniu pojęcia cyklu życia rozwoju oprogramowania. Zgodnie z jego podejściem cykl życia aplikacji z perspektywy biznesowej wiąże się z ponoszeniem kosztów, związanych z aplikacją i rozpoczyna się od idei oprogramowania, a kończy wraz z zaprzestaniem utrzymywania oprogramowania [Chap08].

Zarządzanie cyklem życia aplikacji można zdefiniować jako: „(..) ciągły proces zarządzania życiem aplikacji poprzez kierowanie, rozwój i utrzymanie.” [Ross10]. Jako korzyści płynące z wykorzystywania praktyk definiowanych przez zarządzanie cyklem życia aplikacji Rossman wymienia:

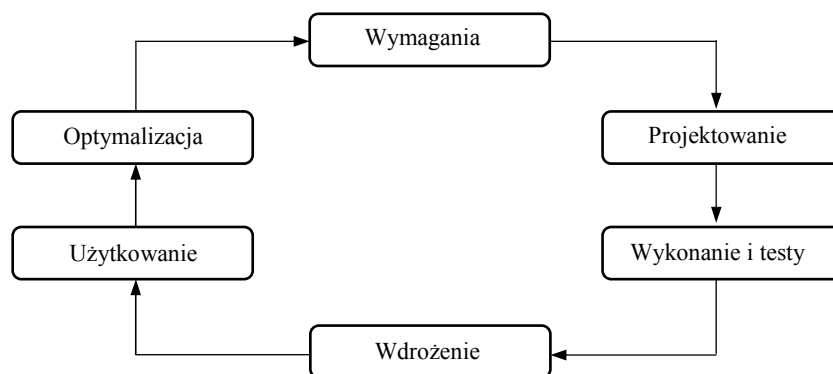
- zwiększenie produktywności, wraz z rozpowszechnieniem wśród zespołu nawyku stosowania najlepszych praktyk, związanych z tworzeniem i publikowaniem aplikacji,
- podwyższenie jakości, skutkujące lepszym dopasowaniem aplikacji do oczekiwań użytkowników,
- znoszenie barier poprzez zwiększenie otwartości przepływu informacji i współpracy,
- przyspieszenie tworzenia aplikacji poprzez uproszczenie integracji,
- zmniejszenie ilości czasu potrzebnego na utrzymanie aplikacji, wynikające z lepszej synchronizacji aplikacji i projektowania,
- maksymalizacja nakładów na podwyższenie umiejętności, jakości procesów i technologii,
- zwiększenie elastyczności poprzez redukcję czasu potrzebnego na dostarczenie nowej funkcjonalności [Ross10].

Ujmując rzecz prościej, zarządzanie cyklem życia aplikacji jest koncepcją, która zakłada zarządzanie projektem tworzenia oprogramowania przez wszystkie fazy jego cyklu życia. Wśród faz najczęściej wymienia się fazę wymagań, zmiany, konfiguracji, integracji, wydania (publikowania) i testów [HuttAg12]. Huttermann [Hutt12] uważa, że ALM to także zarządzanie relacjami między różnymi artefaktami, wykorzystywanymi do wspierania wyżej wspomnianych faz. Wśród tych artefaktów wymienia: dokumenty opisujące wymagania, artefakty kodowania oraz skrypty budowy, służące do automatyzacji procesu kompilacji i instalacji oprogramowania [HuttAg12].

Zgodnie ze wskazówkami płynącymi z ITIL cykl życia aplikacji składa się z sześciu faz (rys. 1):

1. Wymagania – gromadzenie wymagań odnośnie do nowej/zmienianej aplikacji, zgodnie ze zgłoszonym zapotrzebowaniem na realizację procesów biznesowych.
2. Projektowanie – określanie funkcji realizowanych przez aplikację w celu zaspokojenia wymagań biznesowych.
3. Wykonanie i testy – budowa aplikacji, testy i przygotowanie modelu operacyjnego dla aplikacji.
4. Wdrożenie – wprowadzenie nowego modelu operacyjnego do istniejącego środowiska IT oraz przekazanie aplikacji do użytkownika.

5. Użytkowanie aplikacji – świadczenie usług / dostarczanie produktów z wykorzystaniem wdrożonej aplikacji.
6. Optymalizacja – monitorowanie i analiza wykorzystania aplikacji, określanie co należy usprawnić.



Rys. 1. Cykl życia aplikacji

Źródło: Na podstawie: [WWW1].

Rozpatrując zarządzanie cyklem życia aplikacji z punktu widzenia zarządzania usługami informatycznymi w przedsiębiorstwach, ALM można określić jako zestaw metod, narzędzi i modelowych procesów, służących do osiągnięcia następujących celów:

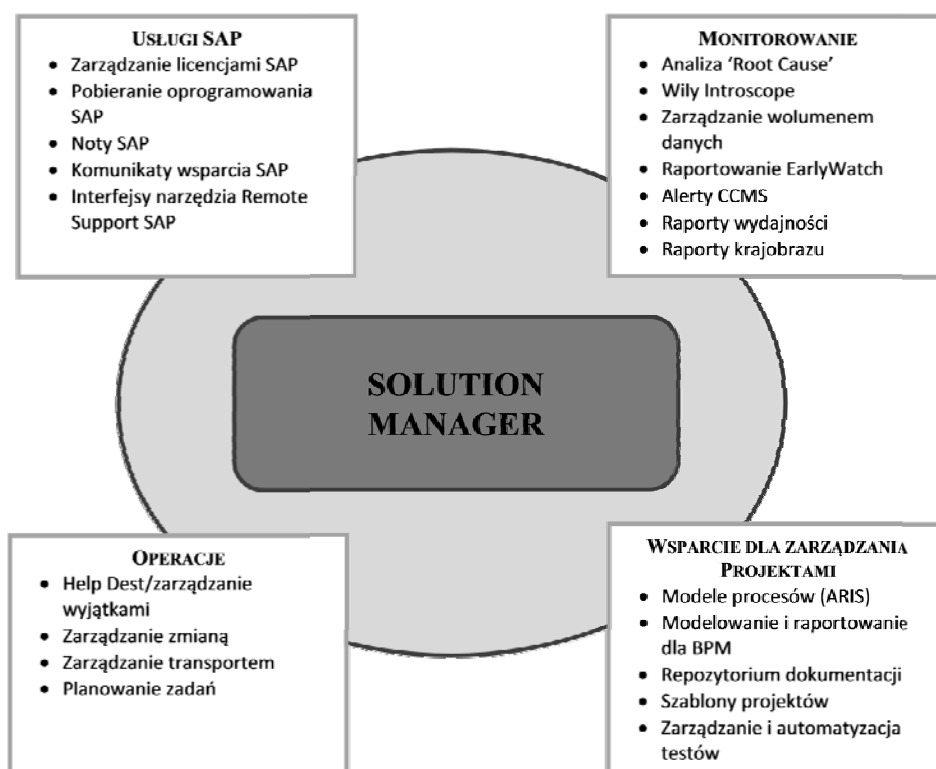
- stałej dostępności rozwiązań informatycznych,
- zarządzania zmianami w zakresie rozwiązań informatycznych – w tym zapewnienia odpowiedniego poziomu innowacyjności,
- redukcji ryzyka w obszarze rozwiązań informatycznych,
- obniżenia całkowitego kosztu posiadania zasobów informatycznych.

3. Application Lifecycle Management – SAP Solution Manager

Zarządzanie cyklem życia aplikacji, według SAP, polega na dostarczaniu standardów, najlepszych praktyk oraz rekomendacji dla organizacji, które dotyczą najbardziej typowych sytuacji, spotykanych u klientów SAP – wdrożenia, użytkowania lub podnoszenia wersji oprogramowania [Gold12].

Koncepcja ALM jest realizowana przez SAP poprzez zastosowanie rozwiązań, takich jak SAP Solution Manager i SAP Netweaver.

SAP Solution Manager (SM) to zestaw narzędzi dostarczanych przez firmę SAP dla klientów, wykorzystujących SAP Business Suite (rys. 2).



Rys. 2. SAP Solution Manager – zestaw narzędzi

Źródło: Na podstawie materiałów promocyjnych firmy SAP.

SAP Solution Manager jest zatem rozwiązaniem, które skupia w jednym miejscu narzędzia, procedury i modele wykorzystywane do zarządzania całością rozwiązań SAP w danej organizacji. Solution Manager nie tylko pozwala zarządzać, ale także monitorować, serwisować i nawet optymalizować parametry pracy poszczególnych części rozwiązania SAP. Wszystkie narzędzia i zadania, związane z zarządzaniem cyklem życia oprogramowania są zintegrowane w ramach Solution Manager. Wykorzystując Solution Manager pracownicy działu IT firmy wspomagani są przy wykonywaniu rutynowych działań, związanych z utrzymaniem oprogramowania.

Patrząc z perspektywy firmy korzystającej z rozwiązań SAP, Solution Manager stanowi odpowiedź na pytanie: w jaki sposób zapewnić poprawne wykonywanie operacji IT, krytycznych dla realizacji kluczowych procesów biznesowych?

Solution Manager umożliwia wykonywanie następujących działań w dziedzinie ALM:

1. Dokumentacja rozwiązania

Dokumentacja rozwiązania powinna pozwolić na osiągnięcie przejrzystości, podniesienie efektywności w utrzymaniu aplikacji oraz podniesienie produktywności poprzez centralizację przechowywania i zarządzania dokumentacją oraz relacjami między procesami biznesowymi.

2. Zarządzanie innowacją

Dostarcza narzędzia i metody ukierunkowane na identyfikację, adaptację oraz wdrożenie nowości i poprawek zarówno w obszarze procesów biznesowych, jak i w technicznym aspekcie działania oprogramowania.

3. Zarządzanie szablonami

Przygotowanie i zastosowanie szablonów projektów dla organizacji posiadających wiele oddziałów. Pozwala na redukcję czasu i kosztu związanego z cyklem życia oprogramowania ze względu na standaryzację podejmowanych działań.

4. Zarządzanie testem

Pozwala, na podstawie analizy wpływu zmian, na zidentyfikowanie zakresu i sposobu przeprowadzania testów. Jest używany do automatyzacji procesu testowania, zarządzania testerami oraz do raportowania przebiegu i rezultatu testów.

5. Zarządzanie zmianą

Umożliwia zarządzanie zmianami w obszarze biznesowym i technicznym w sposób przypominający zarządzanie przepływem pracy. Zarządzanie zmianą integruje zarządzanie projektem i narzędzia do synchronizacji procesu wprowadzania zmian.

6. Zarządzanie wyjątkami aplikacji

Pozwala na scentralizowane przetwarzanie pojawiających się incydentów i problemów z oprogramowaniem. Dodatkowo umożliwia przekazywanie informacji między wszystkimi zaangażowanymi stronami.

7. Operacje techniczne

Reprezentuje wszelkie aspekty funkcjonalności związane z monitoringiem, powiadomieniami, analizą i administracją wszystkimi rozwiązaniami SAP.

8. Operacje procesu biznesowego

Obejmuje wszystkie zagadnienia związane z użytkowaniem aplikacji, wpływające na efektywną realizację kluczowych procesów biznesowych.

9. Zarządzanie utrzymaniem

W zakres zarządzania utrzymaniem wchodzi identyfikacja i testowanie poprawek do utrzymywanych aplikacji. Dodatkowo zapewnia automatyczne wprowadzenie zmian w środowisku produkcyjnym.

10. Zarządzanie podnoszeniem wersji

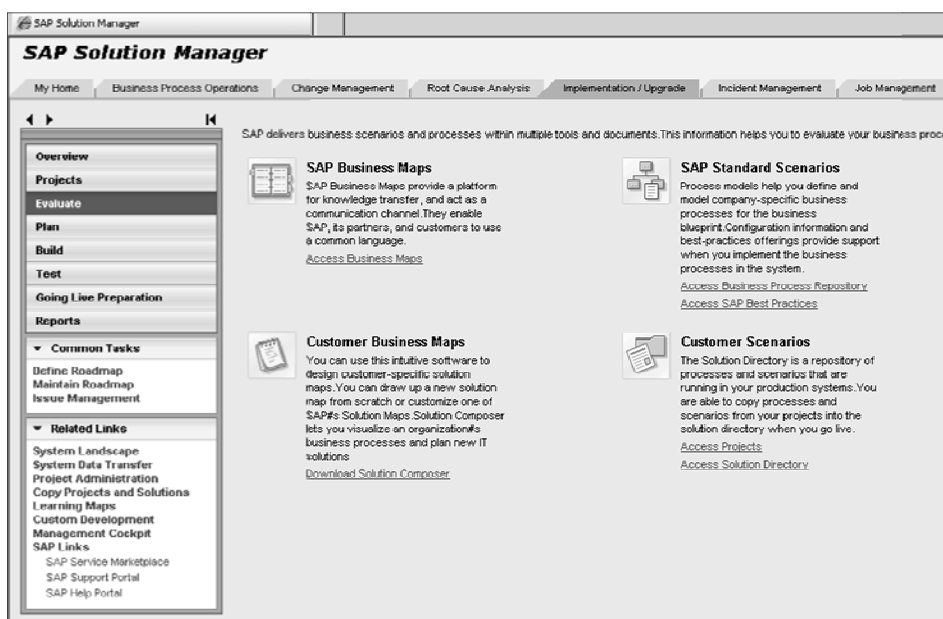
W ramach podnoszenia wersji proces jest opracowywany całościowo: od identyfikacji potrzeby, poprzez instalację nowego oprogramowania po testy całości i start w środowisku produkcyjnym.

SAP Solution Manager jest rozwiązaniem, które zawiera w sobie wszystkie niezbędne funkcje i/lub pozwala na integrację z innymi narzędziami, które są wymagane do zarządzania cyklem życia aplikacji. W skład zestawu oferowanych przez rozwiązanie funkcji wchodzi [WWW2]:

1. Work Centers
2. Implementation Content
3. Service Desk
4. Business Process Monitoring
5. Service Level Reporting
6. Job Scheduling Management
7. End-To-End Root Cause Analysis
8. KPI Measurement Platform
9. End-to-End Integration Testing
10. Maintenance Certificate
11. Maintenance Optimizer

4. Work Centers

Work Centers to narzędzia przyporządkowane do określonych ról organizacyjnych. Ich zadaniem jest zgrupowanie zadań, które są wykonywane przez osoby odpowiedzialne za realizację danego procesu. Dzięki Work Centers użytkownik otrzymuje odpowiednio dostosowane menu – rys. 3.



Rys. 3. Work Centers – SAP Solution Manager

Źródło: Na podstawie zrzutu ekranu z SAP Solution Manager.

Do każdego elementu w cyklu życia aplikacji są przypisane właściwe Work Centers. Niektóre z nich znajdują zastosowanie w wielu fazach cyklu życia aplikacji. Solution Manager jest wyposażony w następujący zestaw Work Centers:

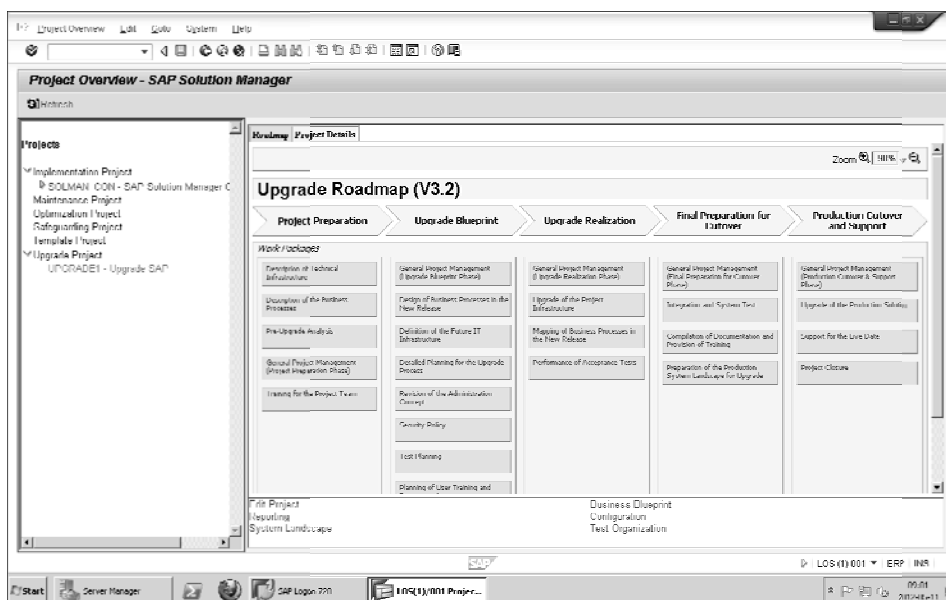
- My Home
- Business Process Operations
- Change Management
- Implementation/Upgrade
- Incident Management
- Job Management
- Root Cause Analysis
- SAP Engagement and Service Delivery
- SAP Solution Manager Administration
- Solution Documentation Assistant
- System Administration
- System Landscape Management
- System Monitoring
- Test Management

5. Implementation Content

To pojęcie opisuje zawartość repozytorium scenariuszy realizacji procesów biznesowych z różnych branż, które mogą zostać pobrane przez użytkowników SAP Solution Manager i zaimplementowane w ramach posiadanego rozwiązania. Zawartość pobierana z repozytorium obejmuje:

- informacje na temat konfiguracji,
- wytyczne, zorientowane procesowo, dotyczące tworzenia szczegółowego planu działania, konfiguracji i przygotowania systemu do pracy produktywnej.

Dzięki wykorzystaniu tych swoistych wzorców projektowych zostaje usprawniony proces implementacji i upgrade'u rozwiązań z SAP Business Suite.



Rys. 4. Projekt typu Upgrade – Roadmap

Źródło: Na podstawie zrzutu ekranu z SAP Solution Manager.

6. Service Desk

Ta część SAP Solution Manager odpowiada za funkcjonalności związane ze zgłaszaniem problemów z aplikacjami przez użytkowników końcowych, a także przez pracowników działu IT w każdej fazie projektu. W ramach procesu naprawczego Solution Manager oferuje następujące funkcjonalności:

- centralne zarządzanie zgłoszeniami serwisowymi:
 - tworzenie zgłoszeń z każdej transakcji,
 - automatyczne przechwytywanie i zapisywanie informacji na temat stanu systemu w momencie wystąpienia błędu,
 - automatyczne dopasowanie poziomu wsparcia (wewnątrz organizacji, zgłoszenie do SAP itp.) dla zgłaszanego problemu,
 - dodatkowo: wyświetlanie zgłoszeń, analiza zgłoszeń, odpowiedzi na zgłoszenia, tworzenie oraz dołączanie dokumentacji, zarządzanie statusem, tworzenie listy zadań i kolejkovanie zadań,
- wyszukiwanie i import not SAP,
- wewnętrzna baza rozwiązań,
- interfejs do SAP Support.

7. Business Process Monitoring

Głównym zadaniem monitorowania procesów biznesowych w SAP Solution Manager jest proaktywne i zorientowane procesowo monitorowanie procesów biznesowych. Pozwala na monitorowanie zarówno od strony technicznej, jak i od strony logiki aplikacji.

Celem monitorowania jest wykrycie nieprawidłowości tak szybko, jak to możliwe i usunięcie jej, tak aby usterki nie spowodowały przerwy w realizacji kluczowych procesów biznesowych. Idealnie byłoby, gdyby udało się wykryć usterkę zanim o jej istnieniu dowie się użytkownik, który odpowiada za realizację procesu biznesowego, którego ta usterka dotyczy. Dane dotyczące liczby i miejsca powstawania problemów, a także czasu potrzebnego na ich usunięcie mogą następnie być analizowane, np. z użyciem narzędzi Business Intelligence.

8. Service Level Reporting

Service Level Reporting stanowi jeden z wbudowanych mechanizmów, wykorzystywanych do raportowania. Na podstawie danych zebranych z systemów zgrupowanych w ramach jednego krajobrazu, SLR pozwala na stworzenie raportów, dostosowanych do potrzeb użytkownika. Dodatkowo raporty te mogą być skonfigurowane tak, aby możliwe było ich automatyczne uruchamianie, a ich wynik był dostarczany do wybranego użytkownika. Celem dla tego typu raportów jest zapewnienie klientom odpowiedniego (uzgodnionego wcześniej) poziomu działania usługi.

Dane prezentowane w raportach pochodzą z Early Watch Alerts (EWA). Mechanizm EWA pobiera dane z systemów znajdujących się w jednym kraju. Pobrane dane dotyczą kluczowych czynników, świadczących o powodzeniu w realizacji danego procesu biznesowego. Innymi słowy gromadzone przez EWA dane to kluczowe wskaźniki efektywności (KPI).

9. Job Scheduling Management

Job Scheduling Management oferuje użytkownikom funkcjonalności związane z tworzeniem i uruchamianiem zadań do wykonania w systemie. Zwykle ta funkcjonalność jest utożsamiana z automatyzacją tego typu prac. Jednakże poza ustalaniem, jakie zadania i kiedy mają zostać uruchomione, JSM pozwala także na:

- przeglądanie zadań uruchomionych w tle,
- poprawę efektywności wykonywania zadań i rozkładanie obciążenia,
- poprawę poziomu obsługi poprzez odpowiednie ustalenie priorytetów zadań.

10. End-To-End Root Cause Analysis

Jest to jedna z funkcjonalności, które w terminologii SAP są określane mianem operacji technicznych. Analiza głównych przyczyn pozwala na badanie przyczyn występowania problemów z oprogramowaniem w niejednorodnym środowisku. Niejednorodność oznacza tutaj środowisko, w którym występują elementy pochodzące od wielu różnych producentów oprogramowania. Operator wykonujący analizę z użyciem tego narzędzia, niezależnie od badanego systemu, będzie korzystał z tego samego interfejsu.

W ramach End-to-end Root Cause Analysis, SAP dostarcza następujące aplikacje:

1. End-to-end Workload Analysis

Analiza obciążenia może być wykonywana w odniesieniu do wybranych komponentów (np. ERP) lub dla wielu komponentów w tym samym czasie.

2. End-to-end Change Analysis

Za pomocą tzw. agentów diagnostycznych są zbierane informacje statystyczne z wybranych systemów, dotyczące parametrów pracy systemu. Możliwe jest także testowanie, jak zareaguje system, po wprowadzeniu zmian w parametrach.

3. End-to-end Exception Analysis

Pozwala na generowanie statystyk występowania błędów w aplikacjach, a także na analizę dla wybranego komponentu.

4. End-to-end Trace Analysis

Polega na śledzeniu działań w systemie/-ach wybranego użytkownika, a następnie na możliwości analizy zapisu tychże działań w SAP Solution Manager.

11. KPI Measurement Platform

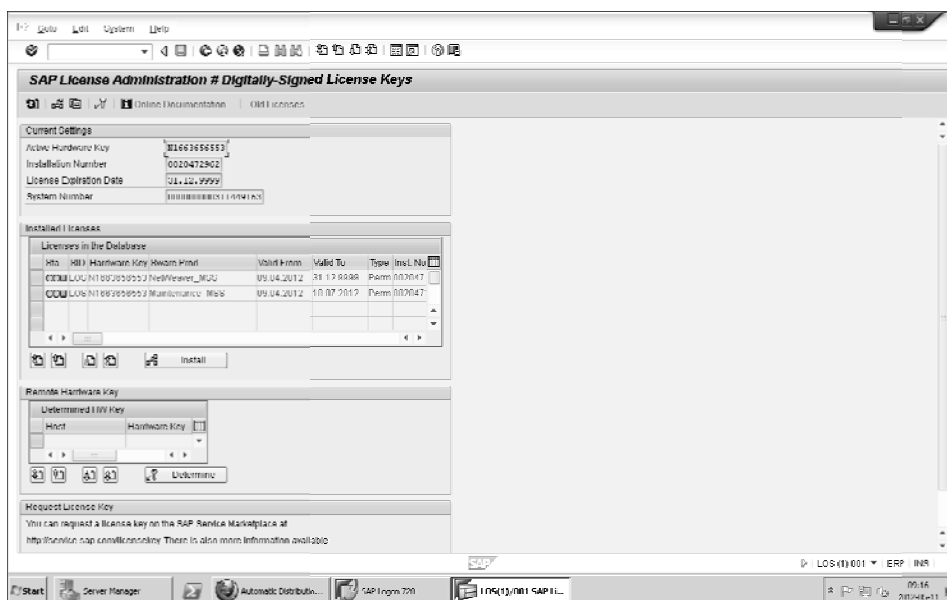
W przypadku realizacji umów serwisowych klient często wymaga dowodów na skuteczność zastosowanych rozwiązań, szczególnie w sytuacji, kiedy rozwiązanie dotyczy usprawnienia pracy aplikacji. Funkcjonalność KPI Measurement Platform odpowiada za przedstawienie jasnego obrazu sytuacji przed zastosowaniem rozwiązania i po, w celu analizy osiągniętych rezultatów. Tak jak inne aplikacje, dostępne w ramach SAP Solution Manager, tak też ta opiera się na koncepcji kluczowych procesów biznesowych. Mierzone są zatem kluczowe wskaźniki efektywności w odniesieniu do realizowanych procesów biznesowych.

12. End-to-end Integration Testing

Wiele firm często decyduje się na dokonywanie zmian w użytkowanych aplikacjach. Niestety, jeżeli aplikacje, których zmiany dotyczą, pełnią ważną rolę w prowadzeniu standardowej działalności biznesowej danej firmy, to po wprowadzeniu zmian należy gruntownie przetestować zdolność aplikacji do realizacji procesów biznesowych. Testowanie jest skomplikowanym zagadnieniem nawet w jednorodnym środowisku, dlatego w ramach SAP Solution Manager jest dostarczane rozwiązanie E2E Integration Testing, służące do zarządzania procesem testowania. Proces testowania obejmuje całokształt zagadnień związanych z przystosowaniem aplikacji do wprowadzanych zmian, począwszy od przygotowania założeń i celów, przez planowanie i przeprowadzenie testów aż do wprowadzenia przetestowanych zmian do systemu produkcyjnego.

13. Maintenance Certificate

Każde z rozwiązań wchodzących w skład SAP Business Suite wymaga odpowiedniej licencji. Klucze licencyjne są generowane po stronie SAP na podstawie tzw. klucza sprzętowego, pobieranego bezpośrednio z maszyny, na której jest zainstalowane oprogramowanie SAP. Aby usprawnić pracę działu IT w środowisku rozproszonym, gdzie wykorzystywanych jest wiele rozwiązań SAP, można skorzystać z funkcjonalności SAP Solution Manager – Maintenance Certificate. Dzięki tej funkcjonalności jest możliwe zarządzanie kluczami licencyjnymi w jednym miejscu (Solution Manager) zamiast na każdej z maszyn. Solution Manager pełni w takim układzie rolę dystrybutora licencji dla rozwiązań SAP.



Rys. 5. Zarządzanie licencjami – SAP Solution Manager

Źródło: Na podstawie zrzutu ekranu z SAP Solution Manager.

14. Maintenance Optimizer

Maintenance Optimizer wspiera zarządzanie skomplikowanym krajobrazem rozwiązań SAP. Pozwala na zarządzanie wszelkimi działaniami związanymi z utrzymaniem oprogramowania SAP. Dzięki tej funkcjonalności można pobierać i instalować w wybranych systemach pakiety aktualizacji dla oprogramo-

wania SAP. Maintenance Optimizer bada bieżącą konfigurację i automatycznie wskazuje, jakie pakiety aktualizacji powinny być zastosowane, pobiera je i uruchamia odpowiednie narzędzie, służące do instalacji aktualizacji. Możliwe jest aktualizowanie oprogramowania SAP zarówno opartego na ABAP, jaki i JAVA – Support Packages, Support Package stacks, supports Upgrades. Maintenance Optimizer pozwala również na instalację rozszerzeń do obecnie wykorzystywanego oprogramowania, takich jak Enhancement Packages, Legal Change Packages, a także rozwiązań typu add-ons.

Podsumowanie

SAP Solution Manager stanowi ucieleśnienie koncepcji Application Lifecycle Management. Jest to doskonały przykład oprogramowania, które pomaga departamentowi IT w zarządzaniu całym pakietem rozwiązań w jednolity sposób i całościowo, tj. w odniesieniu do różnych potrzeb w ramach cyklu życia aplikacji. Główne zalety tego rozwiązania to:

- zarządzanie cyklem życia aplikacji z punktu widzenia realizacji kluczowych procesów biznesowych,
- integracja zarządzaniem cyklem życia wszystkich aplikacji obecnych w przedsiębiorstwie, także tych, których producentem nie jest SAP AG,
- niski koszt użytkowania – SAP dostarcza Solution Manager bez dodatkowych opłat dla posiadaczy licencji na rozwiązanie z rodziny SAP Business Suite.

Literatura

- [Bret05] Office of Government Commerce. Introduction to ITIL Book. TSO, London 2005.
- [Chap08] Chappell D.: *What is Application Lifecycle Management?*, <http://www.microsoft.com/global/applicationplatform/en/us/RenderingAssets/Whitepapers/What%20is%20Application%20Lifecycle%20Management.pdf> [dostęp: 22.04.2012].
- [Gold12] Magen-Goldstein T.: *Introduction – Application Lifecycle Management*, <http://scn.sap.com/docs/DOC-23405> [dostęp: 29.04.2012].
- [Hutt12] Huttermann M.: *Agile ALM: Lightweight Tools and Agile Strategies*. Manning Publications, Shelter Island 2012.
- [Ross10] Rossman B.: *Application Lifecycle Management – Activities, Methodologies, Disciplines, Tools, Benefits*. Alm Tools and Products, Lightning Source Incorporated 2010.

[WWW1] [<http://www.ibm.com/developerworks/tivoli/library/t-appmgtlife/index.html>]
[dostęp: 22.07.2011].

[WWW2] <https://websmp210.sap-ag.de/alm-tools> [dostęp: 22.07.2011].

APPLICATION LIFECYCLE MANAGEMENT IN SAP SOLUTION MANAGER

Summary

There is a growing complexity in information technologies used in today's organizations. In order to facilitate business operations organizations use a lot IT tools, from relatively simple and less meaningful, like word processors, to very complicated and critical for business, like ERP systems. Managing such a huge and turbulent environment is challenging. To do this effectively organizations need a coherent and salient standard, describing the rules for managing IT landscape. ITIL is one of the most popular standards. Here we concentrate on a part of ITIL called Application Lifecycle Management and focus on its implementation by SAP AG in the Solution Manager product.

Artur Machura

ANALIZA BIZNESOWA W PRZEDSIĘWZIĘCIACH IT

Wprowadzenie

Artykuł opisuje praktyczne zastosowanie cenionych technik wspierających warsztat metodyczno-narzędziowy Analityków Biznesowych i Systemowych. Metodyka RUP (Rational Unified Process), która stanowi pewien szablon pracy [KrKru06], wyznaczyła jej określone obszary, na podstawie których zostały zdefiniowane wnioski artykułu. W kolejnych punktach artykułu znajdują się: opis prowadzonych zajęć, uzyskane wyniki z prowadzonych zajęć, dyskusja wyników oraz podsumowanie.

1. Opis prowadzonych zajęć

Wykonywanie obowiązków Analityka Biznesowego w opisywanym przedsięwzięciu IT było związane z następującymi obszarami prac:

- udziałem w spotkaniach z klientem, których celem było głównie omawianie wymagań zawartych w SIWZ (Specyfikacji Istotnych Warunków Zamówienia [WWW3]), pozyskiwanie nowych wymagań, doprecyzowanie wymagań. Zasadniczym rezultatem pracy były notatki ze spotkań;
- realizowaniem prac analitycznych i projektowych na podstawie pozyskanych od klienta informacji. Na uwagę zasługują przede wszystkim zadania dotyczące zarządzania wymaganiami oraz modele opisujące funkcjonalność projektowanego systemu;
- przekazywaniem rezultatów pracy do pozostałych członków zespołu. Uzgodnieniu rezultatów pracy towarzyszyło wykorzystanie przeróżnych środków komunikacji, takich jak: zwykłych rozmów, wymiany tradycyjnych biurowych dokumentów oraz modeli analitycznych;
- weryfikowaniem i konsultowaniem pracy wykonanej przez zespół wytwórczy zarówno z samymi członkami zespołu, jak i klientem. Wyróżnia się tu udział w testowaniu aplikacji, wspieraniu zespołu i klienta przez cały czas realizacji projektu [KrKru06].

2. Uzyskane wyniki z prowadzonych zajęć.

Opisywane przez artykuł wyniki pracy badawczej są związane ze zrealizowanym innym celem pracy, jakim było wdrożenie systemu w środowisku produkcyjnym klienta. Sumując uzyskane wyniki z prowadzonych prac analitycznych w opisywanym przedsięwzięciu, można zauważyć zarówno cechy pozytywne, jak i negatywne dla przedmiotu wiedzy Inżynierii Oprogramowania.

Do zalet zalicza się:

Zdefiniowanie celów biznesowych

Prezentowany system podczas odbioru poszczególnych iteracji i przyrostów oraz końcowego produktu był nieustannie konfrontowany z osiągniętymi za jego sprawą celami biznesowymi przyszłego użytkownika systemu, co pozwoliło zespołowi wytwórczemu na bieżąco poznawać oczekiwania klienta [WrMaWy06].

Specyfikację przypadków użycia

Opis funkcjonalny na podstawie zdefiniowanych celów użycia systemu / przypadków użycia stanowił bardzo istotny produkt pracy [BoRaJa02]. Umożliwił powiązanie szczegółowych oczekiwań użytkowników systemu dotyczących interakcji z systemem z wyobrażeniem zespołu wykonawcy na temat projektowanego systemu.

Do wad zalicza się:

Ewolucyjny i iteracyjno-przyrostowy proces wytwarzania oprogramowania.

Pomimo wielu zalet swojej idei, opartej na ewolucyjnym budowaniu rozwiązania, zastosowanie procesu wiązało się z dużym wyzwaniem analityczno-projektowym. Organizacja projektu na podstawie takiego procesu uniemożliwiła zdobycie dokładnej wiedzy o całym rozwiązaniu – nawet w czasie wdrażania pierwszego modułu na środowisku produkcyjnym. Ponadto w opisywanym projekcie można było zauważyć:

- trudności w rozbijaniu systemu na niezależne fragmenty,
- problemy z integracją fragmentów systemu,
- dodatkowe koszty tworzenia atrap brakujących fragmentów,
- mały stopień dokumentacji prac projektowo-implementacyjnych,
- podejście minimalistyczne do dokumentowania prac własnych przez poszczególnych przedstawicieli wykonawcy utrudniało bieżącą komunikację i ostatecznie zmniejszało produktywność zespołu. Brak właściwej dokumentacji wymagał bezpośredniego kontaktowania się wielu członków zespołu, często kosztem innej pracy przewidzianej w harmonogramie,
- tendencja zespołu do pracy w sposób „zwinny”. Doświadczenie poszczególnych członków zespołu nastawiała zespół do pracy i rozwiązywania problemów w sposób „zwinny”, co z jednej strony umożliwiało szybkie rozwiązy-

wanie bieżących problemów na poszczególnych stanowiskach pracy. Z drugiej natomiast było początkiem poważniejszych problemów. W projekcie rozciągniętym na przestrzeni kilku lat, gdzie skład zespołu był dynamicznie zmienny, tylko stosowanie metod formalnych mogłoby zapewnić standardy Inżynierii Oprogramowania, a zarazem pożądaną jakość budowanego rozwiązania [WWW1].

3. Dyskusja wyników

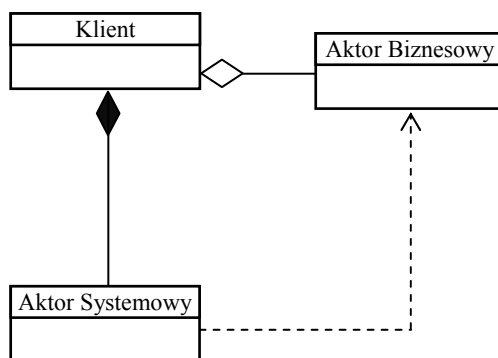
Udział w przedsięwzięciu doprowadził do wniosków, które będą rozpatrywane w następujących kontekstach:

- czy podstawowe problemy projektów, leżące u podstaw metodyki RUP, zostały zniwelowane przez jej zastosowanie?
- wpływ przedsięwzięć na prace analityczno-projektowe,
- rzeczywisty zakres obowiązków wynikających z objętej roli analityka w przedsięwzięciu,
- prace wykonywane przez analityka a inżynieria oprogramowania.

Twórcy RUP (Ken Hartman, Ivar Jacobson, Philippe Kruchten) definiując tzw. zbiór najlepszych praktyk, mieli na uwadze problemy ówczesnych projektów, wśród których wyróżnia się te związane bezpośrednio z analizą, a więc: zarządzanie wymaganiami ad hoc; niejednoznaczna, nieprecyzyjna komunikacja; niewykryte niespójności w wymaganiach, projekcie oraz implementacji [WWW2]. Uzyskane wyniki i scharakteryzowane problemy opisywanego przedsięwzięcia skłaniają do następujących wniosków:

Kryteria oceny przedsięwzięć mają wpływ na warsztat metodyczno-narzędziowy. Większość przedsięwzięć informatycznych jest realizowana w pewnym reżimie czasowym i założonym budżecie, co najczęściej implikuje wykonywanie tylko takiej ilości pracy, jaka jest potrzebna do stworzenia zamówionego produktu. Podczas tej pracy, przyjęte kryteria oceny wyznaczają zwykle górną granicę dla jakości pracy (a więc i Inżynierii Oprogramowania). Rozróżnia się grupę kryteriów inżynierskich (zgodność z przewidzianym harmonogramem i budżetem, zgodność z założoną specyfikacją) oraz wielu ekonomicznych (związanych zasadniczo z oceną poniesionych kosztów i korzyści z tego płynących dla inwestora) [Le07]. Istotny jest tu fakt, że nie istnieją precyzyjne mechanizmy łączące zależność kryteriów oceny przedsięwzięć ze stosowanymi metodami Inżynierii Oprogramowania, co ma zasadniczy wpływ na podejście do zastosowania konkretnych procesów wytwórczych oraz metod i narzędzi.

Organizacja projektu przekłada się na podejście i metody stosowane podczas prowadzenia analizy. U podstaw analizy dokonywanej w przedsięwzięciach stoją wymagania klienta. Jak się okazuje, organizacja projektu może również wpłynąć na sposób pozyskiwania tych wymagań. Z praktyki na stanowisku Analityka wynika, że zamawiający często nie ma wystarczającej wiedzy biznesowej aby precyzyjnie definiować wymagania. Poniższy rysunek przedstawia istotną zależność pewnych pojęć, które jak wynika z dalszej interpretacji mają istotny wpływ na opisywany tu problem.



Rys. 1. Związek pomiędzy aktorem biznesowym, systemowym i klientem

Rysunek przedstawia związek klienta, aktora biznesowego i aktora systemowego. Uwypukla przy tym istotne zależności, z których wynika, że aktor biznesowy może istnieć bez klienta, a sam aktor systemowy jest od niego zależny. Dla analityka pozyskującego wymagania ma to zasadnicze znaczenie. Wymagania zgłaszane przez użytkownika, który jednocześnie nie ma wystarczającej wiedzy biznesowej na temat organizacji narażają istotnie powodzenie projektu. Taka sytuacja może implikować konieczność uruchomienia innego projektu poświęconego wprost analizie przedsiębiorstwa i jego procesów biznesowych.

Podsumowanie

Praca, która jest wykonywana na etapie analizy powinna się cechować nie tylko dbałością o kwestie bezpośrednio związane z samym produktem – systemem informatycznym. Okazuje się bowiem, że produktami wytwarzanymi na tym etapie należy umiejętnie zarządzać w ramach przedsięwzięć informatycznych. Rezultaty i produkty analityczne powinny wiązać dwa równoległe i skomplikowane środowiska:

- organizację klienta – przyszłego użytkownika systemu,
- zespół twórczy wykonawcy i wynikające z tego właściwości.

Pogodzenie powyższych środowisk, które oddziałują na warsztat metodyczno-analityczny Analityka powinno mieć na uwadze cel nadrzędny, jakim jest kierunek ewolucji organizacji klienta i wynikające z tego oczekiwania biznesowe. Produkty analityczne z jednej strony definiują konkretne i teraźniejsze byty, natomiast z drugiej powinny cechować się elastycznością i pozwolić na obsługę przyszłych zmian. Praca Analityka Biznesowego powinna cechować się przewidywalnością nie tylko bezpośrednich skutków wynikających z jej wykonania. Należy także zwrócić szczególną uwagę na konsekwencje tej pracy w powiązaniu z ewolucją organizacji i jej biznesem. Kompetencje osób podejmujących decyzje w tym zakresie często są budowane przez wiele lat, bowiem zgłębienie tej wiedzy nie jest rzeczą trywialną i jest czasochłonne. Na pewno szkolenia i kompetencje potwierdzone międzynarodowymi certyfikatami ułatwiają podejmowanie właściwych decyzji. Dopiero jednak zdefiniowanie odpowiedzialności i zakresu obowiązków Analityka w odniesieniu do konkretnego środowiska pracy zapewnią właściwą realizację pracy.

Doświadczenia autora związane z pełnieniem obowiązków analityka biznesowego w przedsięwzięciach informatycznych uzmysłowiły, że istnieje potrzeba stworzenia definicji, w której wyznaczono by proporcje oraz pogodzone by pojęcia z dwóch oddzielnych obszarów wiedzy: inżynierii oprogramowania oraz ekonomii. Dzięki temu powstałyby uzasadnione punkty odniesienia i wytyczne dla zespołów wytwórczych, które informowałyby, jaki wpływ na przebieg przedsięwzięcia i inżynierię oprogramowania mają – wydaje się najistotniejsze rezultaty pracy w każdym przedsięwzięciu IT – produkty analityczne.

Literatura

- [BoRaJa02] Booch G., Rambaugh J., Jacobson I.: *UML Przewodnik użytkownika*. Wydawnictwo WNT, Warszawa 2002.
- [KrKru06] Kroll P., Kruchten P.: *Rational Unified Process od strony teoretycznej*. Wydawnictwo WNT, Warszawa 2006.
- [Le07] Lech P.: *Metodyka ekonomicznej oceny przedsięwzięć informatycznych wspomagających zarządzanie organizacją*. Wydawnictwo Uniwersytetu Gdańskiego, Gdańsk 2007.
- [WrMarWy06] Wrycza S., Marcinkowski B., Wyrzykowski K.: *Język UML 2.0 w modelowaniu systemów informatycznych*. Wydawnictwo Helion, Gliwice 2006.
- [WWW1] <http://agilemanifesto.org/iso/pl/principles.html>
- [WWW2] http://pl.wikipedia.org/wiki/Rational_Unified_Process
- [WWW3] <http://prawo.legeo.pl/prawo/ustawa-z-dnia-29-stycznia-2004-r-prawo-zamowien-publicznych/>

BUSINESS ANALYSIS FOR IT PROJECTS

Summary

Work that is performed at the stage of the analysis should not only be characterized by addressing the issues directly related to the same product – a computer system. It turns out that the products produced at that stage should be skillfully managed within IT projects. Results and analytical products should merge two parallel and complex environments:

- needs of the organization – customer's, the future user of the system,
- the manufacturing team of the contractor and the resulting properties.

Reconciling of above factors affecting the work, probably should have an overriding goal in mind, where we speak of business needs across the organization – the customer, the direction of its evolution. Analytical products on the one hand define specific and present beings, on the other, should have flexibility and allow support for future changes. This work should be characterized not only predictable effects directly affecting the fate of the implemented projects, but also the future ones related to the evolution of the organization and its business needs, where methodological – tool workshop of the analyst is extremely important.

The competence of decision makers in this area is often built over many years. Indeed, exploration of this knowledge is not trivial and takes time. Certainly the training and competence, confirmed by international certificates helps to make the right decisions. But it was the defining of role the of the analyst in a real and specific workplace – provides appropriate reference point , that is needed in the implementation of relevant work for the client and contractor.

The experiences of a person working as analyst in IT projects, have led to the conclusions outlined in this article, which made it clear that later it would be needed to carry out scientific work, aimed to provide a definition. It would be a definition that would accept the proportions and agreed ideas from two separate areas of expertise: software engineering and economics. Thus, there would be some legitimate points of reference and guidelines for the production teams, which would inform about the impact of analytical products – that seem to be the most important results of the work in any IT project – on the course of the project and software engineering.

Joanna Palonka

ZARZĄDZANIE UMOWAMI Z WYKONAWCAMI SYSTEMÓW IT

Wprowadzenie

Zarządzanie kontraktami (umowami) stanowi dziś istotny aspekt funkcjonowania organizacji i to nie tylko w obszarze wykorzystania systemów IT, ale we wszystkich obszarach działalności, np. w sprzedaży, logistyce itd. Obejmuje ono zarówno zarządzanie cyklem życia kontraktu począwszy od identyfikacji potrzeb, specyfikacji wymagań, poprzez negocjowanie warunków oraz zapewnienie zgodności realizacji z zasadami i warunkami, jak również dokumentację i uzgodnienia w sprawie wszelkich zmian, które mogą powstać w czasie realizacji lub wykonania umowy, a w końcowej fazie monitorowanie i zamknięcie umowy. Jest to proces systematycznego i efektywnego tworzenia umów, ich wykonania i prowadzenia analiz w celu maksymalizacji wyników finansowych i operacyjnych oraz zminimalizowania ryzyka w podejmowanym projekcie. W efekcie powstaje pisemna lub ustna prawnie wiążąca umowa między stronami, zobowiązująca do spełnienia warunków w niej określonych.

Celem artykułu jest przedstawienie zagadnienia specyfikacji projektów informatycznych w kontekście przygotowania umowy na system IT.

1. Potrzeby informacyjne użytkowników systemów IT

Pojęcie potrzeby jest używane w odniesieniu do wszystkich systemów charakteryzujących się celowym zachowaniem i zdolnością samoorganizacji, tj. takich, które cechują się zdolnością uczenia się i wobec tego także optymalizacji zachowania, czyli rozwoju [KolNow99, s. 40]. W związku z tym do najważniejszych potrzeb każdego systemu zalicza się potrzeby informacyjne. Ich istotą jest to, że informacje są wartościami, które zaspokajają wymagania systemu, które są określane za ich pomocą. Potrzeba informacyjna jest wyrażeniem pewnej wartości, pożądanej ze względu na przeszłe, bieżące lub przewidywane zachowanie systemu. Sposób opisu tej wartości, jej strukturalizacji zależy od przyjętych za-

łożeń teoretyczno-metodologicznych. Między potrzebą informacyjną a zachowaniem systemu mogą zachodzić różne relacje w zależności od poziomu w hierarchii potrzeb [KolNow99, s. 40].

Poszczególne potrzeby informacyjne lub ich kolekcje mogą być wykorzystane do identyfikacji cech rozważanego systemu w kategoriach systemowych, tzn. zachowania, struktury i rozwoju. Potrzeba informacyjna jest źródłem poznania, jest konsekwencją celowo dokonanego wyboru sposobu (algorytmu) rozwiązania pewnego problemu, pozwala na poznanie kryteriów wyboru, a na tej podstawie preferencji w zakresie przedmiotu zainteresowania.

W odniesieniu do systemu informatycznego (SI) potrzeba informacyjna pełni funkcje systemotwórcze. W procesach modelowania struktur SI uwzględnia się ich funkcje (postulowane formy i zasady funkcjonowania) stosownie do zakresu zidentyfikowanych potrzeb. Składniki strukturalne potrzeb informacyjnych stają się bezpośrednio lub pośrednio składnikami struktury informacyjnej SI.

Znajomość potrzeb ma istotne znaczenie w realizacji SI. Istnienie potrzeby jest konsekwencją pojawienia się konieczności rozwiązania problemu. Zaspokojenie potrzeby stanowi warunek rozwiązania problemu lub osiągnięcia celu. Wynika stąd, że rozeznanie potrzeby jest warunkiem a priori przejścia do dalszych etapów procesów realizacji SI i głównym zadaniem organizacji, dla której ma być realizowany system.

Identyfikacja potrzeb jest sekwencją czynności badawczo-analitycznych mających na celu określenie zbioru wymagań dotyczących sposobu zaspokojenia potrzeby oraz dalszych etapów procesu działania. Przed tym jednak musi nastąpić uszczegółowienie potrzeby wcześniej rozpoznanej lub uświadomienie potrzeby dotychczas nieznannej. Identyfikacja potrzeb może następować w drodze obserwacji zarówno podsystemów sfery realnej, jak podsystemów sfery regulacji, w tym również funkcjonujących już składników SI. Podstawowymi metodami badania w tej fazie realizacji są metody analizy systemowej oraz metody badania pracy. Wykonawcami prac analityczno-badawczych są specjaliści z zakresu organizacji procesów sfery realnej (przyszli użytkownicy, kadra zarządzająca, klienci) oraz analitycy i projektanci SI.

2. Specyfikacja wymagań dla systemu IT

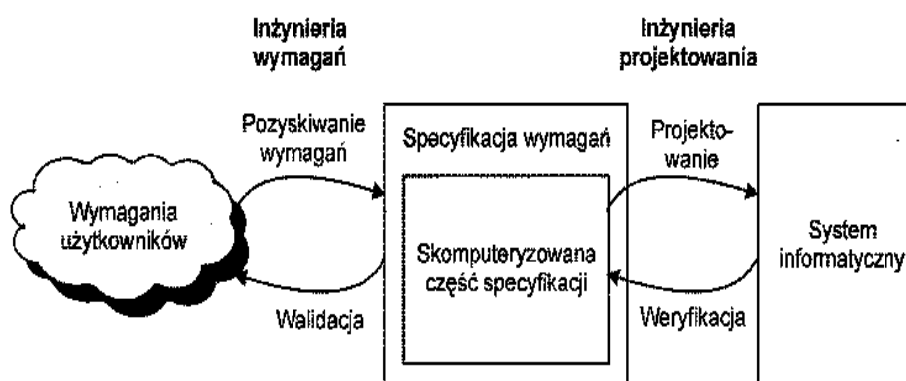
Identyfikacja potrzeb stanowi punkt wyjścia do określenia wymagań użytkowników względem tworzonego SI. Wymagania użytkowników są definiowane jako zbiór właściwości, atrybutów, usług, funkcji i/lub cech charakterystycznych produktu niezbędnych do zaspokojenia celów systemu. Mogą one mieć charakter:

- podstawowy (tzw. nadrzędne, główne, bezpośrednie) – charakteryzują potrzeby użytkownika końcowego względem systemu,
- pochodny (tzw. wtórne, pośrednie) – wynikają z konieczności zrealizowania innych wymagań,
- funkcjonalny – określają zasady działania systemu, dotyczą rezultatów oczekiwanych przez użytkowników systemu,
- нефункциональный – odnoszą się zazwyczaj do całości systemu i opisują takie atrybuty tego systemu, jak: dostępność, niezawodność, odtwarzalność, rozszerzalność, kompatybilność i przenośność [DąbStaWol07, s. 56; Jasz97, s. 48; KoIoISzy05, s. 137].

Proces określania wymagań jest iteracyjnym procesem analizy problemu, dokumentowania rezultatów w różnych formatach, prezentacji oraz kontroli dokładności uzyskanego zrozumienia problemu. Określany jest mianem inżynierii wymagań [KoIoISzy05, s. 137].

Kluczowym aspektem inżynierii wymagań jest komunikacja między członkami organizacji z uwzględnieniem ich punktów widzenia. W efekcie zostaje stworzony sformalizowany model informacyjny organizacji – specyfikacja wymagań SI – będący dokumentacją wymagań użytkowników względem przyszłego systemu. Wymagania te lub ich część zostają zrealizowane w postaci SI.

Na rys. 1 przedstawiono miejsce inżynierii wymagań na tle procesu projektowania SI.



Rys. 1. Miejsce inżynierii wymagań na tle procesu projektowania SI

Źródło: [KoIoISzy05, s. 136].

Dobrze zbudowana specyfikacja wymagań powinna charakteryzować się:

- jednoznacznością – wszystkie wymagania są jednoznaczne,
- kompletnością – zawiera pełny zbiór dokładnie opisanych wymagań,
- poprawnością – każdy rodzaj wymagania został przeanalizowany i potwierdzony przez klienta,

- spójnością – między wymaganiami nie występują konflikty,
- weryfikowalnością – można sprawdzić, czy każdy rodzaj wymagania spełnia parametry systemu,
- modyfikowalnością – możliwe są łatwe, kompletne i spójne zmiany w wymaganiach,
- śladowością – wymagania można „śledzić” (kontrolując powiązania z innymi wymaganiami i dokumentami) – [KolOlSzy05, s. 137].

Tylko taka specyfikacja wymagań stanowi solidną podstawę konstrukcji umowy.

Od jakości specyfikacji wymagań użytkowników zależy często powodzenie całego procesu budowy SI. Jest to jednak zadanie trudne w realizacji, czasochłonne i przez to często ignorowane przez zarządy organizacji, które szybko chcą uzyskać efekty pracy informatyków w postaci działającego systemu. Takie podejście pociąga za sobą drastyczny wzrost kosztów rozwiązywania problemów, które ujawniają się w późniejszych fazach cyklu życia SI. Spowodowane jest to tym, że każde zadanie nieobjęte umową jest traktowane przez dostawcę jako nowe.

3. Aspekty umów na systemy IT

Specyfikacja projektów informatycznych w kontekście przygotowania umowy na system IT powinna obejmować wymagania:

- biznesowe,
- technologiczne,
- dostosowawcze,
- wdrożeniowe

oraz kwestie zarządzania projektem IT, a w szczególności wskazywać osoby odpowiedzialne za poszczególne działania i efekty [GarKon10; Jasz97, s. 48; Wasz10].

Wymagania biznesowe

Skuteczność opisu wymagań można uzyskać realizując kolejno następujące etapy:

1. Identyfikacja oczekiwań Zamawiającego system IT.
2. Przygotowanie specyfikacji funkcjonalnej systemu IT.
3. Przeprowadzenie stadium wykonalności systemu IT.

W etapie 1 Zamawiający opisuje, jak ma działać system po wdrożeniu w organizacji, określa wpływ na systemy pracy i funkcjonowanie organizacji. Dzięki temu zostają wyspecyfikowane cele stworzenia i wdrożenia systemu. Ponadto wstępnie są ustalane efekty poprzez wskazanie kryteriów pomiaru sukcesu dla wdrożenia systemu. Oczekiwania Zamawiającego mogą być zebrane w drodze wywiadu z bezpośrednimi użytkownikami i ze specjalistami, poznawania istniejących systemów w organizacji, obserwacji działań w rzeczywistości.

Prace w etapie 2 polegają na zapisie w języku naturalnym, w formie pisemnej, oczekiwań Zamawiającego dotyczących:

- celów, które mają być osiągnięte po wdrożeniu systemu,
- opisu działania systemu,
- funkcjonalności systemu,
- użyteczności,
- struktury danych w systemie,
- koncepcji nawigacji w systemie,
- integracji i interoperacyjności systemu IT z innymi systemami.

Przeprowadzenie stadium wykonalności systemu IT (etap 3) obejmuje:

- specyfikację funkcjonalną systemu,
- opis sposobu realizacji funkcjonalności systemu,
- graficzną wizualizację wszystkich kluczowych ekranów GUI,
- opis struktury bazy danych,
- specyfikację głównych modułów tworzących system,
- opis technologii wykonania i uruchomienia systemu, w tym języki programowania, bazę danych, środowisko techniczne wymagane do prawidłowego uruchomienia systemu i wymagania sprzętowe,
- opis przeniesienia/migracji danych z innych SI,
- opis integracji i interoperacyjności z innymi SI,
- wymagania/założenia eksploatacyjne,
- metodykę wdrożenia,
- założenia Project Management (punkty krytyczne, zasoby, harmonogram projektu, nakłady, ryzyka, kryteria sukcesu).

Do przeprowadzenia studium wykonalności może posłużyć poniższy schemat postępowania, dostosowany do specyfiki konkretnej sytuacji:

1. Charakterystyka projektu zawierająca wskaźniki monitorowania projektu oraz analizę instytucjonalną odnoszącą się do trwałości i wykonalności instytucjonalnej projektu.
2. Różne analizy specyficzne dla danego rodzaju projektu/sektora.
3. Analiza finansowa.
4. Analiza ekonomiczna.
5. Analiza oddziaływania na środowisko [Sach10, s. 62].

Wymagania względem systemu IT powinny być rozpatrywane w organizacji z dwóch punktów widzenia – samego wdrożenia systemu i obsługi powdrożeniowej systemu.

Analiza działania organizacji przed wdrożeniem systemu IT powinna obejmować istniejące procesy, procedury, dane (dokumenty i informacje), techniki i technologie, workflow (punkty krytyczne) i czynnik ludzki.

Projekt działania organizacji po wdrożeniu systemu IT powinien wskazać zmiany w procesach, procedurach i workflow, zmiany w technice i technologiach, migrację danych i zarządzanie informacją, przygotowanie kadry (np. szkolenia) i zarządzanie zmianą.

Opracowując wymagania dotyczące obsługi powdrożeniowej systemu IT, trzeba wiedzieć, że nie ma gwarancji na system IT w prawie polskim. Umowa stanowi podstawę roszczeń. Trzeba opracowywać SLA (Service Level Agreement) dotyczącą zakresu i trybu świadczenia usług. Ponadto w umowie powinny znaleźć się ustalenia na temat warunków świadczenia usług gwarancyjnych, usług pogwarancyjnych, opieki autorskiej, asysty technicznej, szkoleń, konsultacji, pomocy bieżącej, modyfikacji i perspektyw dalszego rozwoju systemu IT.

Wymagania technologiczne

Definiując wymagania technologiczne, trzeba mieć przede wszystkim na uwadze potrzeby Zamawiającego. Cechami, na które należy zwrócić szczególną uwagę są niezawodność i nowoczesność. W przypadku pierwszej cechy krytyczne znaczenie ma czas i skuteczność rozwiązania, natomiast w przypadku drugiej – efekty biznesowe (poprawa wskaźników) i wizerunkowe.

Jeśli Zamawiający chce rozwiązanie nowoczesne, to musi mieć świadomość, że rozwiązania nowe/nowoczesne są z reguły niesprawdzone, mogą zawierać błędy, mogą być niestabilne, kosztowne w zakupie i wdrożeniu, a także pewne jest, że dotychczas niewiele (lub wcale) było takich wdrożeń. Rozwiązania niezawodne są natomiast opisywane jako takie, których wdrożenia zakończyły się już sukcesem w innych organizacjach i takich wdrożeń było wiele. Dostawcy/wykonawcy tych rozwiązań mają już doświadczenie w tym zakresie. Referencje są dowodem jakości tych rozwiązań.

Czasem jednak rozwiązania niezawodne są uznawane za przestarzałe.

Wymagania dostosowawcze

Podjmując decyzję o wdrożeniu systemu IT w organizacji zawsze pojawia się problem dostosowania dwóch elementów systemu IT i potrzeb Zamawiającego. Definiując wymagania, trzeba przeanalizować dwie sytuacje: dostosowanie IT do potrzeb biznesowych i potrzeb biznesowych do IT.

W sytuacji dostosowania IT do potrzeb trzeba rozważyć następujące kwestie: czy opracowywany projekt będzie miał charakter indywidualnego – dedykowanego oprogramowania oraz jakie jest ryzyko wdrożenia takiego rozwiązania? Z reguły indywidualne rozwiązania są niesprawdzone, wysokie jest ryzyko zmian w trakcie realizacji i wdrożenia projektu IT, wysokie są koszty i długi jest czas jego wytworzenia, wyższe są koszty obsługi powdrożeniowej oraz wysokie jest ryzyko niezadowolenia Zamawiającego [GarKon10].

W sytuacji dostosowania potrzeb do IT również są do przeanalizowania dwie kwestie. Trzeba odpowiedzieć na pytanie: czy istnieje system IT spełniający wymagania w możliwie największym stopniu oraz czy będą wymagane zmiany w organizacji, procesach i procedurach, workflow, zarządzaniu informacją po stronie Zamawiającego? Ważne jest również oszacowanie ryzyka takiego wdrożenia, bo podejście to wiąże się w praktyce z częstym niedopasowaniem rozwiązania do początkowych oczekiwań Zamawiającego, początkowym rozczarowaniem Zamawiającego i dużym oporem czynnika ludzkiego w fazie wdrożenia systemu IT.

Wymagania wdrożeniowe

Wymagania wdrożeniowe są związane z tym, jak będzie mierzony sukces. Wskazuje się następujące kryteria sukcesu:

- utrzymanie projektu IT w zakładanym budżecie – kierownik projektu musi posiadać podstawową wiedzę ekonomiczną,
- realizacja projektu IT w planowanym czasie, tj. wyliczanie czasu realizacji projektu, harmonogramowanie prac, śledzenie postępu prac i reagowanie na odchylenia,
- spełnienie planowanych kryteriów jakościowych, które muszą być zdefiniowane na etapie założeń, muszą być mierzalne (SMART Project Management), mogą odnosić się do wybranych parametrów pracy systemu a parametry muszą pozwalać na pomiar i porównanie działania systemu IT z założeniami [WWW1].

Do punktów krytycznych realizacji i wdrożenia systemu IT zalicza się:

- plan projektu (alokację zasobów, analizę ścieżki krytycznej),
- Komitet Sterujący (zaangażowanie decydentów, uprawnienia decyzyjne, tryb podejmowania decyzji),
- etapowanie projektu IT (modelowanie, prototypowanie, testy Beta),
- weryfikację realizacji projektu IT (metodykę, procesy i procedury),
- odbiory częściowe,
- zarządzanie zmianą [Pal06, s. 51].

Zarządzanie projektem

W zarządzaniu projektem IT najważniejszą kwestią jest ustalenie zakresu odpowiedzialności poszczególnych członków zespołu projektowego:

- Zamawiający jest odpowiedzialny za realizację umowy w zakresie swoich obowiązków,
- Wykonawca jest odpowiedzialny za realizację umowy w zakresie swoich obowiązków,
- Project Manager odpowiada tylko za przygotowanie planu projektu i realizację projektu zgodnie z planem projektu,
- Komitet Sterujący weryfikuje zgodność realizacji projektu z planem projektu i zarządza zmianami w ramach warunków umowy,
- Zespół Decyzyjny (Zamawiający i Wykonawca łącznie) uzgadniają zmiany w trakcie realizacji projektu wykraczające poza warunki umowy [Pal06, s. 51].

4. Specyfikacja do umowy

Przygotowując specyfikację do umowy, trzeba uwzględnić zidentyfikowane potrzeby Zamawiającego. Proces identyfikacji potrzeb przebiega według następujących kroków:

1. Analiza potrzeb po stronie Zamawiającego.
2. Wykorzystanie ekspertów/konsultantów zewnętrznych w procesie definiowania potrzeb.
3. Specyfikacja (dokumentacja) potrzeb.
4. Rozpoznanie potrzeb Zamawiającego przez Wykonawcę.

Analiza potrzeb informacyjnych

Analiza potrzeb rozpoczyna się od inwentaryzacji stanu przed wdrożeniem systemu IT. Dokumentuje się cele, potrzeby, oczekiwania, wyobrażenia i plany Zamawiającego. Działania te są oparte na opisach własnych Zamawiającego, wywiadach i analizach rozwiązań, które podobają się/odpowiadają Zamawiającemu. Następnie grupuje się, wartościuje i priorytetyzuje zdefiniowane wymagania Zamawiającego.

Wiedza o potrzebach informacyjnych użytkownika powinna obejmować:

- opis informatyzowanego obszaru organizacji, czyli dokumentów źródłowych, kartotek i informacji wynikowych (zestawień, wydruków, raportów), a także algorytmów, umożliwiających otrzymanie informacji wynikowych z danych źródłowych,
- analizę opisanego obszaru organizacji, czyli dokumentów źródłowych, kartotek i informacji wynikowych, z punktu widzenia przydatności i użyteczności dla użytkownika oraz realizowanego systemu [Zie03, s. 255].

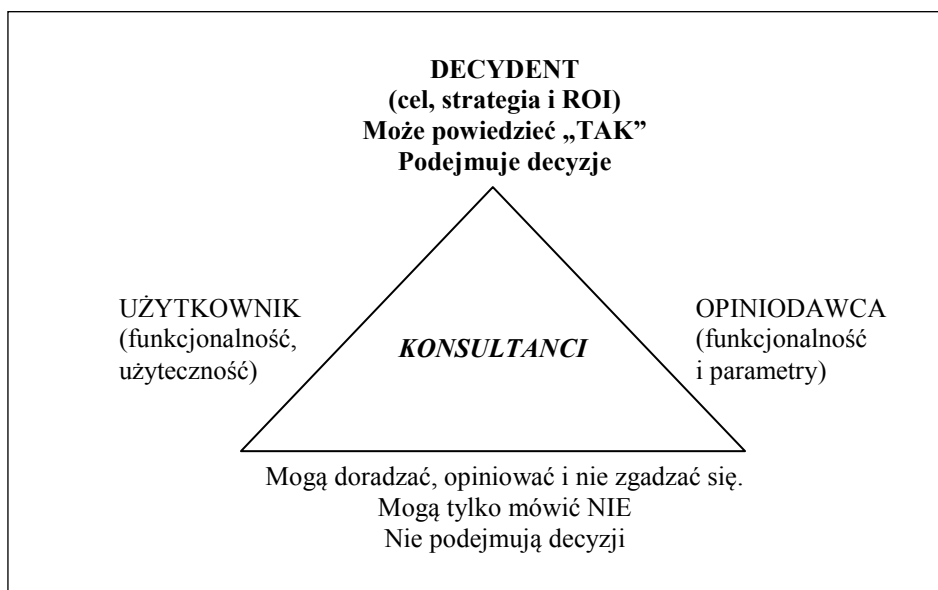
W następnej kolejności odbywają się wizyty referencyjne, a później rozpoczyna się proces tworzenia systemu. Polega on na modelowaniu rozwiązań spełniających ważne wymagania Zamawiającego, prototypowaniu rozwiązań spełniających wymagania Zamawiającego (może być kilka firm, które podejmą się opracowania prototypu; wybrana zostanie jedna firma – pozostałe odrzucone (freelancerzy) otrzymują tzw. opłatę za wysiłek) oraz wdrożeniu pilotowych rozwiązań spełniających wymagania Zamawiającego.

Na etapie analizy wymagań trzeba uwzględnić możliwość wprowadzania zmian odnośnie do wymagań wobec systemu IT. Trzeba zdefiniować obszary możliwych zmian w specyfikacji biznesowej, warunki wprowadzenia zmian w specyfikacji systemowej projektu IT, określić zasady zarządzania zmianami w trakcie realizacji projektu. W celu zminimalizowania zmian trzeba określić sposób realizacji zmian i częstotliwość odbiorów prac – cząstkowe czy przejściowe.

W praktyce przyjmuje się, że 10% zmian wprowadzonych do projektu nie powoduje konieczności zmian w umowie.

Eksperci / konsultanci zewnętrzni

W skład zespołu decyzyjnego w projekcie IT wchodzi: decydent, użytkownik, opiniodawca. Jednocześnie mogą oni pełnić rolę konsultantów. Na rys. 2 przedstawiono zakres kompetencji konsultantów w projekcie IT.



Rys. 2. Konsultanci w projekcie IT

Źródło: Na podstawie: [GarKon10].

Decydent jest konsultantem z doświadczeniem w zarządzaniu, użytkownik jest konsultantem z doświadczeniem w użytkowaniu, a opiniodawca to ekspert w dziedzinie technologii, procesów pracy, optymalizacji, funkcjonalności itd. Ważne jest, aby każdy konsultant posiadał wiedzę specyficzną dla obszaru, w którym doradza. Musi mieć świadomość, że jego praca jest kontrolowana, bowiem są stawiane cele dla jego pracy i określone miary sukcesu, czyli są definiowane kryteria ocen rozwiązań zaproponowanych przez niego.

Specyfikacja potrzeb

Szczegółowy opis przedmiotu zamówienia zawiera:

- specyfikację funkcjonalną systemu IT, obejmującą cele, opis działania, opis funkcjonalności, wymagania odnośnie do użyteczności i nawigacji, priorytety/kryteria wyboru,
- opis przeniesienia/migracji danych z innych SI,
- opis integracji i współdziałania (interoperacyjności) z innymi SI (jeżeli dotyczy),
- wymagania/założenia eksploatacyjne [DąbStaWol07, s. 68].

W trakcie specyfikacji przedmiotu zamówienia mogą pojawić się problemy w zakresie specyfikacji funkcjonalnej systemu, związane z brakiem celu dla systemu IT, niejasnym, niekonsekwentnym, sprzecznym lub przesadnie technicznym opisem działania systemu, brakiem lub opisem w zbyt lakoniczny sposób pozostałych charakterystyk przedmiotu zamówienia.

Formułując kluczowe warunki umowy, trzeba pamiętać o tym co chcemy osiągnąć i co jest dla nas ważne. W sytuacji gdy pojawiają się roszczenia względem Wykonawcy systemu IT, na liście stanowiącej podstawę dla prawnika muszą znaleźć się:

- przedmiot umowy – załącznik 1,
- obowiązki Wykonawcy,
- obowiązki Zamawiającego – załącznik 2,
- terminy realizacji,
- akceptacja i odbiór prac,
- wynagrodzenie,
- kary umowne,
- warunki płatności
- gwarancje i serwis – załącznik 3,
- licencje i prawa autorskie – załącznik 4,
- ograniczenie odpowiedzialności,
- klauzule poufności – załącznik 5,
- obowiązywanie umowy i wypowiedzenie.

W załącznikach 1-5 przedstawiono propozycję fragmentów Umowy z wykonawcą systemów IT na System X.

5. Negocjacje

Elementem nieodłącznym umów są negocjacje. Istnieje kilka tzw. grzechów głównych, które powodują porażki w negocjacjach. Są to:

- nieprzygotowanie (według praktyków 80% czasu powinno się przeznaczyć na przygotowanie do negocjacji, a 20% czasu powinny trwać negocjacje),
- własne emocje (przekonanie o słuszności swojego stanowiska, złość lub irytacja, brak wiary w sukces, strach przed porażką),
- nastawienie od początku na walkę lub poszukiwanie kompromisu [Bł94, s. 61].

Sukces negocjacji tkwi we właściwym określeniu celów negocjacji, czyli tego jaki ma być efekt negocjacji. Pomocne mogą być odpowiedzi na następujące pytania:

- które kwestie będą przedmiotem negocjacji?
- jaki jest cel negocjacji?
- jaki efekt końcowy chcemy osiągnąć?
- jakie są 3 główne priorytety w tej negocjacji?
- co stanowi wyjściową ofertę w tej negocjacji?

Trzeba pamiętać, że do negocjacji trzeba mieć alternatywę. W umowach IT sprawdza się BATNA (Best Alternative To a Negotiated Agreement) – najlepsza alternatywa dla negocjowanego porozumienia. BATNA to najlepszy sposób postępowania, który zabezpiecza interesy strony negocjującej bez porozumienia z drugą stroną. BATNA jest podstawą siły negocjacyjnej i narzędziem oceny potencjalnego porozumienia. Jeśli BATNA jest bardzo dobra, może nie być potrzeby negocjacji, gdyż porozumienie może nie dać niczego lepszego niż dostępna alternatywa. BATNA można wzmacniać przez poszukiwanie informacji o alternatywach (np. innych ofertach). Należy też analizować możliwe alternatywy dostępne dla oponenta – wykonawcy.

Etapy przygotowań BATNA:

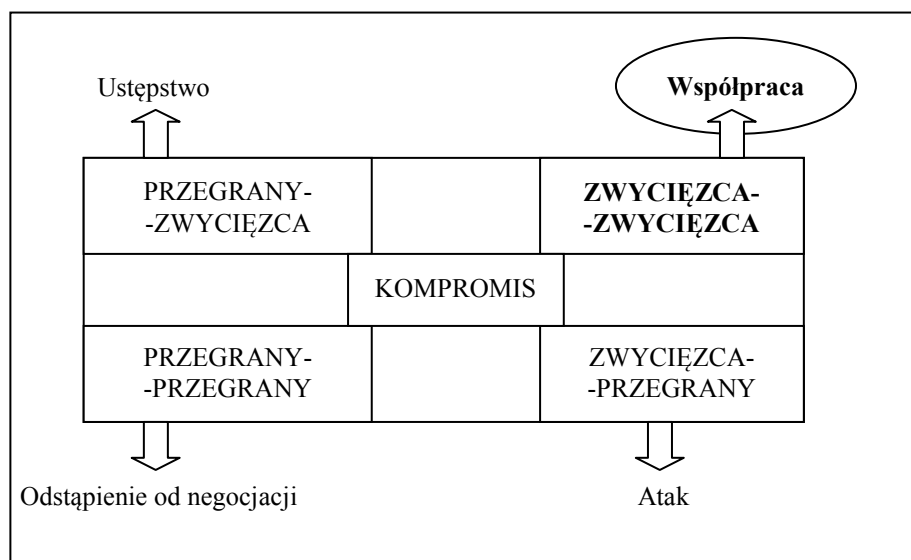
- wymyślenie wszystkich możliwych działań, jakie będzie można podjąć, jeśli porozumienie nie zostanie osiągnięte,
- udoskonalenie kilku najbardziej obiecujących pomysłów i przekształcenie ich w praktyczne możliwości działania,
- doskonalenie tymczasowego wyboru jednej z możliwości – tej, która wydaje się najlepsza [FishUrPat00].

Im lepsza BATNA, tym większa siła negocjującego. W rzeczywistości relatywna siła stron zależy przede wszystkim od tego, na ile atrakcyjne jest dla każdej z nich nieosiągnięcie porozumienia. W negocjacjach bardzo często podkreśla się znaczenie i umiejętność mówienia NIE. Stosowanie i realne postrzeganie BATNA w procesie negocjacji prowadzi do zmniejszenia obciążenia tą presją.

W trakcie negocjacji dostawca/wykonawca musi stać się partnerem – dopiero wtedy odpowiedzialność rozkłada się na obie strony. Istnieje kilka stylów negocjacji [WW2]. Na rys. 3 przedstawiono optymalny styl negocjacji dla umów na systemy IT.

Kolejnym czynnikiem wpływającym na sukces w negocjacjach umów IT jest łączne oddziaływanie 3 składowych komunikacji bezpośredniej. Proporcje rozkładają się następująco:

- słowa (treść) – 7%,
- głos – 38%,
- język ciała – 55% [Lew95, s. 112].



Rys. 3. Style negocjacji – na czym komu zależy?

W negocjacjach można natknąć się na pułapki zarówno po stronie Zamawiającego, jak i Wykonawcy (tab. 1).

W sytuacji gdy projekt „pada” możliwe są dwie ścieżki postępowania:

1. Opcja siłowa – wywarcie presji na drugą stronę (w takiej sytuacji wygrywa Kancelaria Prawna),
2. Renegocjacja warunków umowy (prowadzi ją zewnętrzny negocjator).

Z reguły jeśli dojdzie do renegocjacji, ścieżka zewnętrznego negocjatora wygląda następująco:

1. Poznać i zrozumieć problem drugiej strony – zobaczyć sytuację oczami drugiej strony.
2. Poszukać kilku wariantów możliwych rozwiązań.
3. Wybrać rozwiązanie, które jest DOBRE dla obu stron.

Tabela 1

Pułapki w negocjacjach

Zamawiający	Wykonawca
1	2
<ul style="list-style-type: none"> – wyobraża sobie nowy system IT, – robi to po raz pierwszy, – zakłada, że dostawca powinien wiedzieć jak SI ma działać i znać się na tym oraz przewidzieć konsekwencje, bowiem nasze 	<ul style="list-style-type: none"> – chce (za wszelką cenę) otrzymać zamówienie i jest skłonny zgodzić się na ryzykowne lub nierealne warunki, np.: zbyt krótki czas, zbyt mały budżet, zbyt niskie kompetencje itp.,

cd. tabeli 1

1	2
<p>oczekiwania są przecież oczywiste,</p> <ul style="list-style-type: none"> – system ma być taki, jak go sobie wyobraziliśmy, – dostawca zrealizuje projekt IT i wdrożenie zgodnie z oczekiwaniami Zamawiającego, nawet jeżeli te oczekiwania będą się zmieniać w trakcie realizacji projektu 	<ul style="list-style-type: none"> – nie ujawnia ryzyk wynikających z oczekiwań i wymagań Zamawiającego, obawiając się utraty zamówienia, – dąży do zawarcia umowy z nieświadomym Zamawiającym na warunkach „zgodności ze specyfikacją”, – świadomie akceptuje ryzykowne lub nierealne warunki stawiane przez Zamawiającego, gdyż zakłada, że Zamawiający nie wycofa się w trakcie realizacji projektu i zgodzi się na zwiększenie budżetu lub obniżkę jakości

Źródło: Na podstawie: [GarKon10; Jasz97, s. 60].

Podsumowanie

W sytuacji gdy organizacja podejmuje decyzje o wdrożeniu systemu IT pojawia się pytanie jakiego wykonawcę wybrać? Klienci/Zamawiający często w pierwszej kolejności poszukują dostawcy, który może pracować na miejscu. Tymczasem nie za każdą technologią dostępną na polskim rynku stoi lokalny dostawca. W wykonawcy/dostawcy klienci najchętniej widzieliby doradcę, a nie tylko firmę, która przywiezie oprogramowanie, skonfiguruje je i uruchomi. Zatem od doradcy jest oczekiwana – oprócz wiedzy technologicznej – wiedza biznesowa oraz chęć i gotowość do współpracy.

Dziś cena przestała odgrywać pierwszoplanową rolę. Dla klientów liczą się referencje, doświadczenie i kompetencje wykonawcy. Dla wykonawcy/dostawcy najistotniejsza powinna być pełna współpraca z klientem na każdym etapie, w szczególności w fazie analitycznej projektu, aby powstał system zgodny z oczekiwaniami. Faza analityczna pokazuje możliwe efekty, oszczędności, skalę automatyzacji. Daje też odpowiedź, czy warto wdrożenie realizować, dlatego tak ważne jest zidentyfikowanie wymagań i umiejętne skonstruowanie zapisów w umowie na system IT. Warto zadbać, aby w umowie znalazły się również zapisy o cenach licencji, wynagrodzeniu za pracę, warunkach uaktualnień itd. W negocjacjach warto stosować zasadę fair play.

Dostawca musi zadbać o satysfakcję klienta, aby ten do niego wrócił. Musi sobie zdawać sprawę, że dziś sam system IT nie jest już wartością dodaną, że stanowi standard, na którym opiera się każde działanie. Na wdrożeniu korzysta Zamawiający system, korzysta też wykonawca/dostawca. W sytuacjach spor-

nych traci przeważnie Zamawiający system IT, bo podpisał np. niekompletną umowę, dlatego tak istotna jest znajomość zagadnień z zakresu zarządzania umowami na systemy IT. Zamawiający powinien mieć zawsze na uwadze zasadę: „Żądaj wiele, otrzymasz też wiele”.

Literatura

- [Bł94] Błaut R.: *Skuteczne negocjacje*. CIM, Warszawa 1994.
- [DąbStaWol07] Dąbrowski W., Stasiak A., Wolski M.: *Modelowanie systemów informatycznych w języku UML 2.1 w praktyce*. Mikom, Warszawa 2007.
- [FishUrPat00] Fisher R., Ury W., Patron B.: *Dochodząc do TAK. Negocjowanie bez poddawania się*. PWE, Warszawa 2000.
- [GarKon10] Garstecki B., Konarski X.: *Biznesowe i prawne aspekty umów na usługi (systemy) IT, CPI*. Warszawa 2010, materiały szkoleniowe.
- [Jasz97] Jaskiewicz A.: *Inżynieria oprogramowania*. Helion, Gliwice 1997.
- [KolNow99] Kolbusz E., Nowakowski A.: *Informatyka w zarządzaniu*. Wydawnictwo ZSB, Szczecin 1999.
- [KolOlSzy05] Kolbusz E., Olejniczak W., Szyjewski Z.: *Inżynieria systemów informatycznych w e-gospodarce*. PWE, Warszawa 2005.
- [Lew95] Lewandowska-Tarasiuk E.: *Komunikowanie w biznesie. Jak skutecznie rozmawiać o interesach*. Etidions Spotkania, Gliwice 1995.
- [Pal06] Palonka J.: *Zarządzanie projektem w budowie systemów e-biznesu*. W: *Strategie i metodyka przekształcania organizacji w kierunku e-biznesu na podstawie technologii informacyjnej*. Red. H. Sroka. Wydawnictwo AE, Katowice 2006.
- [Sach10] Sacha K.: *Inżynieria oprogramowania*. Wydawnictwo Naukowe PWN, Warszawa 2010.
- [Wasz10] Waszczuk P.: *Dobre planowanie chroni przed porażką*. „Computerworld” 2010, nr18/19.
- [Zie03] Ziemia E.: *Projektowanie systemów informatycznych*. W: *Informatyka w zarządzaniu*. Red. C. Olszak, H. Sroka. Wydawnictwo AE, Katowice 2003.
- [WWW1] www.brighthub.com/office/project-management/articles/26374.aspx [styczeń 2011].
- [WWW2] www.eioba.pl/a/2ovy/style-negocjowania [grudzień 2011].

Załączniki

Załącznik 1 – Przedmiot Umowy

Przedmiotem zawartej pomiędzy Stronami Umowy jest:

1. Udzielenie KLIENTOWI licencji na użytkowanie Systemu X obejmującego moduły wymienione w Załączniku nr 1 do niniejszej Umowy, o zakresie funkcjonalnym określonym w Załączniku nr 2 do niniejszej Umowy.
2. Wdrożenie i przekazanie do eksploatacji u KLIENTA kompletnego, w pełni funkcjonalnego Systemu X zgodnie z ustalonym harmonogramem ramowym wdrożenia (Załącznik nr 3 niniejszej Umowy).
3. Przeszkolenie przez DOSTAWCĘ użytkowników i administratorów ze strony KLIENTA w zakresie Systemu X, zgodnie z harmonogramem ramowym wdrożenia – specyfikacja dot. szkoleń (Załącznik nr 3 niniejszej Umowy).
4. Świadczenie przez DOSTAWCĘ usługi serwisu gwarancyjnego zgodnie z postanowieniami niniejszej Umowy.

Załącznik 2 – Obowiązki Klienta

KLIENT jest zobowiązany do:

1. Przygotowania środowiska systemowego do instalacji modułów Systemu X zgodnie ze specyfikacją środowiska systemowego (Załącznik nr 4 do niniejszej Umowy).
2. Udostępnienia DOSTAWCY, po wcześniejszym pisemnym uzgodnieniu z KLIENTEM, pomieszczeń w siedzibie lub w miejscu wskazanym przez KLIENTA oraz umożliwienia dostępu do sprzętu komputerowego przeznaczanego do obsługi Systemu X w zakresie niezbędnym dla należytego wykonania Umowy.
3. Współpracy w weryfikacji danych po migracji danych oraz po przekazaniu przez DOSTAWCĘ do eksploatacji Systemu X.
4. Współdziałania z DOSTAWCĄ przy przygotowaniu szkoleń, powołania i zapewnienia czynnego udziału zespołów pracowników podczas szkoleń.
5. Zainstalowania i uruchomienia niezbędnego dodatkowego oprogramowania umożliwiającego sprawną pracę Systemu X w miejscach eksploatacji Systemu X, dla odpowiedniej liczby użytkowników.

Załącznik 3 – Serwis

SERWIS w okresie gwarancji

1. DOSTAWCA udziela 20-miesięcznej gwarancji na System X. Gwarancja obowiązuje od daty zakończenia instalacji System X.
2. Zgłoszenie serwisowe może być realizowane za pośrednictwem faksu lub e-maila. Zgłoszenie serwisowe uznaje się za dokonane z chwilą potwierdzenia przez DOSTAWCĘ przyjęcia zgłoszenia, co następuje z chwilą wpłynięcia do KLIENTA faksu lub e-maila zawierającego potwierdzoną przez DOSTAWCĘ kopię Zgłoszenia serwisowego.
3. Po dokonaniu Naprawy, DOSTAWCA przeprowadzi testy sprawdzające poprawność działania Systemu X.
4. W przypadku gdy KLIENT dokona niezasadnego Zgłoszenia serwisowego lub zgłoszona nieprawidłowość nie będzie objęta serwisem gwarancyjnym, KLIENT zobowiązany będzie do zapłaty na rzecz DOSTAWCY wynagrodzenia zgodnego z aktualnym cennikiem DOSTAWCY za każdą godzinę pracy pracownika DOSTAWCY.

Załącznik 4 – Licencje

LICENCJA

1. DOSTAWCA oświadcza, że jest właścicielem wszelkich autorskich praw majątkowych do Systemu X, w odniesieniu do wszystkich modułów wymienionych w Załączniku nr 1 oraz posiada niczym nie ograniczone prawo do udzielenia KLIENTOWI licencji na jego wykorzystanie.
2. DOSTAWCA udziela KLIENTOWI odpłatnej, odwoławczej, niewyłącznej, nieograniczonej w czasie, nieprzenoszalnej, ważnej na terytorium Polski licencji na używanie Systemu X w wersji wyspecyfikowanej w Załączniku nr 1, spełniającego funkcje określone w Załączniku nr 2 oraz na wykorzystywanie dokumentacji Systemu X.
3. Licencja udzielona w ramach niniejszej Umowy upoważnia KLIENTA do wykorzystania Systemu X na cele i użytek KLIENTA, na następujących polach eksploatacji:
 - użytkowanie zgodnie z dokumentacją i przeznaczeniem, w szczególności w zakresie: zbierania danych, wprowadzania danych, utrwalania danych, przechowywania danych, opracowywania danych itd.,
 - szkolenie własnych pracowników.

4. KLIENT nie może wynajmować, wdzierżawiać, udostępniać bezpłatnie Systemu X ani też wykorzystywać do szkolenia osób trzecich innych niż pracownicy KLIENTA.
5. KLIENT posiada prawo do korzystania z dokumentacji wyłącznie jako pomocy przy posługiwaniu się Systemem X w sposób określony w Umowie.
6. DOSTAWCA dostarczy KLIENTOWI certyfikat licencyjny na System X w dniu podpisania Umowy.

Załącznik 5 – Klauzule poufności

OBOWIĄZEK zachowania tajemnicy

1. DOSTAWCA zobowiązuje się do zapewnienia poufności informacji dotyczących KLIENTA pozyskanych przez DOSTAWCĘ, jego pracowników i zleceniobiorców w związku z wykonywaniem Umowy.

MANAGEMENT OF CONTRACTS WITH INFORMATION SYSTEMS PROVIDERS

Summary

The aim of this article is to present issues regarding to IT projects specifications that must be taken into account during cutting a deal on IT system. The introduction includes the characteristic of information needs which are directly or indirectly the basic elements of future IT system. The next problem mentioned in the paper is the process of gaining and describing requirements for IT system as well as other key elements that must be present in the agreement. All this matters are the basis of potential claims. The author also shows advantages od negotiation process, which is inherent element of cutting deals.

Małgorzata Pańkowska

CELE ROZWOJU ARCHITEKTURY PRZEDSIĘBIORSTWA

Wprowadzenie

Architektura w swym tradycyjnym ujęciu stanowiła, a obecnie także w obszarze technologii informacji (*information technology*, IT) stanowi synergię nauki i sztuki projektowania struktur złożonych dla zapewnienia funkcjonalności i kontroli ewolucji tej złożoności. Niniejsze opracowanie obejmuje prezentację podstawowych zagadnień rozwoju architektury przedsiębiorstwa, koncentruje uwagę na celach i korzyściach rozwoju architektury przedsiębiorstwa oraz stanowi próbę wykazania na podstawie studiów literatury przedmiotu prawdziwości hipotezy, że rozwój architektury przedsiębiorstwa determinuje formułowanie i realizację strategii biznesowej.

1. Interpretacja pojęcia architektury przedsiębiorstwa

Pojęcie architektury przedsiębiorstwa pojawiło się na przełomie lat 80. i 90. ubiegłego wieku i obejmowało zasady i modele graficznej prezentacji struktur technologii i systemów informatycznych. Należy zacząć od tego, że specjaliści rozwoju metod architektury sformułowali odrębną definicję przedsiębiorstwa. Przedsiębiorstwo to celowa kooperacja społeczna [OPWC09]. Przedsiębiorstwo jest kolekcją korporacyjnych lub instytucjonalnych encji, a przykładem przedsiębiorstwa jest cała korporacja, dział lub wydział korporacji, grupa geograficznie rozproszonych organizacji, agencja rządowa, grupa instytucji rządowych [Mino08]. Celem architektury przedsiębiorstwa jest konstruowanie środowiska IT w przedsiębiorstwie, przy ścisłym powiązaniu gospodarowania IT ze strategią przedsiębiorstwa [Mino08]. W ramach rozwoju architektury przedsiębiorstwa istotne jest stworzenie mapy aktywów IT i procesów biznesowych oraz zbioru zasad gospodarowania. Architektura przedsiębiorstwa zapewnia holistyczne wyrażenie kluczowych problemów rozwoju strategii przedsiębiorstwa, oddziałuje na podstawowe procesy i funkcje. Może być interpretowana jako wiedza, która steruje przedsiębiorstwem, jako informacja konieczna dla prowadzenia przed-

siębiorstwa, jako technologia i ogólny proces wdrażania IT w odpowiedzi na zmieniające się potrzeby biznesowe. Wygenerowanie architektury wzbogaca możliwości przedsiębiorstw zarówno komercyjnych, jak i budżetowych dla dostosowania się do przemian technologicznych i rynkowych oraz dla zarządzania zmianą.

Architektura przedsiębiorstwa jest koherentną całością zasad, metod i modeli, które są używane w projektowaniu i realizacji struktury organizacyjnej przedsiębiorstwa, procesów biznesowych, systemów informacji i infrastruktury [Lank05]. Definicja architektury przedstawiona w normie IEEE Standard 1471-2000 stanowi, że architektura jest fundamentalną organizacją systemu złożonego z komponentów, powiązanych ze środowiskiem i wzajemnie oraz obejmuje zasady ustalające metody projektowania i ewolucji. Pięć poziomów architektury przedsiębiorstwa zostało wyróżnionych w 1989 roku przez US National Institute of Standards (NIST). Pozostają one ważne także w środowisku rozszerzonego przedsiębiorstwa prowadzącego działalność w warunkach gospodarki elektronicznej [Gove05], zatem na architekturę przedsiębiorstwa składają się:

- architektura jednostek biznesowych, obejmująca procesy biznesowe, krajowe i międzynarodowe standardy współdzielenia się informacją oraz praktyki biznesowe,
- architektura informacji, obejmująca oryginalne dokumenty, przeglądy i dane odpowiednich jednostek, standardy, procedury niezbędne dla zapewnienia integralności informacji, konwencje nazywania i metody opisu,
- architektura systemu informacji, obejmująca oprogramowanie użytkowe, ustalająca zręby dla zaspokojenia specyficznych wymagań informacji. Komponenty architektury systemów informacji dotyczą specyfikacji wymagań, aplikacji, modułów, baz danych i procedur,
- architektura danych – komponenty mogą obejmować modele baz danych, definicje danych, słowniki danych. Rozwój architektury danych służy minimalizacji redundancji i wspomaga nowe aplikacje,
- architektura infrastruktury technicznej obejmująca sprzęt, oprogramowanie i urządzenia komunikacji.

Podobną specyfikację warstw architektury przedsiębiorstwa przedstawiają Jaap i in. [JDPi06]. Architektura przedsiębiorstwa jest bazą strategicznych aktywów informacji, która definiuje informacje i technologie konieczne dla wypełnienia misji przedsiębiorstwa [JDPi06].

2. Metody i modele architektury przedsiębiorstwa

Architekturę przedsiębiorstwa buduje się na podstawie odpowiedniej metodyki, która jest zestawem pojęć, notacji, modeli, języków, technik i sposobów postępowania służących do analizy dziedziny stanowiącej przedmiot projekto-

wania oraz do projektowania pojęciowego, logicznego lub fizycznego. W rozwoju architektury przedsiębiorstwa stosuje się dwie zasady:

- zasadę dekompozycji: rozdzielenie złożonego problemu na podproblemy, które można rozpatrywać i rozwiązywać niezależnie od siebie i niezależnie od całości,
- zasadę abstrakcji: eliminacja, ukrycie lub pominięcie mniej istotnych szczegółów rozważanego przedmiotu lub mniej istotnej informacji: wyodrębnienie cech wspólnych i zbioru niezmienników [Baze94].

Metoda architektury przedsiębiorstwa jest uporządkowanym zbiorem technik oraz procesów dla tworzenia i konserwacji architektury przedsiębiorstwa. Metody określają różne fazy cyklu życia architektury, definiują produkty, które powinny powstawać na każdym etapie, określają sposoby ich weryfikacji i walidacji [Lank05]. W literaturze przedmiotu wymienia się następujące metody rozwoju architektury:

- Rational Unified Process definiuje iteracyjny proces, który pozwala na wykonanie oprogramowania przez dodanie funkcjonalności do architektury na każdym etapie. Zgodnie z tą metodą jest konstruowany Zunifikowany Proces Przedsiębiorstwa (Enterprise Unified Process);
- UN/CEFACT Modelling Methodology (UMM) jest metodologią konstrukcji modelu informacji i inkrementalnych procesów biznesowych [Huem09];
- TOGAF Architecture Development Method (ADM). Metoda obejmuje działania skupione wokół następujących zagadnień:
 - a) wizja architektury,
 - b) architektura biznesu,
 - c) architektura systemów informacji,
 - d) architektura technologii, którym to zagadnieniom są poświęcone zadania:
 - tworzenie podstaw,
 - analizowanie poglądów,
 - tworzenie modelu architektury,
 - wybór usług,
 - ustalenie obiektów biznesowych,
 - definiowanie kryteriów,
 - definiowanie modelu architektury,
 - analiza luk i nieścisłości,
 - planowanie migracji,
 - zarządzanie implementacją i zmianą architektury [Lank05].

Studia literatury prowadzą do stwierdzenia, że jeśli pojawia się nowa strategia biznesowa w przedsiębiorstwie, to potrzebna jest nowa lub zmodyfikowana architektura przedsiębiorstwa [Mino08]. Konstrukcja nowego modelu architektury wymaga uwzględnienia istniejącej bazy aktywów IT, istniejącej architektury przedsiębiorstwa, istniejących standardów architektury przedsiębiorstwa,

zasad przedsiębiorstwa, praktyk, pożądanego strategii biznesowej, dostępnych modeli i narzędzi dla rozwoju nowej architektury przedsiębiorstwa lub modyfikacji architektury istniejącej. Modele architektury są stosowane do analizy architektur alternatywnych, planowania biznesowego przejścia z obecnej do nowej architektury, komunikacji między organizacjami zaangażowanymi w rozwój, produkcję, działanie i konserwację modelu przedsiębiorstwa, do wspomagania komunikacji między interesariuszami projektu, do weryfikacji i zapewnienia możliwości certyfikacji wdrażanej architektury, do rozwijania i konserwacji dokumentacji, do wprowadzenia systemów informatycznych nowych generacji i nowych narzędzi analizy, do planowania i wspomagania budżetowania oraz przeglądu i oceny w całym cyklu życia systemów informatycznych. Najbardziej znane modele architektury przedsiębiorstwa to modele następujące:

- Model architektury przedsiębiorstwa Zachmana,
- Model TOGAF (The Open Group Architecture Framework),
- Model GERAM (Generic Enterprise Reference Architecture and Methodology),
- Model PERA (Purdue Enterprise Reference Architecture),
- Model CIMOSA (Computer Integrated Manufacturing Open System Architecture),
- Model LEA (Lightweight Enterprise Architecture),
- Model Nolan Norton Framework,
- Model E2AF (Extended Enterprise Architecture Framework),
- Model planowania architektury przedsiębiorstwa (EAP, Enterprise Architecture Planning),
- Model FEAF (Federal Enterprise Architecture Framework),
- Model TEAF (Treasury Enterprise Architecture Framework),
- Model architektury zintegrowanej (IAF, Integrated Architecture Framework),
- Model JTA (Joint Technical Architecture),
- Model C4ISR (Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance),
- Model DoDAF (Department of Defense (DoD) Architecture Framework),
- Model DoD TRM (Department of Defense Technical Reference Model),
- Model TAFIM (Technical Architecture Framework for Information Management),
- Model SAGA (Standards and Architecture for eGovernment Applications) [BNSc03, Lank05, Mino08].

Tabela 1

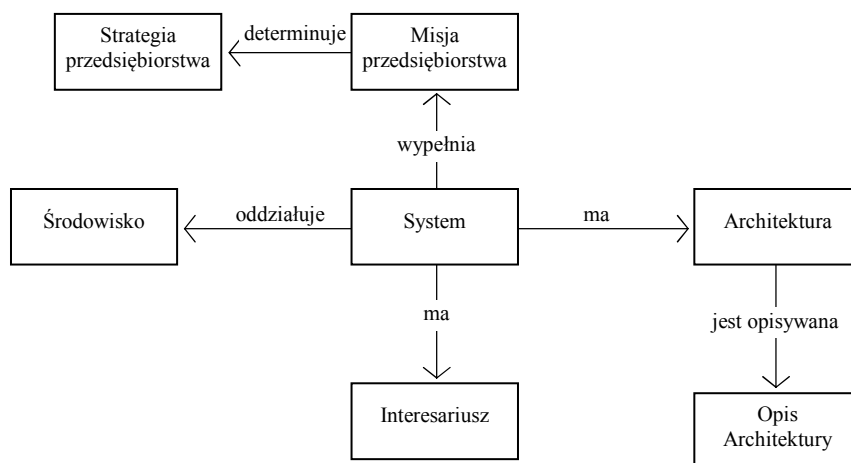
Wybrane modele rozwoju architektury przedsiębiorstwa

Model	Charakterystyka modelu
Siatka Zachmana [Lank05]	Jeden z najstarszych modeli architektury. Model obejmuje pięć aspektów modelowania: zakresu, (kontekstowy), conceptualny, logiczny, fizyczny oraz szczegółowych reprezentacji. Wymaga odpowiedzi na pytania o dane, funkcje, sieci, ludzi, czas i motywacje
Model TOGAF [Lank05]	<p>Komponenty:</p> <ul style="list-style-type: none"> – wysokiego poziomu struktura oparta na wybranych kluczowych koncepcjach i metodzie zwanej Architecture Development Method (ADM), – TOGAF Resource Base, zbiór narzędzi i technik dostępnych dla użycia w zastosowaniu TOGAF i TOGAF ADM (np.: scenariusze biznesowe, język modelowania rozwoju architektury (Architecture Development Modelling Language, ADML), studia przypadków, inne osnowy architektur, mapowanie TOGAF do modelu Zachmana), – TOGAF Enterprise Continuum. <p>Enterprise Continuum zawiera:</p> <ul style="list-style-type: none"> – TOGAF Foundation Architecture, która zawiera Model Referencyjny Techniczny (Technical Reference Model), Bazę Informacji o Otwartych Standardach (Open Group's Standards Information Base (SIB)), Bazę Informacji o Blokach Konstrukcyjnych (Building Blocks Information Base (BBIB)), – Model Referencyjny Infrastruktury Informacji Zintegrowanej (Integrated Information Infrastructure Reference Model)
Model GERAM [BNSc03]	Model podstawą normy ISO 15740:2000. Struktura modelu obejmuje: GERA (Generalized Enterprise Reference Architecture) model integrujący koncepcje integracji przedsiębiorstwa, EEM (Enterprise Engineering Methodology) model opisujący metodę inżynierii przedsiębiorstwa, Język EML (Enterprise Modelling Language) zapewniający konstrukcje dla modelowania ról, procesów i technologii, Narzędzia EETs (Enterprise Engineering Tools) wspomagające inżynierię przedsiębiorstwa, Koncepcje GEMCs (Generic Enterprise Modelling Concepts) określające nowe znaczenie konstrukcji modelowania przedsiębiorstwa, Modele EMs (Enterprise Models) stanowiące projekty przedsiębiorstwa i modele dla wspomagania analizy i dalszego projektowania, Modele PEMs (Partial Enterprise Models) zapewniające wielokrotnego zastosowania modele referencyjne i projekty ról, procesów i technologii, Modele EMOs (Enterprise Models) wspomagające wdrażanie działań, procesów operacyjnych i technologii, EOS (Enterprise Operational Systems) wspomagające działanie konkretnego przedsiębiorstwa
Model CIMOSA [BNSc03]	Celem jest modelowanie procesów gospodarczych oraz identyfikacja informacji niezbędnych do realizacji tych procesów w przedsiębiorstwie przemysłowym. Zintegrowana infrastruktura CIMOSA wspomaga procesy symulacyjne modelu i jego optymalizację w fazie użytkowania systemu

Ze względu na ograniczoną wielkość artykułu opisy pozostałych modeli architektury są dostępne w literaturze przedmiotu [Mino08, Lank05, BNSc03].

3. Modele architektury przedsiębiorstwa dla wspomagania strategii

Strategia może być definiowana jako całość wyborów, które zapewniają ogólną orientację dotyczącą przyszłego rozwoju przedsiębiorstwa [Hoog09]. Rozwój strategii przedsiębiorstwa bardziej jest ogólnym myśleniem i procesem uczenia się, aniżeli odgórnym procesem planowania, zważywszy, że strategia wyłania się ze wspólnej gry różnych wzajemnie na siebie działających tematów. Niektóre ważne tematy to: zewnętrzny rozwój, rozwój architektury przedsiębiorstwa, obszary problemowe, infrastruktura i usługi przedsiębiorstwa, zarządzanie cyklem życia przedsiębiorstwa. Architektura przedsiębiorstwa zmierza do zajmowania się problemami w spójny i zintegrowany sposób. Tym samym oferuje medium dla osiągnięcia wspólnego zrozumienia wśród wszystkich interesariuszy zaangażowanych w ewolucję przedsiębiorstwa i rozwój oparty na konceptualizacji systemów informacyjnych. Znaczenie architektury przedsiębiorstwa dla rozwoju jego strategii biznesowej widoczne jest w definicji architektury sformułowanej jako standard IEEE 1471-2000 i przedstawionej w modelu obiektów [Mino08]. Zgodnie ze standardem IEEE 1471-2000 architektura przedsiębiorstwa jest wyrażana przez swoje opisy. W modelu architektury wyróżnia się jeden lub więcej systemów informatycznych, które oddziałują na środowisko i urzeczywistniają misję. W autorskim rozszerzeniu tej koncepcji przyjmuje się, że architektura przedsiębiorstwa i jej systemy informatyczne poprzez oddziaływanie na misję determinują ogólną strategię biznesową przedsiębiorstwa i jego oddziaływanie na otoczenie (środowisko) – por. rys. 1.



Rys. 1. Rozszerzenie fragmentu konceptualnego ujęcia architektury przedsiębiorstwa według standardu IEEE 1471-2000

Źródło: Na podstawie: [Mino08].

W literaturze przedmiotu wymienia się pragmatyczne szkoły projektowania lub szkoły pozycjonowania jako podejścia do rozwoju strategii biznesowej [BNSc03]. Pragmatyczne podejście zakłada, że organizacje mogą adaptować się do zmian środowiskowych przez swoją restrukturyzację w zamierzony sposób, dokonując wyborów opartych na badaniach. Strategia jest zasadniczo wyborem na wysokim poziomie sposobu, w jaki organizacja zmierza do osiągnięcia swojej misji i wizji. Przełożenie formułowania strategii na wykonanie strategii jest zwane strategią w działaniu (*strategy in action*) i jest związane z trzema głównymi obszarami: przygotowaniem organizacji, alokacją zasobów, zarządzaniem zmianą [OPWC09].

Najwięcej przekonujących argumentów o potrzebie rozwoju architektury przedsiębiorstwa dla wspomagania strategii organizacji dostarcza Theuerkorn w swoim opracowaniu przedstawiającym model LEA (Lightweight Enterprise Architecture) – [Theu05]. Model LEA zawiera ujęcia trzech perspektyw, na które składają się: Strategiczna Architektura, Architektura Koncepcji i Architektura Wykonania. Strategiczna Architektura dotyczy budowania strategicznych zasad i rozwijania wytycznych dla pracy liderów organizacji. Przyjmując na wejściu wizję i cele biznesowe, model Strategicznej Architektury określa środki i praktyki dla wdrożenia technologii przedsiębiorstwa. W modelu LEA Strategiczna Architektura jest ciągłym procesem translacji strategii biznesu i zapewnienia ram wytyczających pomoc technologiczną. Plan strategiczny IT służy jako narzędzie komunikacji dla przywódców w zrozumieniu ogólnego kierunku technologii w organizacji i zapewnieniu kontekstu dla planów szczegółowych. Wyniki Strategicznej Architektury stanowią wejścia informacyjne dla rozwoju następnej perspektywy Architektury Koncepcji. Architektura Koncepcji zawiera mapowanie na wysokim poziomie systemu organizacji dla kształtowania przyszłej ewolucji systemu oraz dla użycia, a także lepszego zrozumienia potrzebnych zasobów i funkcjonalnych potrzeb organizacji. W końcu, ostatnia perspektywa, Architektura Wykonania, zapewnia makroujęcie szczegółów implementacji i pomaga w realizowaniu ewolucji systemu. Wyniki działań w Architekturze Wykonania tworzą artefakty, które pomagają wytyczać implementację projektową. Architektura Wykonania jest budowana na efektach zarówno Architektury Strategicznej, jak i Architektury Koncepcji.

Architektura przedsiębiorstwa może pomóc organizacji i procesom transformacji w efektywnej realizacji strategii. Jako taka działa jako instrument aktywnego planowania i sterowania, co może być zastosowane w translacji strategii na programy i projekty oraz dotyczy czterech głównych komponentów: zasad, modeli, widoków, osnów [OPWC09]. W istocie architektura przedsiębiorstwa jest narzędziem zarządzania złożonością i ryzykiem. Umożliwia informowane podejmowanie decyzji, planowanie i gospodarowanie transformacją. Może być stosowana:

- w ramach strategicznego planowania biznesu/IT,
- dla dopasowania strategicznych celów i IT,
- dla definiowania i wytyczania wielkoskalowych transformacji biznesowych i IT,
- dla reinżynierii struktur organizacyjnych,
- dla umożliwienia projektowania sieci organizacyjnych,
- dla definiowania i monitorowania programów IT [OPWC09].

Mierzalne korzyści rozwoju architektury przedsiębiorstwa to przede wszystkim optymalizacja aktualnych systemów i doskonalenie przyszłych możliwości. Dobrą praktyką jest postrzeganie systemów jako funkcjonalnych komponentów i podejmowanie prób unikania szczegółów implementacji jako nieodpowiednich na tym poziomie. Pomaga to stworzyć rozwiązania zorientowane na zapewnienie interoperabilności z innymi systemami. Wartości generowane dla przedsiębiorstwa dzięki pracy nad jego architekturą to przede wszystkim:

- zapewnienie pełnego oraz spójnego oglądu i zrozumienia przedsiębiorstwa, tj. ról, procesów, celów, polityk, reguł, wydarzeń i lokalizacji,
- doskonalenie procesów biznesowych,
- podejmowanie decyzji w warunkach rozproszenia organizacji i kontraktów organizacyjnych,
- ocena możliwości tworzenia nowych produktów,
- translacja strategii na wykonalne projekty,
- identyfikacja sposobności do outsourcingu i insourcingu w obszarze IT.

Najważniejsze jest, że przez projektowanie spójnej konceptualizacji rozwiązania zapewnia się, że projekty realizujące rozwiązania są wzajemnie komplementarne i kompatybilne. Architektura przedsiębiorstwa nie zmierza do zastąpienia klasycznego zarządzania projektami IT, ale stanowi uzupełnienie. Jest środkiem, a nie celem samym dla siebie, pozwala na uwzględnienie w projektowaniu systemów takich kwestii, jak interoperabilność, korzyści skali, możliwości zastosowania standardów.

Literatura

- [Baze94] Bazewicz M.: *Metody i techniki reprezentacji wiedzy w projektowaniu systemów*. Wydawnictwo Politechniki Wrocławskiej, Wrocław 1994.
- [BNSc03] Bernus P., Nemes L., Schmidt G.: *Handbook on Enterprise Architecture*. Springer, Berlin 2003.
- [Gove05] *Governance of the Extended Enterprise, Bridging Business and IT Strategies*, IT Governance Institute. Wiley & Sons, London 2005.
- [Hoog09] Hoogervorst J.A.P.: *Enterprise Governance and Enterprise Engineering*. Springer, Berlin 2009.

-
- [Huem09] Huemer Ch.: *UN/CEFACT's Modeling Methodology (UMM) in a Nutshell*, UN/CEFACT TMG, 18 September 2009, http://www.unece.org/cefact/umm/UMM_userguide-nutshell.pdf [dostęp: maj 2011].
- [JDPi06] Jaap B., Van Dorn M., Piyusch M.: *Making IT Governance Work in a Sarbanes-Oxley World*. Wiley & Sons, Hoboken, 2006.
- [Lank05] Lankhorst M.: *Enterprise Architecture at Work*. Springer, Berlin 2005.
- [Mino08] Minoli D.: *Enterprise Architecture A to Z, Frameworks, Business Process Modeling, SOA, and Infrastructure Technology*. CRC Press, London 2008.
- [OPWC09] Op't Land M., Proper E., Waage M., Cloo J., Steghuis C.: *Enterprise Architecture, Creating Value by Informed Governance*. Springer, Berlin 2009.
- [Theu05] Theuerkorn F.: *Lightweight Enterprise Architectures*. Auerbach Applications, London 2005.

ENTERPRISE ARCHITECTURE DEVELOPMENT PURPOSES

Summary

In the paper, enterprise architecture is assumed as connection of science and art of complex systems design to ensure their functionalities and evolution control. The paper covers discussion on corporate architecture development purposes and benefits of the development. In the last part, author argues that enterprise architecture determines formulation and realization of corporate strategy.

Małgorzata Pańkowska

JĘZYK SYSML W INŻYNIERII WYMAGAŃ

Wprowadzenie

W konstrukcji każdego systemu ważne jest by wymagania były właściwie rozumiane. Inżynieria systemów oparta na modelach może pomóc w zarządzaniu złożonością, wspomaga doskonalenie jakości produktów i cyklu życia systemu informatycznego, usprawnia komunikację między zespołami oraz ułatwia gromadzenie wiedzy generowanej w projekcie i ewolucję projektu. W opracowaniu przedstawiono zasadnicze własności języka inżynierii systemów – SysML ze zwróceniem uwagi na jego użyteczność w inżynierii wymagań.

1. Inżynieria systemów i wymagań

Inżynieria systemów jest dyscypliną, która koncentruje się na projektowaniu i zastosowaniu systemu jako całości złożonej z części. To wymaga ujęcia holistycznego, przy uwzględnieniu różnych aspektów zarówno technicznych, jak i społecznych. Inżynieria systemów jest procesem iteracyjnym planowania, rozwoju i administrowania złożonymi systemami, szczególnie systemami komputerowymi [EnRo03]. Inżynieria oprogramowania jest tylko częścią inżynierii systemów, która dotyczy systematycznego rozwoju, ewaluacji i konserwacji oprogramowania. Proces inżynierii systemów jest zgodny z modelem SIMILAR:

- postawienie problemu (*state the problem*),
- wyszczególnienie alternatyw (*investigate alternatives*),
- modelowanie systemu (*modelling system*),
- integracja (*integration*),
- uruchomienie systemu (*launching the system*),
- oszacowanie rezultatów (*assess performance*),
- ponowna ocena (*re-evaluation*) [WrMa10].

Wymaganie powinno być zawsze traktowane jako wymaganie systemowe. Wymagania odnośnie do oprogramowania są tylko częścią wymagań systemowych. Inżynieria wymagań staje się istotna z wielu powodów:

- wszystkie podejścia systemowe rozpoznają potrzebę zrozumienia wymagań we właściwy sposób, a to wiąże ze sobą wszystkie rodzaje systemów, technicznych, społecznych, finansowych etc.,
- jakość i dopasowanie do celu, działanie systemu zgodnie z oczekiwaniami,
- projekt sterowany wymaganiami, brak zrozumienia wymagań projektowych generuje brak zrozumienia całości systemu,
- benchmarking i testowanie dla uzyskania akceptacji przez użytkownika,
- generowanie zaufania wśród interesariuszy projektu poprzez demonstracje systemu do akceptacji,
- pełne śledzenie wymagań,
- niezależność wymagań od rozwiązania [HoPe10].

Zarządzanie wymaganiami obejmuje wszystkie działania dla zapewnienia, że artefakty tworzone przy zastosowaniu metod inżynierii wymagań są użyteczne w całym projekcie. Zarządzanie wymaganiami integruje wszystkie dostępne dane projektowe z dostępnymi danymi dotyczącymi wymagań i może być postrzegane jako istota administracji projektowej. Kluczowe działania inżynierii wymagań to ujawnienie, specyfikacja, analiza i przegląd wymagań. Dla rozpoznania wymagań stosuje się techniki, takie jak opisywanie wybranego fragmentu rzeczywistości, analiza dokumentów, listy sprawdzające, burze mózgów, mapy myśli dla graficznej prezentacji powiązań z wybranym obiektem, wywiady, obserwacje, przypadki użycia, scenariusze i opowiadania. Poza tym metody te mogą być stosowane dla identyfikacji oraz definiowania interfejsów, interesariuszy i ich ról w projekcie. Techniki rejestracji wymagań zależą od charakterystyki branżowej przedsiębiorstwa, dla którego jest budowany system, jego aplikacji i projektów. Istnieje wiele technik podstawowych, wspomagających inżynierię wymagań, a zaliczyć do nich można: wywiady, zarówno formalne, jak i nieformalne, wymagania biznesowe zapisane w dokumentach przedsiębiorstwa, informacje zwrotne konserwatorów produktów software'owych, okresowe badania opinii użytkowników, pracę w środowisku docelowym eksploatacji oprogramowania, konferencje użytkowników, studia czasopism fachowych i materiałów handlowych oraz prototypowanie zapewniające użytkownikom wizualizację efektów częściowych produkcji oprogramowania. Wymagania są definiowane w całym cyklu życia systemu, czyli na etapie konceptualizacji, rozwoju, aprobowania projektu, realizacji systemu, konserwacji, aktualizacji i wycofywania z eksploatacji. Inżynieria wymagań obejmuje określenie wymagań klienta i ich akceptację, określenie wymagań dotyczących projektu oraz wymagań systemu, ich integrację i testowanie systemu. Istotne problemy inżynierii wymagań obejmują:

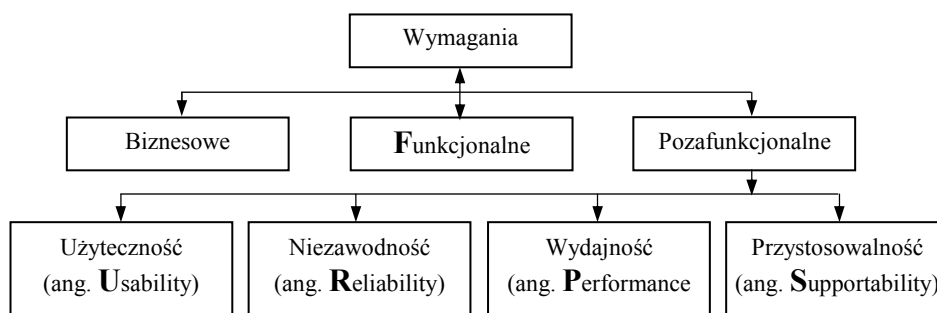
- identyfikowalność wymagań i konieczność zachowania troski, by każde wymaganie zostało unikatowo nazwane i opisane,

- filtrowalność wymagań i zapewnienie możliwości ekstrakcji informacji stosownie do roli interesariusza projektu,
- możliwość śledzenia, czyli obserwowanie różnych komponentów informacji w tym samym czasie lub śledzenie fragmentu informacji (wymagania) w dłuższym przedziale czasowym,
- wiązanie wymagań ze sobą, a także z testami, kryteriami jakości, rolami,
- informacje dodatkowe, tj. autor i właściciel wymagania, data utworzenia, komentarze, priorytety,
- prawa użytkownika odnośnie do przeglądania i edytowania wymagań [HWFPa08].

Należy troszczyć o specyfikację wymagań służyć minimalizowaniu ryzyka, które może być analizowane w różnych kontekstach, jako ryzyko biznesowe, techniczne, finansowe, środowiskowe. W inżynierii systemów istotne są trzy zagrożenia: złożoność wymagań, brak zrozumienia i problemy nieefektywnego komunikowania się interesariuszy projektu [HoPe10].

2. Wymagania i ich własności

W literaturze przedmiotu wyróżnia się trzy rodzaje wymagań: biznesowe, funkcjonalne i pozafunkcjonalne [HoPe10; GrCa87]. Wymagania biznesowe przedstawiają podstawową charakterystykę przedsiębiorstwa i są oparte na strategii biznesowej, planowaniu zmian, analizie rynków obecnych i przyszłych, misji przedsiębiorstwa. Wymaganie funkcjonalne jest ogólnie utożsamiane z wymaganiami użytkownika. Wymagania funkcjonalne i pozafunkcjonalne ujęte są w modelu FURPS (por. rys. 1). Pozafunkcjonalne wymagania reprezentują zalecenia, które ograniczają w pewien sposób inne wymagania. Przykłady wymagań pozafunkcjonalnych obejmują wymagania jakości, wymagania wdrożeniowe, wymagania zastosowania specyficznych rozwiązań.

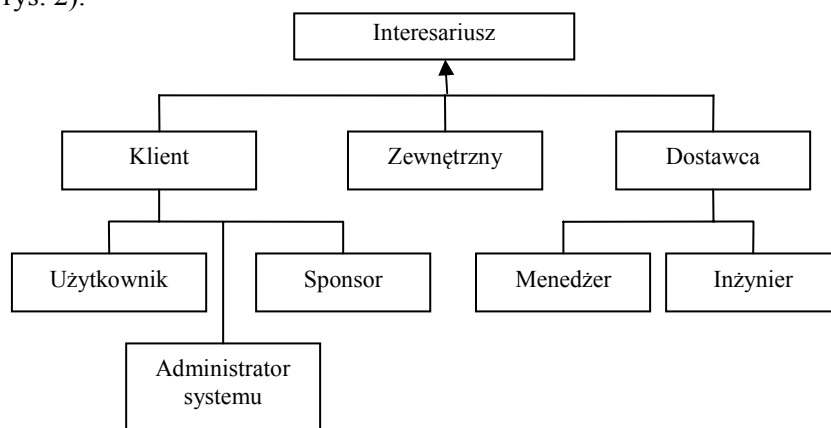


Rys. 1. Wymagania biznesowe i modelu FURPS

Źródło: [GrCa87].

Wymaganie może być prezentowane graficznie jako klasa, a własności wymagania są pokazane jako lista atrybutów. Własności wymagań są następujące: identyfikator, tekst, źródło, priorytet, kryteria weryfikacji i walidacji, opis, aktualność, stabilność. Uogólniony proces zarządzania wymaganiami może również być przedstawiony jako klasa. Atrybuty tej klasy to zbiór interesariuszy, zbiór wymagań, kryteria akceptacji, zbiór wyników przeglądu, wymagania szczególne (specyficzne), dokument rejestracji wymagań, opis projektu. Operacje na tej klasie są zdefiniowane następująco: eksponowanie i oczyszczanie wymagań, definiowanie kontekstu biznesowego, organizacji biznesowej, ról, projektów i usług, cyklu życia projektu, identyfikowanie ryzyka, analiza wymagań, przegląd, definiowanie kryteriów akceptacji, identyfikowanie interesariuszy, dokumentowanie procesu zarządzania wymaganiami [HoPe10]. Interesariuszem może być:

- klient, który korzysta z usług w systemie,
- interesariusz zewnętrzny, np. klient użytkownika, internauta,
- dostawca zaangażowany w rozwój i dostarczanie produktów i usług (por. rys. 2).



Rys. 2. Interesariusze inżynierii wymagań

Źródło: [HoPe10].

W inżynierii wymagań poza modelowaniem wymagań konieczne jest także modelowanie kontekstu. Przyjmuje się, że kontekst określa granice systemu i rozdziela świat zewnętrzny od możliwości i wymagań systemu. Używając technik modelowania, kontekst jest wizualizowany przy użyciu diagramu przypadku użycia i analizowany przy zastosowaniu scenariuszy, przy założeniu, że scenariusz jest sekwencją wydarzeń, które prowadzą do osiągnięcia konkretnego celu. Kontekst przedstawia system z punktu widzenia jakiegoś interesariusza. Zdaniem Jakubczyc i Owoca kontekst to wszystko to, co pośrednio wymusza rozumienie i interpretację pojęć, a więc pojedynczy czynnik lub sytuacja o różnym poziomie złożoności [JaOw11]. Kontekst w modelowaniu systemu służy jako odniesienie znaczeniowe, interpretacyjne lub sterujące.

3. Język modelowania systemów

Systems Modeling Language (SysML) jest językiem modelowania ogólnego zastosowania, który wspomaga specyfikację, analizę, projektowanie i weryfikację systemów informatycznych. SysML jest językiem modelowania graficznego przy zastosowaniu podstaw prezentowania wymagań, zachowania, struktury, własności systemu i jego komponentów [FMSt09]. UML 2 został wybrany jako podstawa dla SysML, co ułatwia integrację modelowania systemów i modelowania oprogramowania. Specyfikacja OMG SysML [WWW1] została przyjęta przez OMG w maju 2006 roku i formalna wersja 1.0 specyfikacji języka została wydana w sierpniu 2007 roku. SysML jest oparty na języku UML, więc nie może być traktowany jako całkowicie nowy język, lecz jako zbiór dodatkowych uzupełnień do istniejącego rdzenia koncepcji i diagramów modelowania UML. SysML korzysta z podzbioru UML2.1, co zwane jest UML4SysML. Znaczna część koncepcji UML została odrzucona, ponieważ uznano je za nieodpowiednie dla modelowania w inżynierii systemów. Zachowano diagram maszyny stanowej, sekwencji, diagram przypadku użycia. Inne diagramy zostały rozszerzone, tak jak diagramy aktywności.

Tabela 1

Porównanie SysML i UML2.x

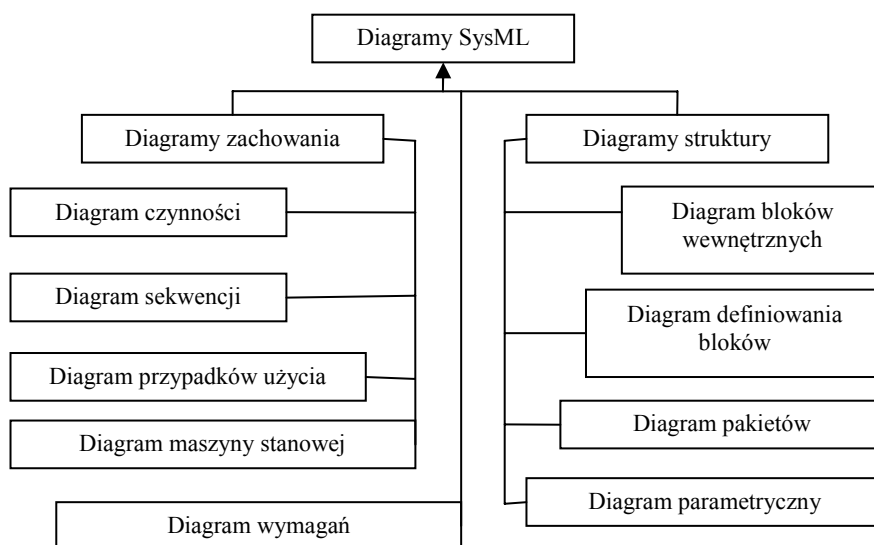
Diagram SysML	Opis	Odpowiednik w języku UML2.x
1	2	3
Diagram wymagań systemowych	Graficzne przedstawienie wymagań systemowych i ich relacji z innymi kategoriami modelowania systemu	Brak
Diagram przypadków użycia	Graficzne przedstawienie przypadków użycia, aktorów oraz związków między nimi	Diagram przypadków użycia
Rozszerzony diagram czynności	Graficzne przedstawienie sekwencyjnych lub współbieżnych przepływów sterowania oraz danych pomiędzy uporządkowanymi ciągami czynności, akcji i obiektów	Diagram czynności
Diagram sekwencji	Graficzne przedstawienie interakcji między aktorami, blokami, częściami bloków i obiektami w postaci sekwencji komunikatów wymienianych między kategoriami modelowania	Diagram sekwencji
Diagram maszyny stanowej	Graficzne odzwierciedlenie dyskretnych zachowań systemów	Diagram maszyny stanowej

cd. tabeli 1

1	2	3
Diagram definiowania bloków	Graficzne przedstawienie struktury systemu w postaci bloków, ich cech i związków	Diagram klas
Diagram bloków wewnętrznych	Graficzne przedstawienie wewnętrznej struktury bloku, wyrażanej przez wzajemnie powiązane części bloków	Diagram struktur połączonych
Diagram parametryczny	Przedstawienie ograniczeń parametrycznych występujących w modelowanym systemie	Brak
Diagram pakietów	Graficzne przedstawienie logicznej struktury systemu w postaci zestawu pakietów połączonych zależnościami i zagnieżdżeniami	Diagram pakietów

Źródło: Na podstawie: [WrM10a].

SysML odrzucił diagramy UML, takie jak diagram obiektu, komponentu, wdrożeniowe, komunikacji, przedziałów czasowych i interakcji. Diagramy struktury złożonej i klas zostały znacznie zmodyfikowane oraz zastąpione definicją blokową i diagramami bloków wewnętrznych. Ponadto, dodano dwa nowe diagramy, czyli diagram wymagań i parametryczny.



Rys. 3. Diagramy SysML

Źródło: [HoPe10; WrM10a].

Język SysML w wersji 1.1 zawiera 9 rodzajów diagramów, są to:

- diagram wymagań systemowych, przedstawia wymagania i ich relacje z innymi wymaganiami, elementami projektu, przypadkami testowymi dla wspomagania śledzenia wymagań,
- diagram czynności przedstawia działania oraz transformacje nakładów na efekty,
- diagram sekwencji przedstawia zachowanie w kategoriach sekwencji komunikatów wymienianych między stronami,
- diagram przypadków użycia przedstawia funkcjonalności używane przez jednostki zewnętrzne (aktorzy) dla osiągnięcia zbioru celów,
- diagram maszyny stanowej, przedstawia zachowania encji w kategoriach jej przejścia między etapami uruchamianymi przez wydarzenia
- diagram definiowania bloków przedstawia strukturalne elementy zwane blokami, ich kompozycje i klasyfikacje,
- diagram bloków wewnętrznych przedstawia interpołączenia i interfejsy między częściami bloków,
- diagram parametryczny przedstawia ograniczenia nałożone na wartości, stosowane do wspomagania analizy systemu,
- diagram pakietów przedstawia organizację modelu w kategoriach pakietów, które zawierają elementy modelu [HoPe10].

SysML oferuje możliwości modelowania, które pozwalają na prezentowanie systemów i ich komponentów przy użyciu:

- kompozycji komponentów strukturalnych, klasyfikacji i łączenia,
- modeli zachowań, co obejmuje przepływy działań, scenariusze interakcji, przekazywanie komunikatów, jak też zachowania reaktywne zależnie od stanu,
- przypisanie jednego elementu modelu do innego, jak np. funkcji do komponentów, oprogramowania do sprzętu,
- ograniczeń nałożonych na system, takich jak dotyczących wykonania, wiarygodności, własności fizycznych,
- hierarchii wymagań i ich relacji z innymi elementami modelu [DFJS10].

Oczekuje się, że SysML zapewni unifikację zróżnicowanych języków modelowania obecnie stosowanych, podobnie jako UML. SysML pozwala na modelowanie wymagań systemu, zachowanie systemu i strukturyzację systemu. Wprowadzenie diagramu wymagań w SysML pozwala na modelowanie strukturalnych relacji między wymaganiami, mimo że wymagania są tradycyjnie przedstawiane przy użyciu diagramów przypadków użycia, które ujmują wymagania w aspekcie zachowań.

4. Diagramowanie wymagań

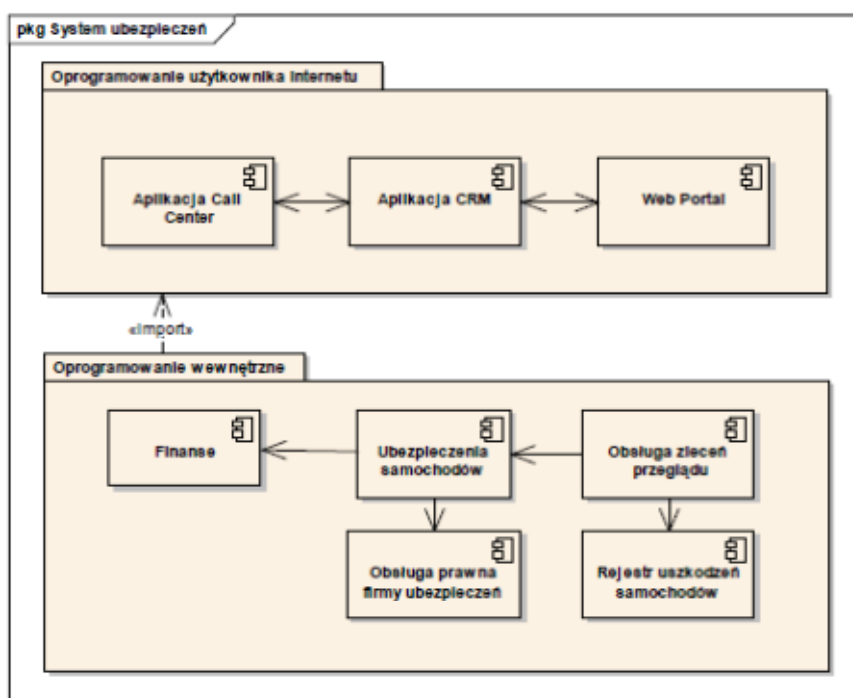
W języku SysML wymagania oznaczają kontrakt między organizacją zamawiającą system a jego wykonawcami. Podstawowymi własnościami, które charakteryzują wymagania są w SysML numer porządkowy i treść wymagania. W praktycznych zastosowaniach wymagania są uporządkowane hierarchicznie. W języku SysML diagram wymagań systemowych umożliwia graficzne przedstawienie wymagań systemowych i ich relacji z innymi kategoriami modelowania systemu. Wymagania specyfikuje się na podstawie kategorii modelowania diagramów wymagań systemowych, tj.:

- wymaganie (*requirement*),
- związek (*relationship*),
- blok (*block*),
- przypadek użycia (*use case*),
- testowy przypadek użycia (*test case*),
- pakiet (*package*).

Wymagania w diagramach wymagań systemowych języka SysML łączy się przez związki zagnieżdżenia (*containment*) umożliwiające tworzenie wielopoziomowej hierarchii wymagań oraz przez zależności (*dependencies*). Wskazują one na charakter logicznej zależności między poszczególnymi wymaganiami. Bloki, przypadki użycia i testowe przypadki użycia są kategoriami modelowania, istotnymi z punktu widzenia precyzji opisu kontekstu poszczególnych wymagań oraz monitorowania sposobu ich implementacji w systemie. Na diagramach wymagań systemowych stosuje się je w połączeniu z odpowiednimi rodzajami zależności. Blok w SysML jest podstawową jednostką struktury używaną dla definiowania encji. Encją może być komponent systemu, część sprzętu, oprogramowania, komponent danych, osoba, usługa. Bloki mają zdefiniowane własności, które można podzielić na własności strukturalne, własności interakcji i ograniczenia. W SysML wyróżniono następujące rodzaje własności:

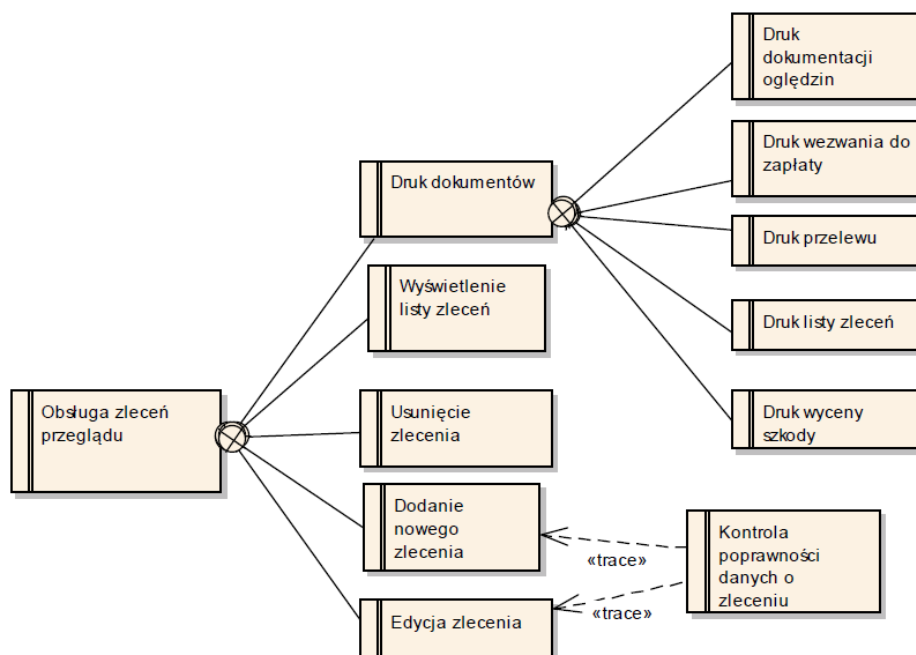
- własności części, które opisują hierarchiczną dekompozycję bloku,
- własności referencyjne, które opisują relacje między blokami,
- własności wartości, które opisują ilościowe charakterystyki bloku [FMSt09].

W SysML pakiety stanowią mechanizm ogólnego zastosowania, służący do organizowania dokumentacji, w tym dokumentacji wymagań. Pakiety i Diagramy pakietów są użyteczne w zarządzaniu złożonością modelu wymagań systemowych. Zastosowanie diagramów pakietów przedstawiono na rys. 4.



Rys. 4. Diagramy pakietów dla oprogramowania firmy ubezpieczeniowej

W SysML wymaganie przedstawia cechę, własność, zachowanie systemu. Zadanie definiowania i listowania wymagań jest wykonane w pierwszych krokach procesu projektowania systemu. Wymagania pozwalają projektantowi jasno ustalić czego się oczekuje od przyszłego systemu. Ponadto, wymagania tworzą punkt centralny procesu weryfikacji i walidacji, ponieważ są one kluczowym składnikiem dla dokładnego ustalania, co budowany system powinien robić i jak powinien to robić. SysML wprowadza diagram wymagań, którego wcześniej nie było w UML. Ten nowy diagram zapewnia środki, by wskazać wymagania i związać je z innymi modelami specyfikacji, projektowania i weryfikacji. Wymagania mogą być prezentowane w formie graficznej, tabularycznej lub struktury drzewiastej. Przykładowy diagram wymagań dla obsługi zlecenia firmy ubezpieczeniowej przedstawiono na rys. 5.



Rys. 5. Diagram wymagań obsługi zlecenia

Wymaganie może być dekomponowane na subwymagania w celu uporządkowania wielu wymagań jako drzewa wymagań składowych. Wymaganie może być związane z innymi wymaganiami, może być generowane lub wydobyte z innego wymagania. Ponadto, wymaganie może być spełnione przez pewne elementy modelu korzystając z relacji satysfakcji. W SysML diagram wymagań jest utworzony z trzech podstawowych elementów: wymagań, relacji i przypadków testowych. Wymagania są stosowane dla prezentacji wymagań systemu, które mogą być zarówno wzajemnie powiązane, jak i powiązane z innymi elementami systemu przez relacje. Przypadki testowe mogą być przyłączone do wymagań dla pokazania jak wymagania są testowane. W SysML relacje wymagań są następujące:

- zagnieżdżenie (*nesting*): wymaganie może być dekomponowane na jedno lub więcej wymagań,
- wyprowadzenie (*derive*): jedno lub więcej wymagań pochodzi z danego wymagania,
- kopiowanie (*copy*): dane wymagania mogą być użyte w innym kontekście, przy czym powielone wymagania mogą być wersją tylko do odczytu oryginalnego wymagania, wymagania powielone nie mogą być zmienione bez zmiany oryginału,

- weryfikowanie (*verify*): pokazanie jak przypadek testowy weryfikuje wymaganie,
- satysfakcja (*satisfy*): pokazanie, że element modelu spełnia wymaganie,
- śledzenie (*trace*): pokazanie, że element modelu i wymaganie są ściśle powiązane [DFJS10].

Diagram wymagań jest sposobem dokumentowania i zarządzania wymaganiami, ponadto może być punktem odniesienia dla przypadków użycia i scenariuszy.

Podsumowanie

Ogólnie ujmując wymagania sterują całością projektu, zatem wszystkie modele i informacje w projekcie powinny wspomagać zarządzanie wymaganiami. Dla oceny wymagań są formułowane metryki i kryteria, które dzielą się na kryteria ogólnej specyfikacji i kryteria poszczególnych wymagań. Kompletność, zgodność, konieczność, swoboda duplikacji stanowią przykładowe kryteria ogólne, natomiast do kryteriów szczególnych zalicza się niesprzeczność, wykonalność, identyfikowalność, atomiczność, poprawność dowodzenia, możliwość monitorowania i duplikowania. Sukces końcowy projektu zależy od tego, czy projekt zgodny jest z wymaganiami, zatem rozwój języka modelowania systemów SysML świadczy o zrozumieniu tych potrzeb i zapełnił istotną lukę w procesach budowy systemów informatycznych.

Literatura

- [DFJS10] Debbabi M., Hassaine Fawzi, Jarraya Y., Soeanu A., Alawneh L.: *Verification and Validation in Systems Engineering, Assessing UML/SysML Design Models*. Springer-Verlag, Berlin, Heidelberg 2010.
- [EnRo03] Endres A., Rombach D.: *A Handbook of Software and Systems Engineering, Empirical Observations, Laws and Theories*. Pearson Addison Wesley, London 2003.
- [FMSt09] Friedenthal S., Moore A., Steiner R.: *A Practical Guide to SysML, The Systems Modeling Language*. Morgan Kaufman, OMG Press, Burlington 2009.
- [GrCa87] Grady R., Caswell D.: *Software Metrics: Establishing a Company-wide Program*. Prentice Hall, New York 1987.
- [HoPe10] Holt J., Perry S.: *Modelling Enterprise Architectures*. The Institution of Engineering and Technology, London 2010.
- [HWFPa08] Hood C., Wiedemann S., Fichtinger S., Pautz U.: *Requirements Management, The Interface Between Requirements Development and All Other Systems Engineering Processes*. Springer, Berlin 2008.

- [JaOw11] Jakubczyc J.A., Owoc M.: *Reprezentowanie kontekstu, Wiedza i komunikacja w innowacyjnych organizacjach*. W: *Komunikacja elektroniczna*. Red. M. Pańkowska. Wydawnictwo UE, Katowice 2011.
- [WrMa10] Wrycza S., Marcinkowski B.: *Specyfikacja wymagań systemowych w języku SysML*. W: *Technologie informatyczne w administracji publicznej i służbie zdrowia*. Red. J. Goliński, A. Kobyliński, A. Sobczak. SGH, Warszawa 2010.
- [WrM10a] Wrycza S., Marcinkowski B.: *Język inżynierii systemów SysML. Architektura i zastosowanie, Profile UML 2.x w praktyce*. Helion, Gliwice 2010.
- [WWW1] <http://www.omg.sysml.org>.

SYSML LANGUAGE IN REQUIREMENTS ENGINEERING

Summary

The paper covers justification of the necessity to develop requirements specification for the high quality of information systems. First, system engineering and requirements engineering as disciplines are explained. Next, author focuses on information system requirements' classification and explains their features. Third, the Systems Modeling Language (SysML) is characterized and compared with UML 2.0. The last part includes presentation of SysML diagrams and discussions on their usefulness on the business information systems modeling.

Teresa Porębska-Miąc

PROJEKTOWANIE I WDRAŻANIE SYSTEMÓW CRM

Wprowadzenie

Pozyskiwanie nowych klientów jest procesem bardzo kosztownym. Szacuje się, że jedynie 1% wykonanych przez sprzedawcę rozmów telefonicznych przynosi spodziewane efekty w postaci podpisania umowy z klientem [CRMw09]. Właściwie zaprojektowany i wdrożony system CRM może znacząco zredukować koszty pracy działu handlowego. System pozwala przeprowadzić analizę klientów dostępnych w bazie danych i wybór jedynie tych, którzy potencjalnie mogą być zainteresowani realizacją transakcji. Celem artykułu jest przedstawienie specyfiki projektowania systemu CRM i zagadnień związanych z wyborem oraz wdrożeniem systemu CRM w organizacji.

1. CRM jako strategia biznesowa

CRM to rodzaj filozofii funkcjonowania firmy, która na pierwszym miejscu stawia klientów. Zarządzanie relacjami z klientem wywodzi się bezpośrednio z koncepcji marketingu relacyjnego (partnerskiego), tj. marketingu 1 do 1. Pojęcie to zakłada konieczność budowania indywidualnych, trwałych relacji między firmą i klientem oraz umieszczenie klienta i jego potrzeb w centrum zainteresowania firmy. Kontakty z klientami traktuje się nie jako zbiór odrębnych epizodów, lecz jako proces, na który można oddziaływać.

Kluczem do sukcesu we współczesnym biznesie jest utrzymywanie dochodowej grupy klientów. Praktyka biznesowa dowiodła, że zysk organizacji jest zazwyczaj generowany przez kilkunastoprocentowy segment klientów. CRM pozwala zidentyfikować ten segment, a potem właściwie nim zarządzać. Najlepsi klienci firmy zasługują na najlepsze traktowanie. Konieczne jest zatem znalezienie odpowiedzi na pytanie: kim są ci klienci i jaki mają potrzeby? Celem filozofii CRM nie jest bowiem zadowolenie wszystkich klientów. Może okazać

się, że najlepszym rozwiązaniem w odniesieniu do tej grupy klientów, która nie generuje zysku i nie rokuje perspektyw w przyszłości, będzie rezygnacja ze współpracy [Kupe10].

CRM to przede wszystkim strategia, która teoretycznie mogłaby funkcjonować bez żadnego narzędzia. Istnieje jednak wiele narzędzi informatycznych, które mogą skutecznie wspierać tę strategię. Należą do nich narzędzia transakcyjne (kalendarz kontaktów z klientem), narzędzia wspierające ten kontakt (call center i contact center), jak i narzędzia do analizy danych. Potrzebę wspomagania zarządzania relacjami z klientami przy pomocy narzędzi informatycznych spotęgował rozwój technologii teleinformatycznych, a zwłaszcza Internetu.

2. System klasy CRM

Technologia informacyjna jest narzędziem pozwalającym na efektywne stosowanie strategii CRM w organizacji. Efektem zastosowania narzędzi informatycznych w strategii CRM jest zintegrowany system obsługi klienta, dający jednolity obraz firmy i jedną bazę informacji o klientach. System klasy CRM to zintegrowany zbiór aplikacji wspierających organizację w ramach zdefiniowanej i wdrożonej strategii CRM.

Technologia informacyjna stwarza firmom możliwości w zakresie zdobywania i przetwarzania informacji dotyczących klientów. Pozwala lepiej poznać ich potrzeby, a tym samym zaproponować konkretnemu klientowi właściwy produkt we właściwym czasie, dzięki czemu klient przestaje być traktowany jako uczestnik pewnej grupy, lecz może być traktowany indywidualnie [Porę02].

Technologia informacyjna w kontekście CRM daje możliwość:

- przechowywania danych o kliencie w tak zintegrowany sposób, aby wgląd do informacji mogły mieć w tym samym czasie wszystkie uprawnione w firmie osoby,
- zautomatyzowania analiz dotyczących klientów i kontaktów z nimi (data mining), pozwalających na przekształcanie informacji w wiedzę [Kost02].

Współczesny system CRM posiada zintegrowany i kompleksowy charakter. Na kompleksowe rozwiązania CRM składa się wiele elementów: narzędzia automatyzujące procesy związane ze sprzedażą, narzędzia wspierające obsługę klientów i usługi serwisowe oraz automatyzujące działania marketingowe (analiza danych, zarządzanie kampaniami, zawartością internetową) – [Porę02]. Podstawowymi obszarami obsługiwanymi przez system CRM są marketing, sprzedaż i serwis. W obszarach tych mieści się m.in. zarządzanie akcjami marketingowymi, encyklopedią marketingową (analizy), procesem sprzedaży (oferty, zamówienia, kontrakty), siecią przedstawicieli terenowych, zgłoszeniami serwi-

sowymi itp. [Kupe10]. W systemie CRM sprzedaż produktu lub usługi może być monitorowana od momentu powstania potrzeby u klienta po świadczenie usług posprzedażowych. Istnieje także potrzeba integracji systemu CRM z systemami klasy ERP oraz innymi systemami informatycznymi funkcjonującymi w firmie (hurtownie danych, zarządzanie dokumentami, zarządzanie projektami, systemy pocztowe itp.), ponieważ procesy zarządzania relacjami z klientem łączą się z procesami *back office*.

W literaturze wyróżnia się trzy główne obszary wykorzystania systemu CRM, tj.:

- CRM operacyjny (*operational CRM*),
- CRM analityczny (*analytical CRM*),
- CRM interakcyjny (*interactive CRM*).

CRM operacyjny wspiera bezpośrednio działania związane z klientem, tj. sprzedaż, działania marketingowe i obsługę posprzedażową. System jest zorientowany na zdobywanie i gromadzenie danych o klientach, przeprowadzonych transakcjach, kontaktach, produktach, konkurencji oraz innych niezbędnych informacji potrzebnych do zapewnienia automatyzacji procesów zachodzących na styku firma-klient. System umożliwia kontaktowanie się z klientami, a także przeglądanie w dowolnym momencie bazy danych z informacjami dotyczącymi współpracy z klientem. System rejestruje wszystkie działania w historii kontaktu z klientem. Informacje te mogą zostać odtworzone w dowolnym momencie.

CRM analityczny umożliwia analityczne przetwarzanie danych, przekształcając je w wiedzę, niezbędną do wspomagania procesu podejmowania decyzji w zakresie optymalizacji relacji z klientami. System przechowuje, przetwarza i interpretuje dane o klientach, tworząc z nich raporty. Dane pochodzące z wielu źródeł, przechowywane w repozytoriach danych, są poddawane skomplikowanym analizom statystycznym, dzięki którym uzyskuje się wiedzę, pozwalającą zrozumieć potrzeby, preferencje nabywcze i zachowania klientów. Podstawową wartością aplikacji jest możliwość personalizacji analiz pod kątem korzystającego z niej użytkownika.

Funkcjonowanie CRM analitycznego jest silnie uzależnione od istnienia infrastruktury hurtowni danych, która integruje dane z różnych źródeł oraz ułatwia dostęp do nich. Do analizy danych o kliencie CRM analityczny wykorzystuje techniki eksploracji danych (data mining). Zastosowanie technologii data mining w systemach CRM pozwala przeprowadzić m.in. segmentację klientów, analizę LTV, analizę czasu przetrwania klienta, analizę odejścia klienta do konkurencji (lojalność klienta), analizy w czasie rzeczywistym, kierowanie kampaniami (analiza wydajności), opiekę nad klientem (analiza obsługi klienta oraz serwisu), analizę sprzedaży itp.

CRM interakcyjny, zwany także CRM kooperacyjnym, wspomaga bezpośredni kontakt firmy z klientami, udostępniając zestaw technologii ułatwiających ten kontakt za pośrednictwem zarówno tradycyjnych, jak i elektronicznych kanałów komunikacji. Jest traktowany jako centrum komunikacyjne – sieć koordynująca kanały kontaktowe z klientami, dostawcami i partnerami biznesowymi. Wszystkie funkcje realizowane przez interakcyjny CRM mają na celu przekształcenie kontaktów z klientami w długotrwałe partnerstwo, oparte na interakcji.

Dwie podstawowe cechy powinny charakteryzować system CRM: modułowość oraz otwartość. Modułowość oznacza, że dostosowując swoje rozwiązanie do potrzeb konkretnego użytkownika, proponuje mu się tylko wybrane elementy aplikacji, które są dla niego niezbędne, bez konieczności wdrażania całego zintegrowanego pakietu. Modułowość stwarza także możliwość dodatkowego instalowania następnych obszarów aplikacji bez potrzeby dokonywania drastycznych zmian w systemie.

Otwartość to cecha pozwalająca na funkcjonowanie systemu CRM w całkowitej symbiozie z innymi systemami funkcjonującymi w firmie, w szczególności z rozwiązaniami wspomagającymi pracę wewnątrz firmy (*back office*), takimi jak systemy MRPII/ERP. Warunkiem udanego wdrożenia systemu CRM jest jego dobra integracja z kluczowymi dla firmy aplikacjami (systemem finansowo-księgowym, produkcyjnym, logistycznym). Pozwala to na uniknięcie dublowania się informacji w różnych systemach oraz uzyskanie dostępu do wszystkich zgromadzonych informacji o kliencie w skali całej firmy, w efekcie czego np. handlowiec ma podgląd w bieżący stan rozliczeń z systemu finansowo-księgowego (od razu wie czy przyjąć zamówienie lub jaki udzielić rabat), przyjęte zamówienie automatycznie trafia do aplikacji produkcyjnej, logistycznej i do systemu płacowego (gdzie jest automatycznie rozliczany target i są naliczane prowizje). Innymi słowy system CRM powinien stanowić naturalne uzupełnienie infrastruktury informacyjnej istniejącej w firmach i nie może być traktowany wyłącznie jako nowe oprogramowanie [Porę02].

3. Specyfika projektowania systemu CRM

W literaturze tworzenie systemu CRM często porównuje się z budową domu. Każdy solidny dom musi powstać na solidnych fundamentach. Tymi fundamentami CRM jest filozofia biznesowa, umieszczająca klienta w centrum zainteresowania organizacji. Dopiero później może być tworzona struktura, którą stanowią strategie biznesowe. Końcowy etap budowy obejmuje prace wykoń-

zeniowe, na które składają się odpowiednie modele marketingowe i aplikacje informatyczne. Jedynie tak skonstruowana budowla pozwoli na udoskonalenie relacji z klientami [Prze07].

System CRM najlepiej jest budować krok po kroku, dodając z czasem coraz to bardziej skomplikowane funkcjonalności. Takie jest doświadczenie firm zachodnich, gdzie filozofia i narzędzia CRM stosowane są od kilkudziesięciu lat. Zdefiniowanie dokładnych potrzeb każdego z kroków powinno być poprzedzone głęboką analizą posiadanych zasobów informacyjnych oraz efektów, jakie zamierza się uzyskać.

Systemy CRM powinny być projektowane indywidualnie do potrzeb i specyfiki konkretnej firmy. Muszą także bezproblemowo współpracować ze wszystkimi systemami funkcjonującymi w firmie. System powinien także spełniać określone wymagania techniczne dotyczące przede wszystkim cech użytkowych, tj. funkcjonalności, bezpieczeństwa, dostępności, integracji z infrastrukturą technologiczną, wspomaganie operacji, użyteczności czy dostarczania informacji (raportowania) – [WąSt08].

Zaprojektowanie systemu CRM, zapewniającego spójność oprogramowania i dostęp do różnych kanałów komunikacji, jest przedsięwzięciem trudnym, kosztownym, czasochłonnym i wieloetapowym. Każdy tworzony system powinien być poprzedzony analizą i musi być projektowany zgodnie ze standardami oraz metodami wytwarzania oprogramowania. System zbudowany na podstawie rzetelnej analizy potrzeb systemu i prognozy rozwoju firmy pozwala podnieść efektywność oraz jakość zarządzania relacjami z klientem.

Projektowanie systemu odpowiada na pytanie, w jaki sposób założona przez analityka funkcja w modelu logicznym powinna zostać zrealizowana. W fazie tej powstaje pełny szczegółowy opis wszystkich założonych funkcji w systemie oraz pełny opis fizycznych struktur danych. Dzięki tak szczegółowemu opisowi, który pokazuje, jak wytworzyć dany system, maleje ryzyko popełnienia błędów w fazie implementacji, testowania i pielęgnacji systemu.

Na polskim rynku systemów CRM znajdują się oferty zarówno krajowych, jak i zagranicznych firm. Organizacje mają możliwość wyboru zakupu i wdrożenia systemu powielarnego lub zdecydowanie się na realizację systemu dedykowanego. W obu przypadkach eksploatacja systemu jest poprzedzona realizacją wszystkich lub wybranych faz cyklu życia systemu. Przy realizacji systemu dedykowanego lub dostosowaniu istniejących już rozwiązań do nowych potrzeb organizacji, konieczne będzie przejście wszystkich etapów budowy systemu. Przy wdrażaniu systemu powielarnego główny nacisk jest kładziony na fazę analizy, modelowania i wdrożenia, lecz często występuje także konieczność realizacji pozostałych faz cyklu życia systemu [Ziem01].

Proces projektowania systemu informatycznego stanowi sekwencję spójnych, powiązanych ze sobą czynności, od poznania potrzeb informacyjnych, przez zdefiniowanie problemu projektowego i opracowanie projektu, do prac związanych z implementacją, utrzymaniem, modernizacją i rozwojem systemu.

W pierwszej kolejności organizacja powinna sprecyzować swoją misję i zreorganizować strategię, a dopiero później zdecydować się na system CRM. System CRM to skuteczne narzędzie, jednak to właściciel lub menedżer musi odpowiedzieć sobie na pytanie: jak powinna wyglądać sprzedaż, marketing czy serwis? Jak zorganizować bazę klientów? Którzy z nich są faktycznymi klientami, którzy dostawcami czy partnerami? Jaką politykę wobec nich stosować? Ważne jest skoncentrowanie wszystkich wewnętrznych działów wokół klienta, gdyż system nie będzie odpowiedni dla firm, które nie stawiają klientów w swoim centrum [MaMa04]. Należy zatem najpierw usystematyzować sferę biznesową, a dopiero potem ją informatyzować. Opracowanie właściwej strategii biznesowej i zmiana modelu działania firmy są kluczowe dla powodzenia przedsięwzięcia, jakim jest system CRM.

System CRM wspiera procesy dotyczące marketingu, sprzedaży i obsługi klienta, zatem przedstawiciele tych działów będą najlepszymi partnerami wspomagającymi proces definiowania realnych potrzeb niezbędnych do pracy ich komórek. System CRM ma usprawniać i automatyzować ich procesy, co w konsekwencji powinno przełożyć się na wymierne rezultaty. W pierwszej kolejności należałoby powołać zespół projektowy, którego zadaniem jest precyzyjne zdefiniowanie wymagań funkcjonalnych i technicznych dla systemu CRM [FiJR01].

4. Wdrażanie systemu CRM

Statystyki dowodzą, że 50%-70% wdrożeń systemów CRM jest nieudanych, a zaledwie 46% firm, którym się to udało, osiąga zamierzone cele. Nie skłania to do podjęcia szybkiej decyzji o wdrożeniu systemu klasy CRM. Rodzą się zatem pytania: czy w ogóle kupić taki system? Czy organizacja jest przygotowana do jego wdrożenia? Jaki system wybrać? Jakie kryteria powinny decydować o wyborze dostawcy? Czy powierzyć dostawcy systemu reorganizację procesów firmy? Jakie jest ryzyko, że się nie uda itp. [StSS02].

Wdrożenie systemu CRM to z jednej strony typowy projekt implementacji narzędzia informatycznego, związany z powołaniem zespołu odpowiedzialnego za realizację, ustaleniem harmonogramu prac, szkoleniem użytkowników, z drugiej zaś strony to proces wymagający przełamania oporu użytkowników i sukcesywnego przyzwyczajania do odmiennych metod pracy i uważnego wsłuchiwania się w potrzeby klienta [Zakr01]. Wdrożenie CRM musi rozpocząć się

od opracowania strategii, zaprojektowania procesów biznesowych i procedur pracy. Konieczne staje się dostosowanie wewnętrznej organizacji firmy, przeszkolenie i odpowiednie motywowanie pracowników oraz wybór i implementacja odpowiedniej technologii (systemy informatyczne, hurtownie danych).

Realizacja projektu i jego wdrożenie powinno być szczegółowo zaplanowane i przebiegać według sprawdzonej metodyki. Wprawdzie żadna metodyka nie gwarantuje zakończonego sukcesem wdrożenia, pozwala jednak nadzorować przebieg wdrożenia. Wdrożenie systemu CRM składa się z trzech podstawowych faz:

- analizy przedwdrożeniowej,
- wyboru rozwiązania i negocjacji z dostawcą,
- wdrożenia [Buch10].

Faza analizy przedwdrożeniowej powinna być traktowana jako punkt krytyczny całego przedsięwzięcia CRM. Rzetelnie przeprowadzona analiza daje gwarancję, że dokonany na jej podstawie wybór systemu będzie trafny. Faza wyboru rozwiązania CRM obejmuje takie działania, jak: specyfikacja wymagań względem systemu CRM, wstępna selekcja rynku systemów CRM, analiza i ocena rozwiązań na podstawie ofert dostawców, wybór systemu i dostawcy, precyzowanie wymagań i negocjowanie kontraktu oraz podpisanie umowy z dostawcą systemu CRM.

Obecnie na rynku oprogramowania jest dostępna bogata oferta systemów CRM posiadających różne cechy użytkowe. W takiej sytuacji wybór najlepszego rozwiązania staje się problemem złożonym, wieloetapowym i wielokryterialnym, uwarunkowanym złożonym zestawem kryteriów merytorycznych. Podstawę do realizacji procesu wyboru systemu CRM stanowi specyfikacja wymagań, jakie powinno spełniać rozwiązanie. Na podstawie wyników analizy przedwdrożeniowej należy przede wszystkim określić funkcjonalność systemu oraz jego wymagania techniczne. Dopiero w dalszej kolejności można przystąpić do wstępnej selekcji rynku systemów CRM.

Warto przeprowadzić analizę porównawczą alternatywnych rozwiązań CRM. Powinna ona obejmować takie kroki, jak: zdefiniowanie kryteriów wyboru, określenie ich wagi, ocenę punktową poszczególnych rozwiązań według ustalonych kryteriów oraz podsumowanie wyników.

Podjmując decyzję o wyborze systemu, warto zwrócić uwagę na kilka ważnych czynników, takich jak m.in.:

- funkcjonalność rozwiązania – analizując dostępne na rynku systemy należy uwzględnić bieżące oraz przyszłe potrzeby operacyjne i raportowe firmy. Dotyczy to m.in. ilości transakcji, skalowalności bazy obecnej i przewidywanej liczby użytkowników oraz możliwości uzyskiwania informacji w różnych przekrojach;

- możliwość rozbudowy i rozwoju – z ekonomicznego punktu widzenia korzystniej jest wybrać rozwiązanie skalowalne, gdzie na początku uruchamia się tylko część funkcjonalności, by dalej rozbudowywać je zgodnie z przyjętą strategią firmy;
- bezpieczeństwo systemu – w bazach danych systemu CRM są przechowywane informacje o znaczeniu strategicznym dla organizacji, które muszą być chronione przed niepowołanym dostępem;
- serwis oprogramowania i adaptacja do zmieniających się warunków – system kupowany w danym momencie spełnia oczekiwania i potrzeby firmy, lecz w trakcie eksploatacji, z powodu braku aktualizacji i właściwego wsparcia ze strony dostawcy, może okazać się kłopotliwy w obsłudze, a z czasem bezużyteczny;
- możliwość współpracy z wykorzystywanymi w organizacji aplikacjami;
- możliwość pracy w terenie [Zakr01].

Podstawą pomyślnego wdrożenia systemu CRM jest procesowe podejście do funkcjonowania przedsiębiorstwa, wyspecyfikowanie zmian organizacyjnych i związanych z tym nakładów finansowych na odpowiednio wczesnym etapie realizacji projektu wdrożeniowego oraz precyzyjne zaplanowanie procesu integracji obecnie funkcjonującej platformy informatycznej z systemem CRM [FiJR01]. To, czy wdrożenie systemu CRM zakończy się sukcesem zależy przede wszystkim od stanu organizacyjnego przedsiębiorstwa, a w znacznie mniejszym stopniu od złożoności samej aplikacji. Posiadając długofalową strategię działania i dobrze zdefiniowany model biznesowy, przedsiębiorstwo jest przygotowane do wdrożenia systemu CRM, a i czas wdrożenia w takim podmiocie będzie o wiele krótszy. Systemy CRM stanowią inwestycję w poprawę jakości obsługi klienta, a każda inwestycja ma swój okres zwrotu. Średni czas życia systemu CRM szacuje się na dziesięć do piętnastu lat.

Faktyczna kalkulacja zysków, które może uzyskać firma wprowadzając system CRM jest dosyć trudna. Wiele korzyści uzyskanych dzięki wdrożeniu systemu CRM ma charakter niemierzalny, gdyż pomiar takich elementów, jak np. wizerunek firmy, znajomość marki czy lojalność klienta jest albo niemożliwy, albo utrudniony. Szacując opłacalność inwestycji w CRM korzysta się najczęściej ze wskaźnika ROI. Wskaźnik ten porównuje koszt danej inwestycji z wygenerowanymi przychodami [MaMa04]. Wskaźnik ten powinien uwzględniać takie elementy, jak:

- koszt nabycia i użytkowania systemu CRM,
- zmiany w dotychczas ponoszonych kosztach obsługi klienta,
- zmiany w przychodach ze sprzedaży,

- koszty trudno mierzalne, „miękkie”, które można szacować wykorzystując wskaźniki np. CLV (*Customer Lifetime Value*) czy RFM (*Recency, Frequency, Monetary*) – [Frąc04].

Strategia CRM łączy ludzi, procesy i technologie po to, by poznać klienta i zrozumieć jego potrzeby. Organizacje, którym uda się wdrożyć strategię CRM mogą oczekiwać korzyści w postaci lojalności klienta i długookresowych dochodów. Skuteczne wdrożenie strategii CRM dla wielu organizacji może się jednak okazać trudne, głównie z przyczyny niedostrzegania, że CRM wymaga reinżynierii procesów biznesowych skoncentrowanych na klienta. Nie można także rozpatrywać CRM wyłącznie przez pryzmat technologii. CRM wymaga zintegrowanego i zrównoważonego podejścia do technologii, procesów i ludzi [Porę10]. System CRM wymusza zmianę stylu pracy oraz zmianę przyzwyczajeń pracowników. Obok konieczności wydania określonej kwoty na zakup licencji i wdrożenie systemu, świadomość, że trzeba będzie zacząć inaczej funkcjonować, stanowi istotną barierę skutecznego wdrożenia CRM.

Barieri pojawiające się w trakcie wdrażania systemu klasy CRM w polskich firmach można podzielić na trzy główne kategorie:

- organizacyjne, które wynikają z niskiej świadomości planowania zmian w modelu biznesowym organizacji, konieczności definiowania długofalowych strategii, trudnych do oszacowania kosztów reorganizacji oraz barier natury psychologicznej w związku z niechęcią do wprowadzania zmian;
- techniczne, które są związane głównie z koniecznością integracji systemu CRM ze wszystkimi funkcjonującymi w organizacji systemami informatycznymi, jak również wymaganiami sprzętowymi systemu CRM;
- finansowe, które są następstwem wysokich cen rozwiązań CRM i kosztownej platformy sprzętowej niezbędnej do ich realizacji oraz trudności w określeniu korzyści wynikających z wdrożenia systemu CRM [FiJR01].

Podsumowanie

Tworzenie systemu CRM przypomina budowę domu. Solidny dom potrzebuje solidnych fundamentów. Fundamentami CRM jest filozofia biznesu, umieszczająca klienta w centrum zainteresowania firmy. Dopiero później jest tworzona struktura, którą stanowią strategie biznesowe. Końcowy etap budowy obejmuje prace wykończeniowe, na które składają się odpowiednie modele marketingowe i aplikacje informatyczne. Właściwie zaprojektowany i wdrożony system CRM gromadzi informacje o klientach pochodzące z różnych źródeł, dając pełny i ujednolicony obraz każdego klienta w czasie rzeczywistym. Wdrożenie CRM należy rozpocząć od opracowania strategii, zaprojektowania procesów

biznesowych i procedur pracy. Konieczne jest dostosowanie wewnętrznej organizacji firmy, przeszkolenie i odpowiednie motywowanie pracowników oraz wybór i implementacja odpowiedniej technologii. Skuteczne wdrożenie systemu CRM jest uwarunkowane wieloma czynnikami. Kluczowymi elementami sukcesu są: ludzie, procesy i technologie.

Literatura

- [Buch10] Buchnowska D.: *Wybór systemu wspierającego zarządzanie relacjami z klientami*. W: *Komputerowo zintegrowane zarządzanie*. Red. R. Knosala. Oficyna Wydawnicza Polskiego Towarzystwa Zarządzania Produkcją, Opole 2010.
- [CRMw09] *CRM w przedsiębiorstwie*, <http://www.biznesplany.biz/2009/08/crm-w-przedsiębiorstwie/> [dostęp: 29.09.2011].
- [FiJR01] Filemonowicz K., Jędrzejek Cz., Radziulis P.: *Organizacyjno-finansowe bariery przy wdrażaniu systemów CRM w Polsce*, http://www.ploug.org.pl/konf_01/materialy/pdf/Filemonowicz_jedrzejek_radziulis.pdf [dostęp: 19.10.2011].
- [Fręc04] Frąckiewicz E.: *Krytyczne spojrzenie na Customer Relationship Management*, <http://mikro.univ.szczecin.pl/bp/pdf/39/8.pdf> [dostęp: 25.09.2011].
- [Kost02] Kostecki M.J.: *CRM krok po kroku, czyli od znaczenia do wdrożenia*, http://masterplan.com.pl/publikacje/163_crm.php [dostęp: 15.10.2011].
- [Kupe10] Kupeć K.: *Jak skorzystać na CRM-owej rewolucji*, <http://www.e-marketing.pl/artyk/artyk70.php> [dostęp: 8.10.2011].
- [MaMa04] Mazur A., Mazur D.: *Jak wdrożyć CRM w małej i średniej firmie*. Madar, Zabrze 2004.
- [Porę02] Porębska-Miąc T.: *System CRM jako narzędzie marketingu partnerskiego*. W: *Systemy Wspomagania Organizacji SWO'2002*. Red. T. Porębska-Miąc, H. Sroka. Wydawnictwo AE, Katowice 2002.
- [Porę10] Porębska-Miąc T.: *Rola technologii, procesów i ludzi w sukcesie strategii CRM*. W: *Systemy Wspomagania Organizacji SWO'2010*. Red. T. Porębska-Miąc, H. Sroka. Wydawnictwo AE, Katowice 2010.
- [Prze07] *Przekonanie kierownictwa do CRM*, <http://www.portalmcr.pl/2007/07/16/jak-przekonac-kierownictwo-firmy-do-zakupu-systemu-crm/> [dostęp: 9.11.2011].
- [StSS02] Stachowicz-Stanusch A., Stanusch M.: *Zanim zaczniesz wdrożenie CRM*, <http://www.crm.pl/druk.php?kategoria=9&rodzaj=&id=195> [dostęp: 16.10.2011].
- [WąSt08] Wątróbski J., Stolarska M.: *Dobór systemów CRM w organizacji wirtualnej – aspekt modelowy*, http://pan.wi.ps.pl/pliki/MetInfStos/200802/MetInfStos_2008_02_Art_13.pdf [dostęp: 12.10.2011].
- [Zakr01] Zakrzewski P.: *Jak się przygotować do wdrożenia CRM?* „Modern Marketing” 2001, nr 7.

- [Ziem01] Ziemia E.: *Modelowanie i projektowanie zintegrowanych systemów informatycznych zarządzania*. W: *Zintegrowane systemy informatyczne w zarządzaniu*. Red. C.M. Olszak, H. Sroka. Wydawnictwo AE, Katowice 2001.

DESIGNING AND IMPLEMENTATION OF CUSTOMER RELATIONSHIP MANAGEMENT (CRM) SYSTEMS

Summary

To build a CRM system is like to build a real building. Solid building needs a solid foundation. This foundation is the customer-centric business philosophy. Then we build the structure based on this foundation – business strategies. Then are the detailed finishes of building – such as marketing models and applications of information technology. True designed and implemented CRM system brings together information from all data sources within an organization to give one, holistic view of each customer in real time, makes information available throughout organization and supports all enterprise business processes. The implementation of CRM involves both software applications and a change in the way the organization operates. The organization must establish clear goals and objectives, identify the processes that need to be in place to achieve those goals and implement the communication and training required for employees to act in support of the desired objectives. There are many factors that could influence the success of CRM implementation. The three key elements are: people, process and technology.

Anna Sołtysik-Piorunkiewicz

PROJEKTOWANIE LOGISTYCZNYCH SYSTEMÓW INFORMATYCZNYCH

Wprowadzenie

Współcześnie realizowana logistyka opiera się na technologiach informatycznych i bez nich byłaby niewydolna. Coraz szerszy dostęp do technologii informatycznych powoduje, że w logistyce zachodzą obecnie głębokie zmiany. Logistyka wkroczyła we wszystkie obszary funkcjonalne przedsiębiorstwa i dawno wyzwoliła się z popularnego kojarzenia jej tylko z transportem. Jedną z przyczyn tego stanu rzeczy jest powszechne stosowanie zintegrowanych systemów informatycznych (ZSI), coraz częściej działających w układzie rozproszonym, pozwalającym na zarządzanie logistyką zarówno wielkich, jak i małych przedsiębiorstw.

Na rynku informatycznym nie istnieje jedno zdecydowanie dobre rozwiązanie systemowe, które zapewniłoby kompleksową obsługę informatyczną logistyki przedsiębiorstwa w stopniu je zadawalającym. Zawsze zachodzi konieczność wykorzystywania wielu różnych rozwiązań informatycznych wzajemnie się uzupełniających, które na zasadzie synergii wypełniają wszystkie potrzeby informacyjne przedsiębiorstwa, umożliwiające sprawną realizację logistyki.

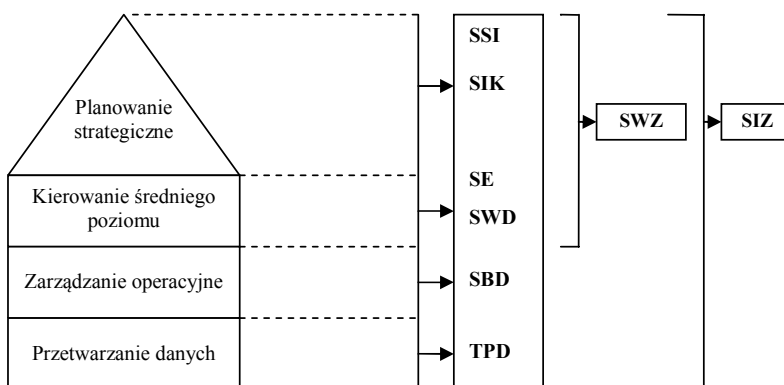
Należą do nich systemy informatyczne zaliczające się umownie do określonych grup tematycznych, takich jak: CRM, DSS, MRP, MRP II, WSM i inne. Wiele funkcji związanych z realizacją procesów logistycznych jest bardzo często oferowanych w postaci systemu ERP. Systemy ERP dzięki zapewnieniom integracji między modułami pozwalają na wymianę danych pomiędzy nimi. Obok systemów ERP osobną klasę systemów dla logistyki stanowią rozwiązania klasy SCM.

1. Zakres systemów informatycznych zarządzania w logistyce

Zintegrowany system informatyczny klasy ERP jest z natury systemem planistycznym, jego szeroka funkcjonalność operacyjna sprowadza się do czynników zarządzających na różnych szczeblach zarządzania przedsiębiorstwem, są to m.in.:

- informacje niezbędne do osiągnięcia celów strategicznych, planowania polityki i podejmowania decyzji na szczeblu zarządu przedsiębiorstwa,
- informacje niezbędne do planowania i podejmowania decyzji na szczeblu średniego kierownictwa,
- informacje niezbędne do planowania operacyjnego i kontroli,
- informacje niezbędne do przetworzenia zamówień, obsługi transakcji itp. [Kis00, s. 208].

Relacje systemów informatycznych zarządzania w stosunku do szczebli decyzyjnych przedstawia rys. 1.



Rys. 1. Relacje SIZ w stosunku do szczebli decyzyjnych zarządzania

Źródło: [Chm96, s. 40].

Systemy te wyrosły z praktycznych doświadczeń stosowania modeli ilościowych w zarządzaniu. W tradycyjnym podejściu do SIZ wyróżnia się:

- systemy trakcyjne przetwarzania danych TPD, zwane w praktyce systemami elektronicznego przetwarzania danych EPD,
- systemy informacyjne zarządzania SBD,
- systemy wspomaganie decyzji SWD,
- systemy eksperckie SE,

- systemy informowania/wspomagania kierownictwa SIK,
- systemy sztucznej inteligencji SSI,
- systemy wspomaganie zarządzania SWZ [Kis03, s. 86].

Głównym celem systemów jest dostarczanie standardowej informacji dla decydentów i menadżerów, zwłaszcza zwiększanie wydajności działania szczebla taktycznego i strategicznego. Każdy z wymienionych standardów stanowi następną fazę rozwoju. Opierając się na literaturze, można przedstawić typowe zastosowania SWD w zarządzaniu logistycznym (tab. 1).

Tabela 1

Zastosowanie SIZ w zarządzaniu logistycznym

Działalność kierownicza na poziomie	Typowe zadania decyzyjne	Niezbędne wsparcie
Operacyjnym	Należności, zobowiązania, dostarczenie zamówień, ewidencja, rejestracja, sortowanie, wyszukiwanie, techniczne przygotowanie produkcji, gospodarka magazynowa, zabezpieczenie pokrycia dostaw, zakup oprogramowania, rynek obligacji, zapewnienie powierzchni magazynowej	SBD
Taktycznym	Analiza budżetu, prognozy krótkoterminowe, analizy produkcji lub sprzedaży, ocena możliwości kredytowej, opracowanie systemu wynagrodzeń, harmonogramowanie projektów, negocjacje, rekrutacja kadry kierowniczej, zakup sprzętu komputerowego, ustalenie poziomu produkcji lub zapasów, przewidywanie popytu	SWD, SE, SIK
Strategicznym	Zarządzanie finansami (inwestycje), lokalizacja hurtowni lub filii, systemy dystrybucji, planowanie nowych produktów, planowanie kontroli jakości, planowanie odszkodowań, rozwój nowych technologii, planowanie badań i rozwoju, analiza rynku kapitałowego, ustalenie poziomu obsługi klienta	SIK, SE, SSI

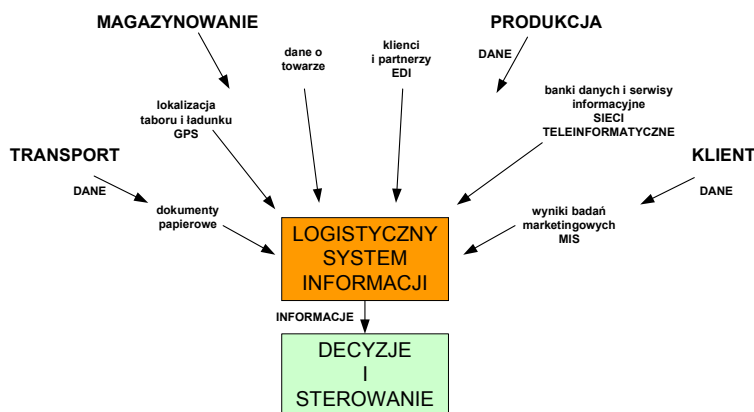
Źródło: [PIP03, s. 87].

2. Logistyczny system informacyjny

Komentarz [v1]: wstawiono tytuł rozdziału

Logistyczny system informacyjny to system zapewniający ciągły dostęp do aktualnych i prawdziwych informacji w łańcuchu dostaw.

Obejmuje zasoby informacyjne oraz elementy umożliwiające ich zasilenie, utrzymanie i dostarczenie użytkownikowi. Do elementów tych zalicza się dostawców i odbiorców informacji oraz techniczno-organizacyjne środki zbierania, komunikacji, przetwarzania, czyli gromadzenia, przechowywania, filtrowania, udostępniania oraz ochrony danych i informacji.



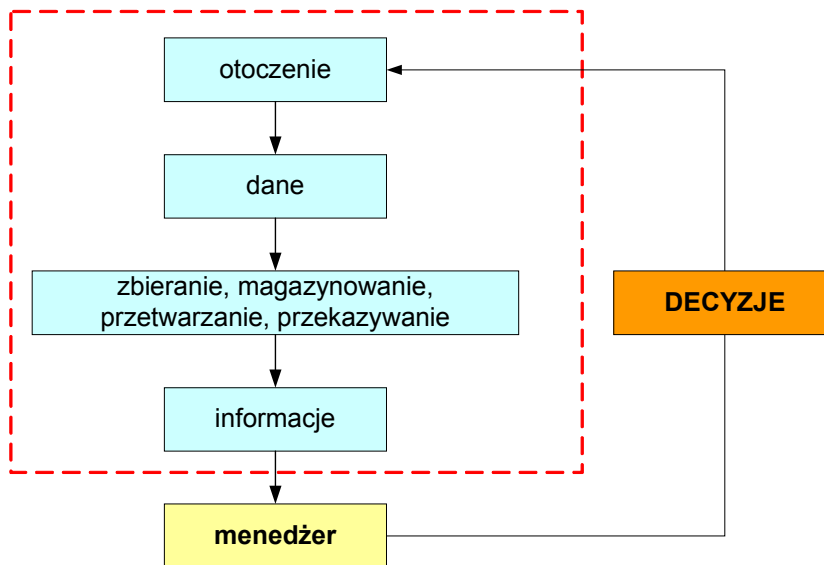
Rys. 2. Logistyczny system informacyjny

Źródło: Na podstawie: [Goł02, s. 176].

Komentarz [v2]: wstawiono rysunek

Podstawowym środkiem technicznym logistycznego systemu informacji są wszelkiego rodzaju sieci teleinformatyczne, wykorzystujące różne media przesyłowe przewodowe i bezprzewodowe. Zastosowanie mediów transmisyjnych stanowi podstawę realizacji przepływów informacyjnych, a tym samym dobrze zaprojektowana i działająca sieć komputerowa jest warunkiem koniecznym logistycznego systemu informacji. Jednocześnie dobrze i stabilnie działająca sieć komputerowa jest podstawowym czynnikiem zapewniającym działanie ADC i EDI, a także pozwala na wykorzystanie pozostałych technik i narzędzi (np. GPS), które służą nie tylko pozyskiwaniu, ale także wymianie danych i ich przetwarzaniu na zewnątrz, np. do klientów lub kontrahentów.

Do najważniejszych funkcji projektowanego systemu informacyjnego logistyki zalicza się pozyskiwanie danych, ich gromadzenie i przetwarzanie w celu nadania im wartości informacyjnej oraz udostępnianie tych informacji menadżerom logistyki. Model projektowanego systemu informacji logistycznej przedstawia rys. 3.



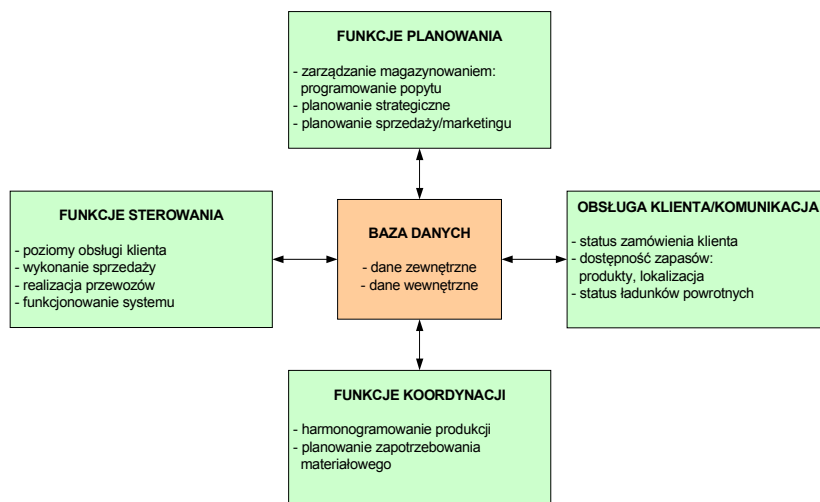
Rys. 3. Model systemu informacji logistycznej

Komentarz [v3]: wstawiono rysunek

Można wymienić wiele specyficznych funkcji, które powinny być realizowane przez idealny system informacji logistycznej, z których najważniejsze to:

- funkcja obsługi klienta i komunikacji,
- funkcja planowania i kontroli,
- funkcja koordynacyjna [Kis00, s. 209].

Szczegółowo funkcje systemu informacyjnego logistyki przedstawia rys. 4.



Rys. 4. Funkcje systemu informacyjnego logistyki

Komentarz [v4]: wstawiono rysunek

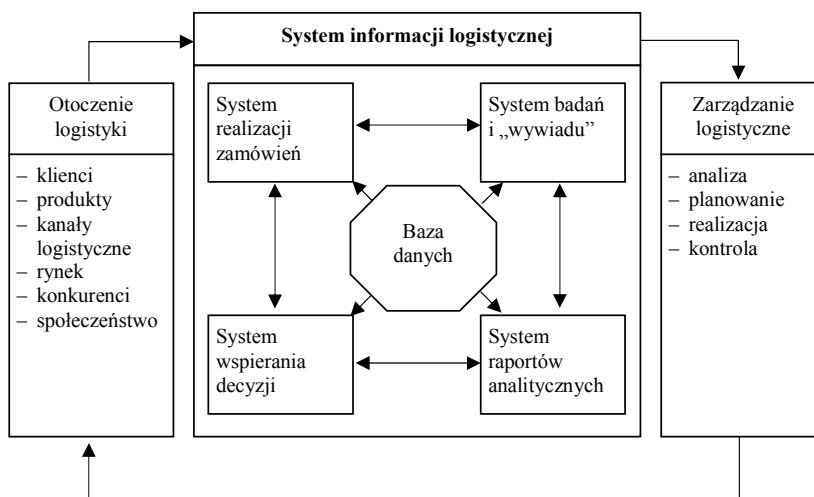
W systemie informacji logistycznej występują cztery zasadnicze podsystemy (rys. 5), które powinny zapewnić właściwe podejmowanie decyzji w sferze zarządzania oraz sprawne funkcjonowanie systemu logistycznego.

System realizacji zamówień obejmuje ciąg czynności związanych z przyjmowaniem zamówień od klientów, przygotowaniem tych zamówień do realizacji, przygotowaniem wysyłki i realizacją dostawy. Czynności te w formie skomputeryzowanej przyjmują postać podstawowych systemów transakcyjnych firmy.

System badań i wywiadu służy obserwacji otoczenia procesów logistycznych w zakresie:

- otoczenia zewnętrznego w skali makroekonomicznej,
- otoczenia w postaci grupy firm współpracujących w ramach wspólnych kanałów logistycznych,
- otoczenia procesów logistycznych w firmie.

Komentarz [v5]: poprawione pierwsze litery na małe



Rys. 5. System informacji logistycznej

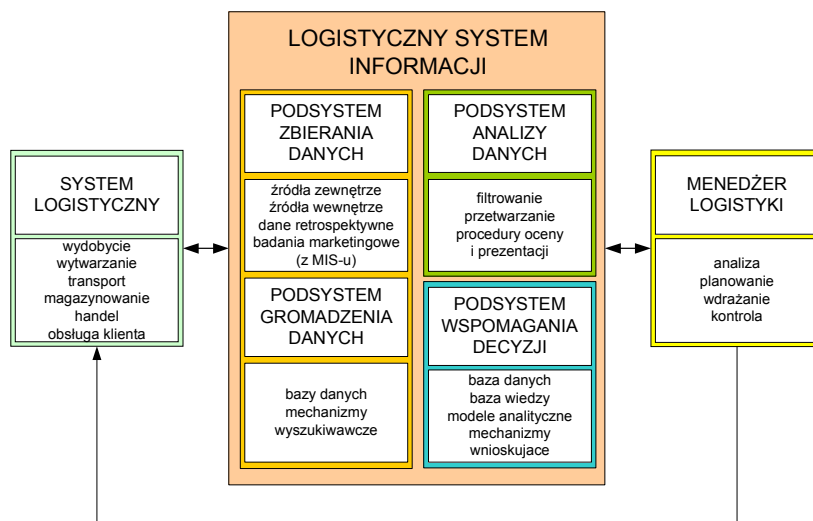
Źródło: Na podstawie: [CoBaLa02, s. 84].

Sposoby badania otoczenia mogą być bardzo różnorodne, począwszy od przypadkowej styczności z pożądanymi informacjami do sformalizowanych badań o określonym celu. Badania te mogą mieć charakter nieregularny lub ciągły. Istotną częścią tego podsystemu są prace prognostyczne, zwłaszcza w sferze prognozowania popytu.

Systemy wspomaganie decyzji są interaktywnymi systemami komputerowymi, które dostarczają danych i modeli analitycznych w celu rozwiązania zagadnień z wieloma trudno definiowalnymi zmiennymi. Szczególną rolę integracyjną należy przypisać bazie danych, której zbiory muszą być wykorzystywane w różnego typu analizach. Występują tu najczęściej takie rodzaje zbiorów, jak:

- zbiory podstawowe, czyli dane dla metod analitycznych,
- czynniki krytyczne określające zakres i główne zasady podejmowania decyzji,
- parametry i polityka regulacyjna precyzująca zasadnicze elementy polityki logistycznej,
- zbiory rozwiązań, czyli rezultaty analityczne stale porównywane z założeniami na przyszłość [Kis00, s. 206].

Komentarz [v6]: poprawiono słowo Systemy



Rys. 6. Funkcjonalny diagram logistycznego systemu informacji

Źródło: Na podstawie: [Goł02, s. 179].

3. Oczekiwania przedsiębiorstw odnośnie do wykorzystywania informatyki w logistyce

Podstawową kwestią przy identyfikacji potrzeb odnośnie do wykorzystania ZSI (typu ERP) jest to, że kadra zarządzająca bardzo często nie jest w stanie określić faktycznych problemów związanych z zarządzaniem przez nią przedsiębiorstwami. Panuje ogólne przekonanie, że źródłem problemów przedsiębiorstwa jest tylko i wyłącznie sytuacja zewnętrzna. W trakcie kontaktów z menadżerami przedsiębiorstw niezwykle często pojawiają się uwagi odnośnie do polityki rządu, zachowania banków i urzędów skarbowych, a także klientów, którzy w swojej nieświadomości wybierają ofertę konkurencyjności, zamiast „naszej najlepszej”. Niewiele przedsiębiorstw dostrzega potrzebę reorganizacji i szukania nowych możliwości we własnej organizacji.

Istnieją jednak firmy, które starają się zmienić sposób działania i szukać nowych możliwości w zarządzaniu. Firmy te najczęściej wskazują na następujące obszary wymagające usprawnienia:

- zwiększenie efektywności działu sprzedaży,
- obniżenie kosztów produkcji,

- zwiększenie kontroli nad kontraktami,
- zwiększenie kontroli nad pracownikami (szczególnie w firmach handlowych),
- obniżenie kosztów magazynowania [Maj08, s. 48].

4. Inteligentne łańcuchy dostaw, czyli nowoczesna informatyka w logistyce

Informacja i sprawność jej przekazu to współczesny klucz do organizacji systemów logistycznych. Nowoczesna i efektywna logistyka nie istnieje bez teleinformatyki [Zloch10].

Według specjalistów, efektywna logistyka jest obecnie jednym z kluczowych warunków powodzenia przedsiębiorstwa na rynku. W myśl marketingowej zasady 4P: Product, Place, Price, Promotion (produkt, miejsce, cena, promocja), o powodzeniu produktu na rynku decyduje nie tylko właściwa promocja, lecz także jego cechy, jakość, cena oraz dostępność towaru w określonym czasie i miejscu. W związku z tym firmy logistyczne oraz wewnętrzne działy zaopatrzenia i dystrybucji poszczególnych przedsiębiorstw odgrywają tak ważną rolę we współczesnym biznesie.

Przed nowoczesną logistyką stają nowe wyzwania związane z koniecznością szybkiej adaptacji do zmieniającego się otoczenia, możliwością integracji z wieloma zróżnicowanymi partnerami biznesowymi oraz potrzebą szybkiego zdobycia wartościowych informacji. Przedsiębiorstwa, które będą potrafiły sprostać tym wyzwaniom, uzyskają przewagę nad konkurencją.

Pod hasłem logistyka rozumie się wszystko, co jest związane z wpływaniem dóbr i usług do organizacji, procesami dotyczącymi tego, co dzieje się w organizacji, jak choćby gospodarka magazynowa oraz aspektami dotyczącymi wyprowadzania dóbr i usług z organizacji. Oferta firm dostarczających systemy informatyczne dla logistyki jest bogata. Dostępne są na przykład rozwiązania dotyczące zarządzania materiałami, sprzedaży i dystrybucji, utrzymania ruchu, remontów czy planowania produkcji.

Wśród rozwiązań teleinformatycznych dedykowanych procesom logistycznym można na przykład wyróżnić oprogramowanie dedykowane obsłudze procesu wejścia – zakupu towarów, usług i materiałów. Umożliwia ono zarządzanie procesami zakupowymi, poszukiwanie partnerów biznesowych, dostawców materiałów, towarów czy usług. Za obsługę logistyki w organizacji, czyli wewnętrzne przemieszczenia dóbr obejmujące usługi transportowe, planowanie i zarządzanie transportem oraz zarządzanie środkami transportu, są odpowiedzialne rozwiązania z kategorii Transportation Management. Osobne aplikacje służą za-

rzządzaniu magazynami, czyli obsłudze logistyki magazynowej (co wchodzi do magazynu, na jaką półkę, jakie są strategie pobierania towarów z magazynu, ich pakowania czy kompletacji materiałów).

W ciągu ostatnich kilku lat coraz więcej firm decyduje się, aby ich procesy logistyczne były realizowane w centrach logistycznych. Przesłanki dla takich decyzji wynikają z wielu przyczyn. Traktowane jest to jako szansa obniżenia kosztów (forma outsourcingu), możliwość skoncentrowania się na swojej podstawowej działalności, jak również efektywna droga do podniesienia poziomu zadowolenia klienta (np. poprzez zmniejszenie czasu przetwarzania zamówienia i wysyłki) – [Złoch10, s. 2].

Dla realizacji tych celów operatorzy logistyczni szukają nowych rozwiązań pozwalających obniżyć jednostkowy koszt realizacji pojedynczej transakcji, efektywniej wykorzystać powierzchnię magazynową i posiadany potencjał. Powszechną praktyką stało się wykorzystywanie dokumentów przesyłanych drogą elektroniczną pomiędzy firmami w ramach łańcucha dostaw faktur. Przykładowymi dokumentami są np. zamówienia, potwierdzenia dostaw czy też faktury. Czas przesyłania i przetwarzania przesyłanych w ten sposób danych jest krótszy, a liczba występujących błędów ograniczona.

Coraz częściej firmy podporządkowują wdrożenie nowoczesnego logistycznego systemu informacyjnego zarządzaniu jakością swoich produktów. Jednym ze sposobów utrzymywania wysokiej jakości jest identyfikowalność, zapewniająca szybkie dotarcie do źródła defektu. Według Złocha jest ona zdolnością do odtworzenia informacji o historii produktu, łącznie z informacjami o pochodzeniu surowców czy elementów wykorzystywanych do jego produkcji [Złoch10, s. 4], głównie dzięki zastosowaniu technologii RFID, EDI i ADC oraz odpowiednich czytników kodów, kolektorów danych itp. W systemie logistycznym identyfikowalność pozwala na śledzenie produkcji na każdym jej etapie oraz odnosi się również do innych etapów łańcucha dostaw pełnej identyfikacji numeru partii i serii, dotyczącej surowców czy podzespołów.

Identyfikowalność rozciąga się znacznie ponad jedno przedsiębiorstwo, przechodzi przez pełen proces logistyczny, przede wszystkim dostawców i poddostawców przedsiębiorstwa, a idąc wstecz, można dojść do pierwszego elementu (ogniwa łańcucha). Nadzór narzuca się wszystkim firmom w pełnym łańcuchu logistycznym. W konsekwencji analiz jakościowych zdarza się wycofanie całej partii materiałów czy surowców, a nawet wyeliminowanie dostawcy surowców niedotrzymujących standardów jakościowych. Rolą systemu jest więc gromadzenie wszystkich informacji służących do znalezienia przyczyny defektu.

5. Charakterystyka systemów klasy ERP

Obecnie ZSI (klasy ERP) obejmuje swym zasięgiem funkcjonalnym wszystkie dziedziny działalności przedsiębiorstwa. Jak już wspomniano, jest to efektem naturalnego rozwoju dotychczas oferowanych rozwiązań informatycznych zawierających algorytmy metody MRP II, a której funkcjonowanie nie jest możliwe bez właściwego systemu informatycznego. Wdrożenie i użytkowanie metody MRP II w planowaniu działalności przedsiębiorstwa jest zagadnieniem dość złożonym i niejednoznacznym. Wdrożenie metody MRP II w logistyce przedsiębiorstw jest związane z pozyskiwaniem świadomości potrzeby jej używania, wyborem, zakupem i umiarkowaniem oprogramowania dostosowanego funkcjonalnie do specyfiki prowadzonego biznesu oraz, co jest najważniejsze, nauczeniem się korzystania z jej zalet. Złe oprogramowanie, a ściślej mówiąc, źle przeprowadzone wdrożenie ZSI w zakresie metody MRP II może skutecznie zniweczyć projekt jej wprowadzenia do powszechnego wykorzystania w przedsiębiorstwie, co jednak nie oznacza, że dobre oprogramowanie, nawet dobrze wdrożone, zapewni, że MRP II będzie powszechnie funkcjonować [Maj08, s. 55].

Z nowoczesną logistyką jest związane pojęcie łańcucha dostaw. Łańcuch dostaw to przede wszystkim planowanie, kontrolowanie i koordynowanie przepływu surowców i gotowych wyrobów od dostawców do odbiorców. Coraz częściej w przedsiębiorstwach są wykorzystywane systemy klasy ERP, które zapewniają tym procesom kompleksowość i ciągłość.

Skuteczna integracja wielu złożonych procesów zachodzących w łańcuchu dostaw czy innych działań firmy jest możliwa ze względu na budowę systemów ERP. Oprogramowanie to jest platformą, na której są umieszczone aplikacje usprawniające pracę w poszczególnych działach firmy, tj. produkcji, finansach czy transporcie i spedycji. Dzięki wspólnej „podstawie” informacje są płynnie przekazywane z jednego modułu, np. obsługującego proces dostawy, do innego wykorzystywanego w zarządzaniu magazynem czy też księgującego wszystkie transakcje oraz ułatwiającego rozliczanie z kontrahentami firmy.

Systemy ERP są wykorzystywane m.in. do rejestracji przyjęć i wydań towaru, zarządzania poziomem towarów, powierzchnią czy ruchem w magazynie oraz stanowią na bieżąco aktualizowaną bazę danych. Baza ta zawiera informacje dotyczące kluczowych procesów logistycznych, takich jak stan magazynu, dostępność poszczególnych substratów czy gotowych produktów. Zgromadzenie wszelkich danych, niezbędnych do sprawnego zarządzania łańcuchem dostaw

w jednej zintegrowanej aplikacji zamiast w kilku różnych programach, przekłada się na dużo szybsze, wygodniejsze i bardziej dokładne wyszukiwanie potrzebnych informacji, co jest niezwykle ważne w przypadku konieczności podjęcia natychmiastowych decyzji.

Systemy pozwalają na zoptymalizowanie stanu magazynów. Automatyczne powiadomienia odpowiednio wcześniej informują pracowników o kończących się zapasach poszczególnych materiałów, niezbędnych do produkcji konkretnych produktów. Zastosowanie takiego „systemu wczesnego ostrzegania” zapobiega wyłączeniu linii produkcyjnej, co wyklucza opóźnienia w dostawach i poważne straty finansowe dla firmy. System ERP przyczynia się także do utrzymania najbardziej efektywnego stanu zaopatrzenia magazynu bez zamrażania nadmiernych środków na składzie, tym samym zwiększając wskaźnik rotacji towarów i obniżając koszty utrzymania zapasów.

Dodatkowo narzędzie na podstawie aktualnych stanów magazynowych i historii zamówień klientów samo oblicza zapotrzebowanie na uzupełnienie poszczególnych towarów w magazynie oraz automatycznie generuje dokumenty zlecające wykonanie odpowiednich działań. Funkcjonalność ta w znacznym stopniu odciąża pracowników centrali firmy. Aplikacje ERP przyczyniają się do utrzymania porządku w kartotekach towarów i ułatwiają inwentaryzację składowanych towarów, ze względu na bezproblemową współpracę z urządzeniami zewnętrznymi, takimi jak czytniki kodów kreskowych, drukarki kodów, kolektory danych itd. Narzędzia te dobrze radzą sobie także w sytuacjach przesunięć magazynowych, które niejednej firmie potrafią przysporzyć wielu trudności odbijających się na tak ważnych działaniach, jak chociażby terminowe dostarczenie transportu do klienta. Jeśli takie przesunięcia występują, aplikacje pomagają zachować ład w dokumentacji, a także pozwalają szybko i precyzyjnie zlokalizować konkretny towar.

System klasy ERP musi się odznaczać kilkoma cechami, które są charakterystyczne dla nowoczesnych systemów informatycznych. Przede wszystkim powinien być:

- otwarty – dający się łatwo integrować z otoczeniem (np. z urządzeniami przenośnymi, zewnętrznymi konsolami, innymi systemami lokalnymi, usługami internetowymi), ponieważ wartość informacji w organizacji jest tym większa, im więcej wiedzy wnosi ona o procesach i zjawiskach zachodzących na zewnątrz firmy;

- przyjazny dla użytkownika – zamiast zmuszać pracownika do uczenia się skomplikowanych procedur obsługi, system powinien stworzyć jego naturalne środowisko pracy i rozwoju;
- elastyczny (ze zdolnością do adaptacji) – powinien dysponować takimi zasobami funkcjonalności, aby wspomagać zmieniające się procesy i praktyki biznesowe;
- innowacyjny – dający użytkownikom możliwość nie tylko działania w ramach wdrożonych procedur, lecz także eksperymentowania, symulacji i tworzenia własnych, nowych praktyk biznesowych, które można modyfikować i dostosowywać do indywidualnych potrzeb;
- stabilny – niezakłócający codziennej pracy przedsiębiorstwa;
- działający w czasie rzeczywistym, tzn. zawsze prezentujący aktualne informacje;
- wspomagający strategię – dający zagregowany obraz zgodności działań z założeniami strategicznymi (np. zapewniający sprawny system pomiaru efektywności działań logistycznych) – [Cie06, s. 74].

6. Charakterystyka systemów SCM

Pomiędzy implementacją systemu planowania zasobów przedsiębiorstwa (ERP) i systemu zarządzania łańcuchem dostaw (SCM) występują istotne różnice. W odróżnieniu od systemów ERP w rozwiązaniach SCM jest wymagany model danych, pozwalający obsługiwać w czasie rzeczywistym dużą liczbę złożonych transakcji. Dotychczas, aby uzyskać kompleksowe rozwiązania, należało zintegrować wyspecjalizowane oprogramowanie z własnym systemem ERP i utworzyć specjalne interfejsy obsługujące zewnętrzne źródła danych. Metoda przynosi rezultaty, ale wiąże się z dużymi kosztami [Maj08, s. 60].

Korzyści wdrożenia systemu SCM są następujące:

- zwiększenie zysku we współpracy z dostawcami;
- usprawnienie obsługi klienta oraz zmniejszenie niedoborów materiałowych;
- zmniejszenie kosztów związanych z transportem materiałów;
- optymalizacja wartości łańcucha dostaw w celu zmniejszenia kosztów oraz zwiększenia zysku;
- zredukowanie kosztów operacyjnych na poziomie przedsiębiorstwa oraz obniżenie kosztów wytworzenia wyrobów gotowych;

- zwiększenie konkurencyjności poprzez zoptymalizowanie przepływu materiałów i towarów oraz obniżenie kosztów magazynowania, a także planowanie przepływów materiałowych;
- uzyskanie przejrzystości łańcucha dostaw, w tym wśród partnerów handlowych;
- sprawna adaptacja przedsiębiorstw do zmieniających się warunków i sytuacji rynkowych z myślą o kliencie.

Wszystkie działania realizowane w systemie SCM można podzielić na cztery podstawowe procesy: planuj, nabądź, wytwórz, dostarcz [OISr03, s. 314].

Proces „planuj” jest związany z planowaniem popytu na wyroby oraz planowaniem pozyskiwania tych wyrobów.

Proces „nabądź” odnosi się do działalności zaopatrzeniowej. Są to takie działania, jak: nabywanie, przyjmowanie i kontrola dostaw, składowanie i wydanie do zużycia oraz zarządzanie infrastrukturą zaopatrzeniową obejmującą certyfikowanie i ocenę jakościową dostawców, logistykę dostaw, kontrakcje i płatności.

Proces „wytwórz” wiąże się z działalnością produkcyjną. Obejmuje zgłaszanie zapotrzebowań i pobieranie materiałów, wytwarzanie i testowanie wyrobów, pakowanie, składowanie i wydawanie wyrobów gotowych oraz zarządzanie infrastrukturą produkcyjną, w skład której wchodzi: zarządzanie zmianami konstrukcyjnymi, wyposażeniem pomocniczym, standardami jakościowymi, planowanie warsztatowe i bilansowanie zdolności produkcyjnych w krótkim czasie.

W procesie „dostarcz” znajdują się działania: zarządzanie zamówieniami klientów (wprowadzanie i obsługa zamówień, tworzenie ofert, konfiguracja produktów itp.), gospodarka magazynowa (pobieranie, pakowanie i konfigurowanie produktów itp.), zarządzanie transportem (spedycja, fracht, import, eksport), zarządzanie infrastrukturą dostaw (określenie zasad biznesowych odnośnie do funkcjonowania kanałów sprzedaży oraz przetwarzania zamówień, zarządzanie dostarczonym zapasem oraz jego jakością). Z usprawnienia łańcucha dostaw można uzyskać liczne korzyści. Jakościowe zmiany uzyskuje się w następujących obszarach:

- aktywne planowanie i reagowanie na popyt,
- usprawnienie procesów w firmie,
- krótszy czas reakcji na zmiany popytu,
- usprawnienie procesu dostaw,
- optymalizacja wykorzystania zasobów,
- redukcja kosztów zapasów i kosztów magazynowych.

Systemy SCM umożliwiają prowadzenie skutecznej współpracy logistycznej w całym łańcuchu tworzenia wartości. Przykładem technologii informatycznej realizującej powyższą ideę jest Internet Transaction Serwer oraz SAP [OISr03, s. 314].

Do komponentów systemów SCM można zaliczyć:

- systemy optymalizacji łańcucha dostaw (Supply Chain Optimization – SCO)
- zarządzanie popytem sieci dostaw (Demand Driven Supply Network – DDSN)
- systemy zarządzania zdarzeniami łańcucha dostaw (Supply Chain Event Management – SCEM),
- systemy zarządzania magazynem (Warehouse Management System – WMS),
- systemy do obsługi kodów kreskowych RFID (Radio Frequency Identification – RFID),
- systemy zarządzania transportem (Transport Management System – TMS)
- systemy zarządzania popytem (Demand Management – DEM)
- systemy zarządzania relacjami z dostawcami (Supplier Relationship Management – SRM).

7. Charakterystyka systemów WMS

Logistyka to także magazyny. Tu są wykorzystywane aplikacje klasy WMS (Warehouse Management System) – programy do zarządzania ruchem produktów w magazynach. Nowoczesny system WMS powinien cechować się dużą elastycznością pozwalającą na dostosowanie do wykorzystania w różnych branżach, takich jak: produkcja, transport i logistyka czy sprzedaż detaliczna i dystrybucja. Systemy WMS powinny charakteryzować się możliwością współpracy z zewnętrznymi urządzeniami i rozwiązaniami stosowanymi w magazynach, takimi jak systemy automatycznej identyfikacji, automatyka magazynowa itp. Dodatkowo powinny posiadać mechanizmy optymalizacji w zakresie wykorzystania przestrzeni magazynowej.

Aplikacje informatyczne pomagają w odciążeniu przestrzeni magazynowych. Oprogramowanie wspomaga firmę w obszarze przeładunku kompletacyjnego w ramach koncepcji ECR (cross-docking). Metoda ta jest stosowana zwłaszcza w firmach dystrybucyjnych, a polega na przeładowywaniu i wysyłce towaru do odbiorcy bezpośrednio po dostarczeniu go do magazynu, bez składowania. Po dostawie towary nie są przechowywane w odpowiednich miejscach składowania w magazynie, ale bezpośrednio z rozładunku trafiają do strefy wydań. Pozwala to obniżyć koszty łańcucha dostaw, jednak pod warunkiem ścisłej synchronizacji przyjmowania dostaw i realizacji wysyłek. Cross-docking za-

pewnia duże oszczędności czasu i powierzchni magazynowej – przyspiesza realizację zamówień klientów, przyczyniając się jednocześnie do powstania oszczędności w wykorzystaniu przestrzeni magazynowej. Rozwiązania, które kiedyś wydały się bardzo drogie i niemożliwe do powszechnego zastosowania, dzięki spadkowi ceny jednostkowej stały się dostępne i są z sukcesem wdrażane. Doskonały przykład stanowią nośniki RFID (Radio Frequency Identification), które pozwalają na kontrolę przepływu towarów w ramach łańcucha dostaw. Ilość danych, która może być zapisana na takim nośniku jest wielokrotnie większa niż w standardowym kodzie kreskowym. Ważne jest, aby system informatyczny wspierający danego operatora potrafił wykorzystać możliwości, jakie otwierają się przy wykorzystaniu takich nośników. Niektórzy producenci oprogramowania mają opracowane gotowe produkty wspierające wykorzystanie technologii RFID.

Innym przykładem zastosowania technologii informatycznych w centrach logistycznych są rozwiązania wspierające procesy poprzez sterowanie głosem, np. systemy typu voice picking. Rozwiązania takie pozwalają na zwiększenie wydajności przy zachowaniu tego samego składu osobowego. Działanie systemu głosowego polega na użyciu mowy jako naturalnego sposobu komunikacji między użytkownikiem a systemem informatycznym. Wszystkie zadania dla pracownika magazynu są przetwarzane na głos w terminalu głosowym, który dostarcza precyzyjnej informacji, jaki produkt i z jakiego miejsca w magazynie należy pobrać. Pracownik również głosowo potwierdza pobranie towaru, co natychmiast zostaje zarejestrowane przez system informatyczny.

Logistyka to także flota samochodów umożliwiająca dostarczanie towarów. Tu bardzo pomocny i coraz powszechniej jest wykorzystywany system GPS (Global Positioning System) – Globalny System Pozycyjny. Służy do określania położenia, czasu oraz prędkości obiektów. Pozwala np. na rejestrowanie i wizualizację tras pojazdów, czasów postojów i przeładunków, kontrole czasu pracy kierowców i wielu innych parametrów pojazdu, w tym np. poziomu paliwa w zbiorniku. Wykorzystując tego typu rozwiązania, można zredukować koszty transportu i usprawnić procesy logistyczne, a jednocześnie zapewnić wyższy poziom bezpieczeństwa komunikacji.

Pierwsze w Polsce systemy klasy WMS zaczęły wprowadzać przed 10 laty duże zachodnie firmy, m.in. Coty Polska, Frantschach Świecie (obecnie Mondi Packaging), Kronopol. Firmy te miały specjalistów z dziedziny logistyki, którzy doceniali potrzebę nowoczesnych rozwiązań logistycznych, niezbędnych na coraz bardziej konkurencyjnym rynku. Oni rozumieli, że duży przepływ materiałów musi być dobrze zarządzany i że bardzo dużo można stracić z powodu źle zorganizowanej logistyki. Niedostarczenie czegoś na czas to olbrzymia strata, którą bardzo trudno nadrobić. Na miejsce niesolidnego dostawcy czeka wielu

Komentarz [v7]: poprawione

innych, dlatego tak ważna jest jakość w logistyce. Zaczęło się od zachodnich firm, ale w ostatnich latach coraz więcej polskich średnich i dużych przedsiębiorstw sięga po systemy WMS. Ich menadżerowie doceniają fakt, że dobrze zorganizowany przepływ materiałów ma znaczenie dla poprawienia pozycji konkurencyjnej.

Zarządy wielu firm często się przekonują, że granicą ich rozwoju nie są ani finanse, ani marketing, ani też zarządzanie. Granicą rozwoju staje się coraz częściej logistyka. Dobrym przykładem średniej wielkości polskiej firmy, która doceniła fakt, że zarządzanie magazynem na odpowiednim poziomie jest warunkiem konkurencyjności jest firma PHUP Andrzej Szeszycki. Ta hurtownia spożywcza, która zajmuje znaczącą pozycję na rynku Polski Zachodniej, po wprowadzeniu systemu klasy WMS zwiększyła wydajność pracy magazynu ponad dwukrotnie.

Z dużych polskich firm można wymienić TZMO (Toruńskie Zakłady Materiałów Opatunkowych), gdzie nowatorski system logistyczny SKM (System Kierowania Magazynem) wraz z radiowym System Kierowania Wózkami (SKW) sprawił, iż z powodzeniem konkuruje ona z potężnymi koncernami międzynarodowymi [Złoch10, s. 3].

Oprogramowanie klasy WMS wspiera przedsiębiorstwa produkcyjne i dystrybucyjne w zakresie: organizacji, kontroli, sterowania, procesami związanymi z fizycznym przepływem surowców i towarów.

Systemy WMS zakresem działania obejmują: całość operacji związanych z dystrybucją wewnętrzną firm, począwszy od przyjęcia towaru i jego kontrolę, poprzez składowanie w magazynie po wysyłkę do klienta. W systemach jest odzwierciedlona struktura całego systemu składowania z możliwością odwzorowania najmniejszego miejsca składowego. Zasztyte algorytmy optymalizacyjne prowadzą magazynierów „krok po kroku” w zakresie wykonywanych zadań, przyczyniając się w ten sposób do zmniejszania ilości pomyłek i zwiększania wydajności magazynów.

Główne moduły systemów WMS to:

- obsługa dostaw,
- kontrola wejściowa,
- planowanie i kompletacja,
- obsługa wysyłek, załadunek,
- kontrola wyjściowa,
- wspomaganie spedycji,
- przesunięcia wewnątrzmagazynowe,
- inwentaryzacja,
- konfekcjonowanie,
- raporty.

W realizacji procesów logistycznych są wykorzystywane:

- radiowa technika przesyłania danych (standardy WiFi 802.11 a/b/g/n);
- urządzenia mobilne (ręczne i wózkowe terminale przenośne);
- kody kreskowe i RFID [Maj08, s. 231].

W wybranych projektach systemy WMS współpracują również z elementami automatyki przemysłowej (przenośniki, regały automatyczne etc.).

8. Systemy informacji wspierające zarządzanie procesami logistycznymi

Ogromną rolę odgrywają również systemy elektronicznego monitorowania oraz śledzenia ruchu taboru i przesyłek. Dzięki nim firma logistyczna może swobodnie i racjonalnie koordynować pracę w swoim przedsiębiorstwie. Z wykorzystaniem telematyki dyspozytor przez cały czas obserwuje drogę przesyłki i na tej podstawie wybiera odpowiednie działanie. Telematyka łączy w sobie informacje i techniki komunikacyjne. Oznacza to zastosowanie zaawansowanych technologii telekomunikacyjnych i informatycznych w określonej dziedzinie, obecnie istnieje wiele zastosowań telematyki w szeroko rozumianym transporcie towarów.

Za technologie telematyczne uznaje się:

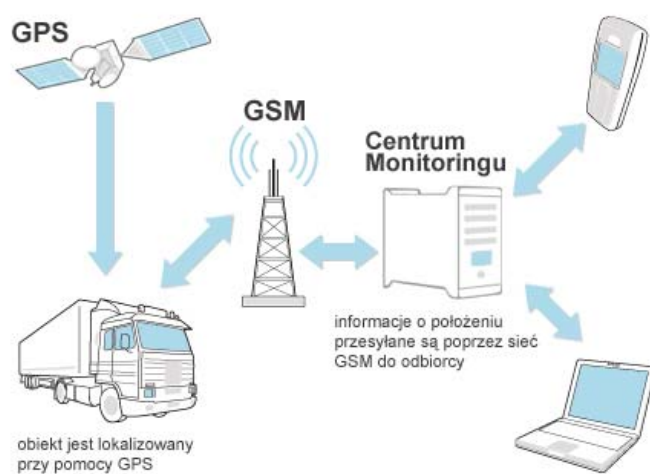
- systemy pozycjonowania satelitarnego GPS,
- systemy łączności radiowej GSM,
- systemy łączności dedykowanej na bliskie odległości DSRC,
- geograficzne bazy danych CIS,
- drogowe bazy danych, tablice o zmiennej treści VIS,
- inteligentne karty (*smart cards*),
- systemy automatycznej lokalizacji pojazdów AVLS,
- systemy nadzoru za pomocą kamer,
- systemy zarządzania ruchem ulicznym.

Celem zastosowania telematyki w transporcie jako usługi jest rozwiązanie, które w znacznym stopniu ułatwia optymalizację i integrację różnych rodzajów transportu. Rynek usług transportowo-logistycznych, na którym funkcjonują różne podmioty, coraz częściej docenia znaczenie szybkiej i rzetelnej wymiany informacji w nasilającej się konkurencji.

Najczęściej wykorzystywanym systemem do monitorowania jest system GPS, który umożliwia wyznaczenie trójwymiarowej pozycji odbiornika (rys. 8). Poza dokładnym wyznaczeniem miejsca, możliwe jest także realizowanie dodatkowych funkcji, np. odbiór sygnałów alarmowych, rejestracja wybranych parametrów dotyczących przestrzeni ładunkowej, temperatury w chłodni, ciśnienie w zbiorniku cysterny.

Projekty systemów informacji są stosowane do wykrywania, śledzenia oraz komunikacji pojazdów w całym łańcuchu transportowym. Należą do nich:

- system „IVHS” – jest to system łączności drogowej (Intelligent Vehicle Highway System), który umożliwia zarządzanie ruchem, zwiększa bezpieczeństwo i efektywność jazdy pojazdu. W całym systemie dąży się do automatyzacji ruchu. Do takich systemów należą systemy sterowania ruchem, sieć kolumn wzywających pomoc, system automatycznego pobierania opłat oraz system sygnalizacji dynamicznej,
- system „EUTELTRACS” – jest to telekomunikacyjno-nawigacyjny system satelitarny, który pozwala sprawnie zarządzać flotą. System ten jest głównie kierowany do dyspozytorów, zapewniając im łączność z kierowcami. Wysyłane informacje mogą być dwustronne, pomiędzy odbiorcą a nadawcą, co pozwala zoptymalizować ich przepływ,
- system monitoringu „MULTITRACK” – celem systemu jest monitorowanie miejsca pobytu i stanu ładunków poprzez cały łańcuch logistyczny. Pozwala to na efektywne zarządzanie zasobami przez dyspozytora pojazdów, poprzez udoskonalenie procesu zarządzania. Wskazany projekt zastosował łączność satelitarną do pozycjonowania ładunków, a także system czujników do wykrywania stanu ładunków,



Rys. 7. Modelowe rozwiązanie lokalizacyjne pojazdu

Wszystkie prace koncepcyjne nad systemami telematycznymi powinny iść w kierunku dalszej automatyzacji procesów zarządzania i usprawniania przepływu informacji za pomocą technologii mobilnych. Na polskim rynku powstało już wiele firm oferujących swe usługi w systemach lokalizacji pojazdów oraz usprawniających prowadzenie taboru, np. „Data System Group”, „Traskocars system”, „Elfro”, „Super Visior”.

Podsumowanie

Scharakteryzowane powyżej systemy kategorii ERP, SCM, WSM wspomagające działania logistyczne mają dużo wspólnego, ale dzieli je też dużo różnic wynikających z ich specyfikacji. Nie ma panaceum w postaci jednego dobrego rozwiązania informatycznego na wszystkie problemy przedsiębiorstwa. Teoretycznie można takie rozwiązanie zaprojektować, a nawet stworzyć, jednak problem wdrożeniowy tego zagadnienia urośnie do skali działań być może kilkuletnich. Systemy klasy ERP już się zadomowiły w wielu przedsiębiorstwach, mimo że stopień wykorzystania ich potencjału nie jest zadawalający. ZSI w większości wypadków prawidłowo wspomagają sfery zarządzania przedsiębiorstwem w rozumieniu obsługi finansów, personelu, kontaktów z dostawcami i odbiorcami oraz gospodarki materiałami rozumianymi jako wspomaganie logistyczne. W rzeczywistości jednak to systemy typu WMS wspomagają logistykę, a systemy, o których się mówi, że są klasy SCM, będą wspierać logistykę wtedy, gdy pozwolą na fizyczną kontrolę przepływu materiałowego, a nie na kontrolę wirtualną, polegającą na manualnym rejestrowaniu danych z dokumentów sprzedaży i zakupów. W logistyce efektywnie wspomaganą kompetentnym oprogramowaniem można uzyskać więcej, wykorzystując system typu WMS niż ERP. Nasuwa się myśl, aby systemowi klasy ERP pozostawić w logistyce sferę zarządzania zapasami logistycznymi, a systemowi WMS oddać zarządzanie zapasami fizycznymi. Niektórzy dostawcy ERP postulat ten wprowadzają już w życie.

Dbając natomiast o interesy ostatecznego odbiorcy, czyli użytkownika informatyki, wspierającej go w działalności logistycznej, należy zapewnić mu możliwość integracji tych dwóch klas systemów. Nadal istnieją jednakże duże obawy, co do współpracy dwóch różnych systemów informatycznych realizujących ten sam cel.

Literatura

- [Chm96, s. 40] Chmielarz W.: *Systemy informatyczne wspomagające zarządzanie*. Elipsa, Warszawa 1996.
- [Cie06] Ciesielski M.: *Instrumenty Zarządzania logistycznego*. PWE, Warszawa 2006.
- [CoBaLa02] Coyle J.J., Bardi E.J., Langley C.J.: *Zarządzanie logistyczne*. PWE, Warszawa 2002.
- [Goł02] Gołomska E.: *Kompendium wiedzy o logistyce*. Wydawnictwo Naukowe PWN, Warszawa 2002.
- [Kis00] Kisperska-Moroń D.: *Wpływ tendencji integracyjnych na rozwój zarządzania logistycznego*. Wydawnictwo Naukowe AE, Katowice 2000.
- [Kis03] Kiperska-Moroń D.: *Zarządzanie logistyczne w firmach usługowych*. Wydawnictwo Naukowe AE, Katowice 2003.
- [Maj08] Majewski J.: *Informatyka dla logistyków*. Instytut Logistyki i Magazynowania, Poznań 2008.
- [OlSr03] Olszak C.M., Sroka H. (red.): *Informatyka w zarządzaniu*. Wydawnictwo Naukowe AE, Katowice 2003.
- [PiPi03] Plączek E., Piniński R.: *Zarządzanie logistyczne w firmach usługowych*. Wydawnictwo AE, Katowice 2003.
- [Złoch] Złoch M.: *Inteligentne łańcuchy, czyli nowoczesna informatyka w logistyce, 2010*, http://www.logistyka.wnp.pl/inteligentne-łańcuchy-czyli-nowoczesna-informatyka-w-logistyce6334_2_0_3.html (dostęp: 22.10.2011).

LOGISTICS INFORMATION SYSTEM' DESIGNING

Summary

Today logistics is based on the information technology, and without them would be inefficient. More and more access to information technology means that logistics is currently undergoing profound changes. Logistics has entered into all functional areas of business. One reason for this is the widespread use of integrated information systems, allowing for logistics management in both large and small businesses.

On the IT market there is definitely a good system solution that would provide a comprehensive IT service logistics company to the extent it satisfactory. Always necessary to use a variety of solutions complementary to the principle of synergy meet all information needs of businesses, enabling efficient implementation of logistics IT systems.

Stanisław Stanek*

ANALIZA WYBRANYCH KONCEPCJI W OBSZARZE PROJEKTOWANIA WYMAGAŃ

Wprowadzenie

W kontekście podejmowanej działalności projektowej pojawiają się zazwyczaj cztery podstawowe pytania: o cele, dla których system ma być realizowany (dlaczego?), o metareguly obowiązujące w procesie realizacji systemu (jak?), o interesariuszy, których system ma wspomagać (kto?), o usługi jakie system ma realizować (co?). Specyfikę projektów informatycznych odzwierciedla ciąg ukierunkowanych na informatykę zaleceń organizacji normalizacyjnych. W szczególności standard ISO/IEC 12207-2008, opracowany dla potrzeb łatwiejszego łącznego posługiwania się standardami 21207 oraz 15504, definiuje używane w artykule pojęcia klient¹, użytkownik², developer³ oraz interesariusz⁴. Interesariusze będą wspomagać działania projektowe w nawiązaniu do postrzeganych potrzeb jakie projektowany system będzie zaspakajał. Ian Sommerville w jednej z częściej cytowanych pozycji [Somm03] podejmujących problematykę analizy wymagań pisze: „Problemy, które mają rozwiązywać inżynierowie oprogramowania, są często bardzo złożone. Zrozumienie ich natury może być trudne, zwłaszcza w wypadku nowych systemów. Trudno jest zatem dokładnie ustalić, co system powinien robić. Opisy usług i ograniczeń są wymaganiami stawianymi systemowi. Proces znajdowania, analizowania, dokumentowania oraz sprawdzania tych usług i ograniczeń nosi nazwę „inżynierii wymagań”.

* Prof. nadzw. dr hab. inż. Stanisław Stanek, Wyższa Szkoła Oficerska Wojsk Lądowych im. generała Tadeusza Kościuszki, 51-150 Wrocław, ul. Czajkowskiego 109, e-mail: stan_stanek@neostrada.pl

¹ Customer – organizacja lub osoba, która otrzymuje produkt lub usługę.

² User – jednostka lub grupa, korzystająca z systemu podczas jego użytkowania.

³ Developer – organizacja, która wykonuje zadania rozwoju (w tym analizy wymagań, projektowania, testowania) podczas procesu cyklu życia.

⁴ Stakeholder – osoba lub organizacja mająca prawo, akcje, roszczenia albo zainteresowana systemem lub jego charakterystykami nawiązującymi do potrzeb oraz oczekiwań. Można zauważyć (za [Some03, s. 122]), że w określaniu i analizowaniu wymagań mogą wziąć udział osoby z różnych stanowisk, np.: użytkownicy, którzy będą pracować z systemem, inżynierowie budujący lub pielęgnowujący inne powiązane systemy, menadżerowie przedsiębiorstwa, eksperci z danych dziedzin, a być może nawet reprezentanci związków zawodowych.

Problematyka inżynierii wymagań wydzieliła się z obszaru badań nad inżynierią oprogramowania. Za prekursorów tej problematyki badawczej uważa się Bella oraz Thayera, którzy w 1976 roku w następstwie analizy badań empirycznych konkludowali, że:

- niekompletne, niewłaściwe, niezgodne lub niejasne wymagania są liczne i mają krytyczny wpływ na jakość oprogramowania,
- wymagania dla systemu nie powstają w sposób naturalny, przeciwnie, muszą być zaprojektowane a następnie konsekwentnie przeglądane i dostosowywane.

Rozwój podstaw metodycznych inżynierii wymagań oraz szerzej inżynierii oprogramowania, jaki nastąpił w latach 80., nie spowodował znaczącego postępu w obszarze prac projektowych, co ilustrują np. następujące często cytowane wyniki raportu Standish Group's:

- 31% projektów zostało przerwanych przed zakończeniem,
- 53% projektów kosztowało ponad 189% wartości estymowanych,
- tylko 16% projektów zakończyło się zgodnie z planem (termin, koszty),
- projekty ukończone dostarczały jedynie 42% oryginalnych cech oraz funkcji,
Ponadto trzy najczęstsze problemy projektowe to:
 - brak informacji wejściowych pochodzących od użytkownika – 13%,
 - niekompletność wymagań oraz specyfikacji – 12%,
 - zmiany wymagań oraz specyfikacji – 12% [StGr94].

Wyniki te oznaczają, że na progu nowego tysiąclecia projektowanie wymagań w dalszym ciągu sprawiało znaczące trudności projektantom.

Dalsza interpretacja badań empirycznych z tego okresu utrwalała przekonanie o potrzebie bardziej adekwatnej pracy z użytkownikiem. Podejście zorientowane na użytkownika (UOD – User Oriented Design) uzyskało międzynarodową nobilitację najpierw w 1999 roku w postaci standardu ISO 13407, a ostatnio po aktualizacji, w nawiązaniu do numeracji zgodnej z innymi standardami użyteczności, w postaci obowiązującego obecnie standardu ISO 9241-210⁵. Standard jest postrzegany jako manifest skierowany do rozproszonych środowisk stosujących UOD nawołujący do zjednoczenia. Organizacje, które decydują się na jego wykorzystanie uzyskują terminologiczne i metodyczne wsparcie wypracowane przez międzynarodowe gremia ekspertów.

Krytyczną dyskusję dotychczasowych praktyk w obszarze specyfikowania wymagań wskazującą na przesłanki wprowadzenia podejścia zwinnego przedstawioną przez Dona Reinertsenę we wprowadzeniu do ostatniej książki Deana Lefingwella można streścić następująco:

⁵ W Polsce 21 lutego 2011 roku ukazał się w postaci normy PN-EN ISO 9241-210:2011 Ergonomia interakcji człowieka i systemu -- Część 210: Projektowanie ukierunkowane na człowieka w przypadku systemów interaktywnych.

- badania niezmiennie ukazują, że od 80% do 85% niepowodzeń w projektowaniu wiąże się z niepoprawnymi wymaganiami. Wiemy o tym od ponad dekady, jednak nie udało nam się dotychczas tego wyniku poprawić;
- dlaczego? Początkowo byliśmy organizowani funkcjonalnie, po prostu problem pozostawał poza granicami inżynierii – mogliśmy obwiniać marketing i zarządzanie produktem. Później, gdy przyjęto cross-funkcjonalne zespoły nakazano, aby słuchać głosu klienta, co miało rozwiązać problem;
- nie rozwiązało. Zignorowano fakt, że niejednokrotnie klienci nie wiedzą, czego chcą. Jeśli nawet wiedzą, to nie potrafią tego opisać. A jeśli potrafią, to raczej opisują swoje rozwiązanie problemu, a nie samą rzeczywistą potrzebę. Aby trzymać się prawdy, nie ma jednego „głosu klienta”. Mamy do czynienia z kakofonią głosów wskazujących na różne rozwiązania. Jeśli skupimy się jedynie na użytkowniku, to możemy przeoczyć „wymagania niefunkcjonalne”. A przecież, szczególnie w kontekście dynamiki zmian w otoczeniu, nie można interesariuszom odmówić prawa do rozwoju, do zmian, bo wówczas będziemy ich ograniczać, a nie wspomagać;
- tak więc musimy zaprzestać „chowania głowy w piasek”, bardziej niż mozolnie i kosztownie tworzyć **idealne** wymagania, musimy inwestować w organizowanie procesów i infrastruktury, wspierających wykrywanie i korygowanie braku dopasowania pomiędzy naszymi rozwiązaniami a zmieniającymi się potrzebami klienta [Lef11].

Celem artykułu jest analiza trzech wymienionych nurtów badawczych (inżynieria wymagań, podejście zorientowane na użytkownika, podejścia zwinne) w kontekście rozwiązań proponowanych dla potrzeb projektowania wymagań. Artykuł ma zidentyfikować, opisać oraz poddać pod dyskusję wybrane, proponowane w ramach wymienionych koncepcji, mechanizmy.

1. Prace nad inżynierią wymagań

Podejście inżynierskie koncentruje się na wykorzystaniu sprawdzonych podstaw teoretycznych w procesie rozwiązywania problemów lub tworzenia produktów. Nawiązując do przytoczonej we wstępie opinii Iana Sommerville, inżynieria wymagań jest zorientowana na wyodrębnienie oraz analizę procesu wymagań w organizacji. Aktywności w procesie inżynierii wymagań analizowali m.in. Houdek oraz Pohl [HoPo00], Lamswerde [Lams09]⁶:

- **tworzenie projektu**: aktywność związana z uruchomieniem projektu w celu opracowania nowego produktu lub modyfikowania istniejącego produktu;
- **odkrywanie wymagań** (określane też jako zbieranie, pozyskiwanie, wydobywanie wymagań): aktywność, w ramach której developer zwykle w interakcji z interesariuszami działają w celu zdefiniowania dziedziny projektu usług, jakie system powinien zapewniać oraz przyszłych ograniczeń;

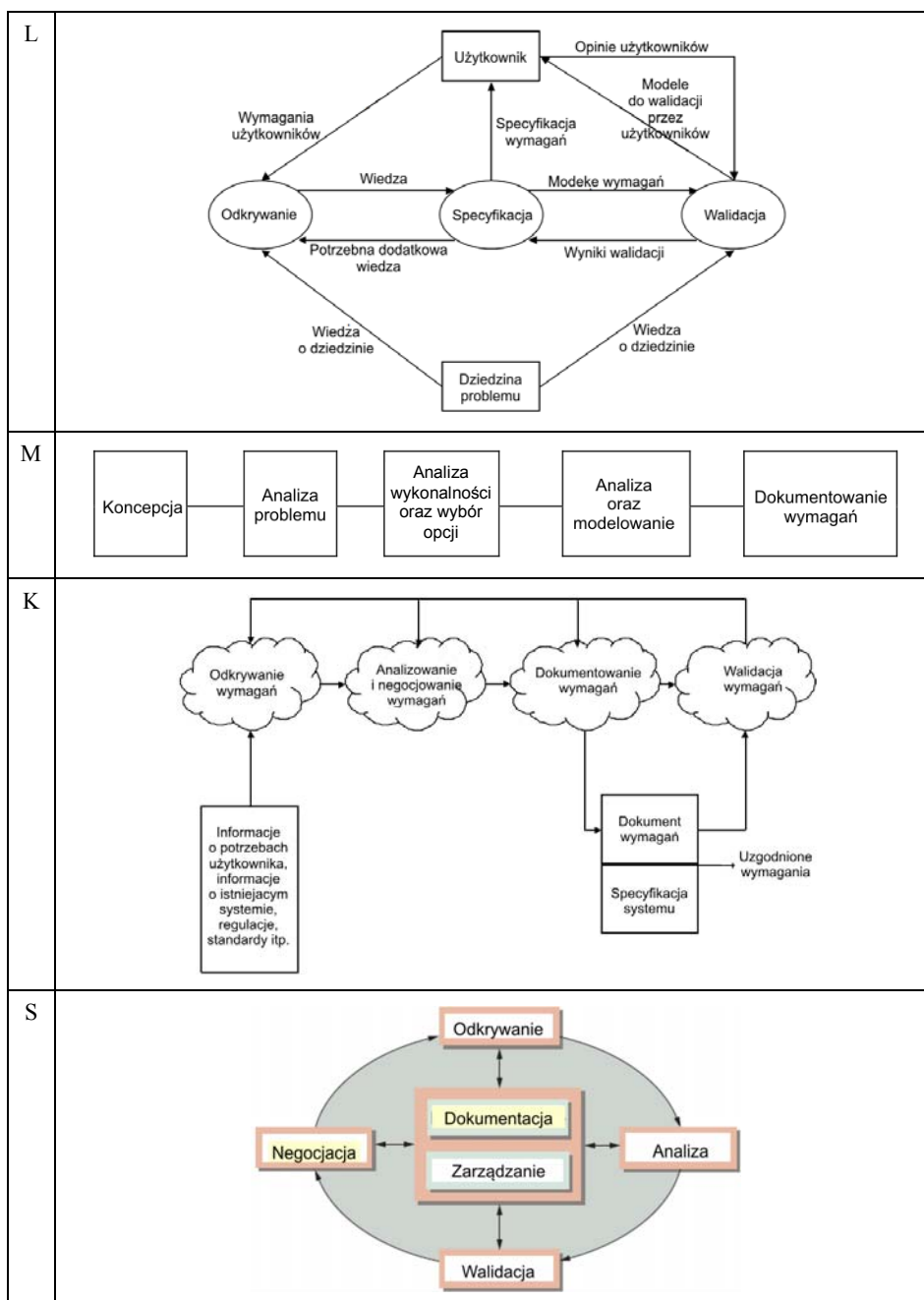
⁶ Szerszy opis poszczególnych aktywności można także znaleźć np. w [Lams09].

- **analiza (doskonalenie) wymagań:** po odkryciu wymagania są interpretowane oraz strukturalizowane. Następnie są dokumentowane. Tutaj aktywność ta jest wyodrębniona, jednak często przeplata się z odkrywaniem wymagań, ponieważ analiza niezmiennie ma swoje miejsce w fazie odkrywania;
- **negocjacja wymagań:** aktywność związana z negocjacjami z interesariuszami w celu uzyskania porozumienia, co do definicji wymagań;
- **sprawdzanie wymagań:** weryfikacja formalna ma na celu zapewnienie formalnej poprawności wymagań. Nawiązuje do poprawności zapisu oraz wewnętrznej zgodności. Walidacja wymagań ma z kolei na celu stwierdzenie czy zebrane wymagania i stworzony dokument wymagań definiują system zgodny z oczekiwaniami interesariuszy. Dobrą praktyką jest organizowanie inspekcji wymagań, w których biorą udział interesariusze lub ich przedstawiciele;
- **zarządzanie wymaganiami:** dotyczy planowania oraz zarządzania zmianami wymagań. Zarządzanie zmianą sprawia, że ustandaryzowane informacje są gromadzone dla każdej zmiany oraz że całkowite koszty oraz korzyści proponowanych zmian są analizowane, dlatego też zarządzanie zmianą obejmuje ocenę ryzyka i analizę oddziaływań;
- **śledzenie wymagań:** pozwala na analizowanie pochodzenia wymagań, związków zachodzących pomiędzy wymaganiami oraz między wymaganiami a ich realizacją w projektowanym systemie. Ułatwia utrzymanie spójności w obliczu zmian.

Do wielokrotnie cytowanych później modeli procesu inżynierii wymagań można zaliczyć (rys. 1):

- L – model iteracyjny Loucopoulosa, Karakostasa [LoKa95],
- M – model liniowy Macaulaya [Maca96],
- K – model liniowy z iteracjami między działaniami Kotonaya oraz Sommerville’a [KoSo98],
- S – podsumowujący doświadczenia model cykli aktywności Sommerville’a [Some05].

Przeprowadzane w dalszej kolejności, w dwóch firmach A i B, badania [SAJP02] zmierzające do wskazania, który z trzech modeli M, K, L adekwatnie odzwierciedla realizowane w praktyce prace nad wymaganiami, choć odsłoniły wiele prawidłowości, to jednak nie przyniosły rozstrzygnięcia. W dwóch projektach (A2, B2) inżynieria wymagań była realizowana jako wydzielona faza, a proces wymagań nawiązywał do modelu liniowego M. W trzech pozostałych projektach inżynieria wymagań była realizowana ciągle w czasie całego procesu projektowania.



Rys. 1. Modele inżynierii wymagań

Odnotowano wiele iteracji, co było szczególnie wyraźne, gdy w ramach przyrostów były realizowane prototypy (A3). Podsumowując, należy zauważyć, że proces inżynierii wymagań kształtuje się w nawiązaniu do szerszego kontekstu. Model przyrostowy inżynierii wymagań L w miarę dobrze oddaje iteracyjny charakter procesu projektowania, jednak nie pokazuje progresji w procesie projektowania. Mając na uwadze powyższe doświadczenia, za podstawę do dalszych rozważań przyjęto późniejszy model S (por. rys. 1).

Tabela 1

Badania nad modelami L, K, M

	A – duża firma produkcja			B – duża firma finanse	
	Proj. A1	Proj. A2	Proj. A3	Proj. B1	Proj. B2
Cel projektu	Wdrożenie ERP	Upgrade systemów	Wsparcie promocji	Rozwój witryny	Wymiana CRM
Rozmiar	Duży	Średni	Mały	Średni	Mały
Osób	26-100	11-25	1-10	1-10	1-10
Osobomiesiący	540	24	3	100	8
Priorytet	Czas/Koszt	Czas/Funk.	Funk.	Czas	Czas/Funk.
Tworz. projektu	Nie	Explicite	Implicite	Implicite	Explicite
Odkrywanie wym.	Explicite	Explicite	Explicite	Explicite	Explicite
Analiza wym.	Explicite	Implicite	Explicite	Explicite	Implicite
Negocjacje wym.	Implicite	Implicite	Implicite	Explicite	Explicite
Zarządzanie wym.	Explicite	Implicite	No	No	Implicite
Śledzenie wym.	Implicite	Implicite	No	No	Implicite
Model	(K)	(M)	(L)	(K) (L)	(K) (M)

Źródło: Na podstawie: [SAJP02].

Obserwowane w praktyce duże zróżnicowanie procesów inżynierii wymagań po części tłumaczy koncepcja dojrzałości procesowej. Dojrzałość procesowa wyraża się zakresem, w jakim procesy są formalnie: zdefiniowane, zarządzane, elastyczne, mierzone i efektywne [Gaje03]. W latach 90. wielu developerów oprogramowania, w celu doskonalenia procesów wewnętrznych, podjęło prace (por. np. [Paulk95]) nad wdrożeniem opracowanego przez Software Engineering Institute SEI⁷, modelu Capability Maturity Model for Software (SW-CMM). W 2000 roku wydał CMMI-SE/SW zintegrowany model obejmujący zarówno problematykę oprogramowania, jak również problematykę inżynierii

⁷ Przy Cornege Melion University, por. <http://www.sei.cmu.edu/>.

niarii systemów. Zarówno SW-CMM, jak również CMMI-SE/SW dostarczają specyficzne rekomendacje dla rozwijania oraz zarządzania wymaganiami. Krótki opis tych modeli z punktu widzenia problematyki wymagań można znaleźć w załączniku do książki Karla Wiegensa [Wieg03]. Modele te nie pokrywają całości procesu wymagań (por. np. [Rog98; BeHR03]), co zaowocowało dużą ilością modeli dojrzałości ukierunkowanych na doskonalenie procesu wymagań u developerów. Oto najlepiej zweryfikowane modele, zalecane do przeanalizowania: REGPG [SoSa97], REPM [GoTe02], RMM [Heum03], R-CMM [BeHR03], [Tra108], Uni-REPM [Nguy10], IAG [WWW1]. Ze szczególnym zainteresowaniem spotkał się pierwszy z wymienionych modeli (por. np. [KaAK02; SoRa05; XuSS06; SDYS09]). Jak zauważają autorzy, „(...) odmiennie niż CMM lub ISO 9001-3, REGPG w swoim zamiarze nie ma służyć jako standard lub do celów akredytacji, lecz jako praktyczny przewodnik łatwy do zrozumienia, a także łatwy w stosowaniu”. W procesie oceny developera rozważono 66 dobrych praktyk sklasyfikowanych według ośmiu obszarów oraz trzech poziomów (Zał. 1). Przydzielono oceny powszechności stosowania każdej z praktyk.

- 0 – praktyka nigdy niestosowana (lub prawie nigdy),
- 1 – praktyka stosowana sporadycznie (jeśli ktoś zadecyduje o jej zastosowaniu),
- 2 – praktyka często stosowana (nie jest jednak standardem),
- 3 – praktyka zawsze stosowana (jest standardem).

Obliczono wskaźniki sumaryczne S1 – suma ocen za kryteria z poziomu podstawowego oraz S23 – suma ocen za kryteria z pozostałych poziomów. Poziom dojrzałości odczytano z poniższej tabeli.

Tabela 2

Poziomy dojrzałości modelu REGPG

Lp.	Poziom dojrzałości	Warunek	Komentarz
1	Początkowy	$S1 < 56$	Proces wymagań planowany ad hoc. Wynik zależy od realizatorów
2	Powtarzalny	$S1 > 55, S23 < 41$	Developer otwarty na narzędzia wspomagające prace oraz na ulepszanie praktyk
3	Zdefiniowany	$S1 > 85, S23 > 40$	Developer stosuje dobre praktyki inżynierii wymagań na co dzień

Źródło: Na podstawie: [SoSa97].

Słabo ocenione praktyki stają się wskazówką dla potrzeb dalszego doskonalenia. Poziomy 1 oraz 2 korespondują z odpowiednimi poziomami modeli SEI, w przypadku poziomu trzeciego, w nawiązaniu do znaczącej nieokreśloności w obszarze wymagań nie da się dokładnie sprecyzować poziomu korespondującego modelu SEI (3, 4 lub 5).

W analizie przypadków skonfrontowano jak przedstawione modele i zalecenia są realizowane w praktyce. Arao, Goto i Nagata [ArGN05] podejmują istotny problem przejścia od modelu biznesowego do modelu systemowego, zestawiając „jaki być powinien proces wymagań” z „jaki proces wymagań jest” por. tab 3.

Tabela 3

Lp.	Jak być powinno?	Jak jest?
1	Klienci i developer zgadzają się na nowy proces biznesowy dla realizacji celów projektu	Spojrzenie klienta oraz developera na nowy proces nie jest jednoznaczne
2	Developer adekwatnie wydobywa oraz definiuje wymagania klientów dotyczące systemu/oprogramowania	Developer określa wymagania klienta do pewnego stopnia, ale nie na poziomie szczegółowym
3	Developer definiuje specyfikację, która spełnia wymogi klienta	Developer odkrywa wymagania klienta w nawiązaniu do postępu prac nad specyfikacją wymagań
4	Developer zarządza zmianami wymagań szybko i dokładnie, zgodnie ze zmianami wymagań klienta	Developer nie jest zdolny zarządzać zmianami wymagań adekwatnie oraz terminowo, z powodu zbyt dużej liczby zmian

Popularne wyjaśnienie dla rozbieżności między tym, co być powinno, a tym, co jest, głosi, że użytkownik nie jest zdolny stanowić o swoich preferencjach przy przejściu do modelowania systemowego bez kontaktu z artefaktami świata rzeczywistego. Autorzy rozważanych badań nie do końca podzielają powyższe wyjaśnienie. Analizując przykładowy proces „śledzenie wyników biznesowych w tygodniu poprzedzającym złożenie zamówienia”, dostrzegają złożoność techniczną (m.in. potrzeba współdziałania punktu sprzedaży utrzymującego sprzedaż z ubiegłego tygodnia z systemem finansowym naliczającym wskaźniki oraz mechanizmami adekwatnej prezentacji). Każdemu procesowi odpowiada wiele funkcji systemowych, następuje zmiana terminologii oraz dodanie szczegółów technicznych, które mogą w istotny sposób oddziaływać na proces biznesowy. Narzuca się tutaj powiedzenie, że niestety „diabeł tkwi w szczegółach” takiej operacji. W rzeczywistości ponieważ niemożliwe jest wyartykułowanie wymagań w tym samym czasie zarówno dla klientów, jak i developerów, to proces musi być do pewnego stopnia ciągły i interaktywny, aby wymagania okazały się zrozumiałe i dokładne. Autorzy budują narzędzia autorskie wspomagające

specyfikowanie oraz śledzenie (tracing) wymagań pozyskiwanych w procesie przejścia od modelu biznesowego do SRS na podstawie Microsoft Excel oraz Access. Dalsze prace projektowo wdrożeniowe prowadzą do następujących konkluzji:

- nie można rozpoczynać procesu przechodzenia do SRS zanim spojrzenie biznesowe nie będzie ustabilizowane,
- należy starać się negocjować model wymagań ze wszystkimi interesariuszami,
- proponowane rozwiązanie sprawdzało się w różnorodnych realizowanych projektach,
- brak szkoleń członków projektu oraz liderów powoduje niepowodzenie procesu implementacji.

2. Podejście zorientowane na użytkownika

Podejście zorientowane na użytkownika (UOD – User Oriented Design) uzyskało międzynarodową nobilitację najpierw w 1999 roku w postaci standardu ISO 13407, a następnie po aktualizacji, w nawiązaniu do numeracji zgodnej z innymi standardami użyteczności, w postaci obowiązującego obecnie standardu ISO 9241-210⁸. Standard jest postrzegany jako manifest skierowany do rozproszonych środowisk stosujących UOD nawołujący do zjednoczenia. Organizacje, które decydują się na jego wykorzystanie uzyskują terminologiczne i metodyczne wsparcie wypracowane przez międzynarodowe gremia ekspertów.

Rozważania rozpoczęto od przypomnienia pryncypiów podejścia zorientowanego na użytkownika (Por. Zał. 3).

W standardzie ISO 9241-210 podstawową, wyróżnioną aktywnością jest identyfikowanie i specyfikowanie wymagań. Zaleca się utworzenie jednoznacznie określonych wymagań użytkownika w stosunku do zamierzonego kontekstu wykorzystania oraz celów biznesowych systemu.

Tradycyjne podejścia inżynierii wymagań koncentrowały się na identyfikacji funkcjonalnych wymagań oraz na zapewnieniu, aby rozwijany produkt je spełniał. Inne niefunkcjonalne wymagania (wydajności, niezawodności, użyteczności, łatwości konserwacji, rentowności) miały mniejsze znaczenie. Dopiero z perspektywy użytkownika, niefunkcjonalne wymagania stają się krytyczne dla powodzenia we wdrożeniu nowego systemu. Z perspektywy użytkownika zasadne staje się pytanie czy dotychczasowe procesy, również te, które wiążą się z działaniem przyszłego systemu, ale nie będą informatyzowane,

⁸ W Polsce 21 lutego 2011 ukazał się w postaci normy PN-EN ISO 9241-210:2011. Ergonomia interakcji człowieka i systemu -- Część 210: Projektowanie ukierunkowane na człowieka w przypadku systemów interaktywnych.

nie powinny być zreorganizowane. W szerszym oraz lepiej umocowanym gronie interesariuszy rozwija się problematyka modelowania biznesowego, która nabiera większego znaczenia oraz uzyskuje potrzebne możliwości realnego oddziaływania. Zaczyna uwierać niezrozumiała dla użytkownika tradycyjna specyfikacja. Rozwija się alternatywa tradycyjnej specyfikacji, w tym prototypy i bardziej iteracyjne oraz przyrostowe podejścia, znajdujące swoje miejsce w rozszerzonej specyfikacji wymagań, która przekonująco opisuje w jaki sposób przyszły system będzie funkcjonował. Przyszły użytkownik wiele uwagi poświęca ważnej i zrozumiałej dla niego specyfikacji wejść i wyjść. Napotykając na różnorodne problemy w komunikacji bezpośredniej, można dostrzec potrzebę modelowania umysłowości odbiorcy naszej pracy, utworzenia tzw. modelu mentalnego. Model mentalny daje głębsze zrozumienie ludzi, ich motywacji, procesów myślowych, a także otoczenia emocjonalnego i filozoficznego działań. Może być reprezentowany poprzez diagram afiniczny (podobieństwa) zachowań użytkowników [You08].

W wielu pracach są analizowane zalety i wady modelowania biznesowego oraz systemowego z wykorzystaniem UML oraz innych rozwiązań. Wiele interesujących, z perspektywy problematyki specyfikowania wymagań, analiz przedstawił Martin Glinz ze współpracownikami. W artykule z 2000 roku [Glin00] w nawiązaniu do analizy przykładowego systemu (zdalnej opieki medycznej) przedstawił, szeroko wzmiankowane w innych pracach, dziewięć słabości analizy wymagań opartej na UML⁹. W kolejnym artykule [FrGK06] w nawiązaniu do wcześniejszych badań podjęto obserwację oraz analizę pracy wyróżnionego zespołu projektowego. W trakcie pierwszego miesiąca pracy zespół poszukiwał, techniką prób i błędów, adekwatnej do kontekstu, metody pracy z wymaganiami. Poczynając od najprostszych metod, rozwijał swój arsenał przydatnych w danej sytuacji rozwiązań, dochodząc do metody Ericssona-Penkera, która została przyjęta jako metoda wiodąca. W trakcie kolejnych dwóch miesięcy udało się zespołowi z wykorzystaniem wypracowanych, w szerokim zakresie autorskich, podstaw metodycznych opracować adekwatną specyfikację wymagań. W trakcie poszukiwań zespół zaadaptował Enterprise Architecta, które to narzędzie zostało z powodzeniem wykorzystane w trakcie dalszych prac.

⁹ 1. Brak elementów aktywnych w diagramach przypadków użycia. 2. Brak możliwości reprezentowania kontekstu – zależności między aktorami. 3. Brak możliwości adekwatnego modelowania struktury przypadków użycia i hierarchii. 4. Brak możliwości adekwatnego specyfikowania interakcji między przypadkami użycia. 5. Trudność modelowania zależnych od stanu zachowań systemu. 6. Nieporęczne modele przepływu informacji. 7. Brak możliwości modelowania zachowań charakterystycznych dla komponentów wysokiego poziomu, takich jak podsystem. 8. Trudności przy modelowaniu dekompozycji systemu rozproszonego. 9. Brak aspektowo zorientowanego spojrzenia na złożony system.

3. Podejście zwinne

Podejście zwinne wyłoniło się w kontekście dyskusji nad cyklem życia oprogramowania na początku naszego wieku. Dominujące w poprzednim wieku modele: kaskadowy, spiralny oraz prototypowania, wypracowane przez takie autorytety, jak Winston Royce, Barry Bohm, Daniel McCracen, Michael Jacobson oraz Garry Russel Gladden dały podstawę dla metodyk określanych jako twarde, z racji ich ukierunkowania na rozległą dokumentację, rozbudowane procesy, długie oraz sekwencyjne fazy. Jak wynika jednakże również z przedstawionych wcześniej rozważań, w praktyce szczególnie w niewielkich firmach rozwiązania te w całości nie przyjęły się oraz trwały poszukiwania alternatywnych (lekkich) rozwiązań (por. tab. 4).

Tabela 4

Przykładowe lekkie rozwiązania metodyczne
jake wyłoniły się w latach 90.

Rok	Metodyka	Twórca	Liczba praktyk	Liczba ról	Liczba artefaktów
1991	Crystal Methods	Alistair Cockburn	14	10	25
1993	Scrum	Jeff Sutherland	5	3	5
1993	Dynamic Systems Dev.	Grupa firm brytyjsk	15	12	23
1998	Extreme Programming	Kent Beck	28	7	7

Twórcy podejść alternatywnych dostrzegając potrzebę wymiany doświadczeń, w 2000 roku skorzystali z zaproszenia Kenta Becka i spotkali się po raz pierwszy na konferencji w Oregon. Bob Martin we wrześniu 2000 roku zaprosił liderów wyłaniającego się lekkiego podejścia na następne spotkanie w 2001 roku. Alistair Cockburn rekomendował zastąpienie, używanego dla potrzeb identyfikacji proponowanych rozwiązań metodycznych, określenia „lekkie” określeniem „zwinne”. Następne spotkanie odbyło się w dniach 11-13 lutego 2001 roku w Utah, gdzie zgromadziło się 17 entuzjastów nowego kierunku. Rezultatem spotkania było powołanie zrzeszenia nazywanego Agile Alliance mającego na celu promowanie podejść zwinnych i wspieranie osób zainteresowanych [WWW4]. Ogłoszono Manifest Zwinnego Wytwarzania Oprogramowania sformułowany w postaci czterech ogólnych reguł oraz dwunastu szczegółowych zasad (por. np. [MaMa08, s. 39-46]). Już pierwsze spojrzenie na pryncypia podejścia zwinnego wskazuje na zasadniczą zmianę optyki patrzenia na problematykę analizy wymagań. Rozwój polega na rozszerzeniu dotychczasowego logicznego, systemowego spojrzenia o: kontekst retrospektywnej, krytycznej

analizy dotychczasowych praktyk, przejęte z nauk zarządzania zagadnienia organizacji pracy wysoko wydajnych zespołów oraz tworzenie samoorganizujących się, szczupłych rozwiązań, a także na fascynacji kulturą kaizen.

Bardziej szczegółową analizę rozpoczęto od zagadnienia wyłaniania wizji systemu. Podstawowym założeniem podejścia zwinnego jest ograniczenie wstępnych prac koncepcyjnych (tzw. paraliżu przez analizę), krótki cykl dostarczenia produktu, a następnie szybkie adaptacje produktu zgodnie z aktualną odpowiedzią płynącą z rynku. Niemniej jednak osoba reprezentująca interesariuszy, w SCRUM – właściciel produktu, w trakcie sesji poświęconej planowaniu wydania poddaje pod dyskusję wizję produktu. Aby dobrze spełnić swoje zadanie, właściciel projektu powinien zastanowić się nad obszarami projektu (tzw. drajwery projektu), w których oczekuje na generowanie wartości, patrząc z punktu widzenia biznesu. Uczestnicy mogą zastanawiać się nad zwrotem z inwestycji. W nawiązaniu do problemu projektowania produktu dobrze jest wcześniej opracować tzw. minimalny zestaw jego cech marketingowych (MMFS). Zwinna wizja komunikuje intencje strategiczne oraz odpowiada na następujące pytania:

- dlaczego budujemy ten produkt, system lub aplikację?
- jaki problem rozwiązuje?
- jakie cechy oraz korzyści dostarcza?
- komu są dostarczane te cechy oraz korzyści?
- jaka wydajność, niezawodność i skalowalność wiąże się z produktem?
- jak wygląda porównanie produktu z istniejącymi już na rynku oraz z przygotowywanymi do wydania?
- jak będziemy uzyskiwać zysk na sprzedaży? Jakie będą źródła przychodów oraz jaki będzie model biznesowy?
- czy mamy zdolność wykonania produktu? Czy będziemy w stanie produkt rozwijać? Jaka będzie roadmapa rozwoju produktu?

Analizując dobre praktyki, szczególnie w obszarze RUP, gdzie dokument wizji jest jednym z kluczowych, szczegółowo opracowywanym artefaktem, Dean Leffingwell zaleca [Leff11], aby również w zwinnych projektach, zwłaszcza w dużych, taki dokument opracowywać. Postuluje jednak, aby w nawiązaniu do zasad podejścia zwinnego opracowywać tylko jeden dokument wizji liczący od 5 do 10 (maks. 20) stron, tak aby aktualizacja tego dokumentu nie była uciążliwa. W załączniku do ostatniej książki Deana można znaleźć przykładowy wzorzec dokumentu wizji dla potrzeb projektów zwinnych (4 strony).

W odpowiedzi na pytanie jak wypełnić wzorzec wizji dla potrzeb projektów zwinnych Roman Pichler [Pich10] zaleca takie techniki, jak: prototypy (papierowe, sketches, spikes, mock-up), personas and scenerios (identyfikowanie się z klientem oraz zbadanie jak produkt oddziałuje na jego życie), vision box lub trade journal review (zastanowienie się nad trzema wypunktowaniami „sprzedającymi” produkt do centralnego umieszczenia na opakowaniu lub przez

retrospekcje przeanalizowanie, co chciałoby się przeczytać o produkcie po jego wydaniu), Kano model (projektowanie funkcjonalności produktu poprzez wyróżnienie funkcji podstawowych, wydajnościowych oraz wywołujących zachwyty).

Znaczącym wkładem podejścia zwinnego do problematyki specyfikowania wymagań jest wyartykułowana (Kent Beck) pierwotnie w ramach podejścia ekstremalnego (XP) koncepcja krótkich wypowiedzi użytkowników (historii użytkownika) – spisu rzeczy jakie system powinien zdaniem użytkownika robić. Historia użytkownika rozpoczyna tzw. grę planistyczną, w ramach której najpierw w wyniku intensywnych kontaktów między developerami oraz użytkownikami zbyt duże historie są dzielone na mniejsze oraz zbyt małe są scalane w większe. Developerzy przypisują każdej historii użytkownika koszt jej wykonania, a użytkownicy wybierają historie, które będą realizowane w następnej iteracji, nawiązując do szacowanej wydajności zespołu. W wyniku prac Mike'a Cohna historie użytkownika zostały zaadaptowane również w SCRUM jako narzędzie do budowy rejestru produktowego oraz przy definiowaniu zawartości poszczególnych iteracji – tzw. Sprintów. W nawiązaniu do problematyki specyfikowania wymagań historia użytkownika może być postrzegana jako wysokopoziomowy opis wymagań do zaimplementowania. W zrozumieniu tego związku pomaga rozwijana w ramach podejścia zwinnego koncepcja poruszającego dokumentu¹⁰ – spełniającego rolę pamięci zewnętrznej ilustrującej w obrazowej, poruszającej wyobraźnię formie bieżącą komunikację w zespole projektowym. Poruszające dokumenty silnie nawiązują do kontekstu, są dokumentacją do bieżącego wykorzystania w odróżnieniu od dokumentów reprezentacyjnych (por. np. [Elss10]). W podejściu zwinnym historia użytkownika jest przykładem poruszającego dokumentu o bardziej sformalizowanej postaci (np. według Mike'a Cohna „Jako <rodzaj użytkownika>, chcę <cel>, tak aby <uzasadnienie>” [Co04, s. 127]), który przykładowo mógłby przyjąć następującą postać: „Jako użytkownik standard chcę, aby przy dużym jednorazowym zakupie był udzielany rabat, tak aby duży jednorazowy zakup był bardziej konkurencyjny”.

W podejściach zwinnych centralne znaczenie, z perspektywy wymagań, ma proces opracowywania testów w nawiązaniu do praktyk: zautomatyzowane testy programisty, wytwarzanie kończone testami lub wytwarzanie poprzedzone testami (WPT). Jakkolwiek dalsze rozważania są w dużej części prawdziwe również dla dwóch pierwszych praktyk, to najbardziej wyraźne powiązania można uzyskać w przypadku wdrożenia praktyki WPT. Na wstępie należy zauważyć, że sztuka pisania testów rozwiązuje w ramach podejść zwinnych problem wyższego piętra specyfikacji wymagań. Odnosi się to zarówno do testów jednostkowych (białej skrzynki), gdy programista powodowany wymaganiami wyraża swoje intencje w formie testu, jak również akceptacyjnych (czarnej skrzynki),

¹⁰ Ang. *evocative document*.

w których interesariusz sprawdza czy jego wymagania zostały właściwie zrozumiane, a programista implementuje wymagania w postaci kodu. W jednej z konkluzji książki Roberta Martina i Mican'a Martina można przeczytać, że „(...) zarówno testy jednostkowe, jak i akceptacyjne pełnią funkcję swoistej dokumentacji. Ponieważ dokumentacja w tej formie może być kompilowana i wykonywana, jej wiarygodność i precyzja nie powinny budzić najmniejszych wątpliwości. Co więcej, tego rodzaju testy są pisane w jednoznacznych językach zrozumiałych dla właściwych grup odbiorców” [MaMa08, s. 82]. Nie można zakończyć analizy problematyki dokumentowania wymagań w podejściach zwinnych bez spojrzenia z perspektywy refaktoryzacji kodu, stanowiącego w swojej istocie również kolejne piętro specyfikacji wymagań. Dla programisty kod nie-refaktoryzowany jest kodem ciężkim, zwykle niezrozumiałym, a jego leczenie w formie ekstremalnej nawiązuje do leczenia objawowego według mechanizmu czarnej skrzynki. Przeciwnie lekki, refaktoryzowany kod stanowi o zdolności do przetrwania oprogramowania w burzliwym, zmieniającym się otoczeniu.

Z uwagi na wymogi dotyczące objętości artykułu, przeanalizowano jedynie wyzwanie jakie stoi przed twórcami gier komputerowych. Dynamika rozwoju kosztów jest w tym obszarze znacznie większa niż dynamika rozwoju rynku. Branża staje na rozdrożu: Czy czeka ją kryzys podobny do tego z 1983 roku, czy też uda się wypromować nowe rynki, a może wystarczy odchudzająca terapia z wykorzystaniem podejścia zwinnego? W tym sektorze rynku liczy się radość z grania, o której można orzekać trzymając w ręku manipulator. Każdy dzień zyskany dla bardziej obiecującego projektu poprzez precyzyjniejszą identyfikację oraz wycofywanie projektów brnących ku klęsce może stanowić o „być albo nie być” firmy. Znamienne są doświadczenia firmy CCP Games zatrudniającej ponad 400 pracowników w trzech lokalizacjach: Reykjavik, Atlanta oraz Shanghai. Jesienią 2008 roku, CCP podjęła ambitne wyzwanie zaprojektowania dodatku do EVE Online o nazwie Apokryfy w ciągu sześciu miesięcy. W przypadku Apocrypha udana realizacja oznaczała współdziałanie ponad 120 developerów w 13 zespołach Scrum zatrudniających 9 właścicieli produktu oraz pomieszczonych na trzech kontynentach (por. [Keit10]). Zwinne rozwiązania sprawdzają się więc również w dużych projektach.

Podsumowanie

Uczyniono znaczące wysiłki badawcze w zakresie poszukiwania modelu inżynierii wymagań. Weryfikacje empiryczne wskazują, że pierwsze wielokrotnie cytowane modele mają charakter raczej normatywny niż opisowy. Model Sommerville'a wydaje się być rozsądnym kompromisem. Ciekawą własnością tego modelu jest jego podobieństwo do zalecanego dla procesów innowacyjnych modelu Stewarta – Deminga PDCA.

Kolejny rozważony problem dotyczy rozwoju dojrzałości prowaiderów w obszarze wymagań. Modele CMMI-SW dostarczają stabilną, choć nie zawsze wystarczającą podstawę. Wskazane środowiska badawcze oraz prowaiderzy rozwijają te rozwiązania dodając własne narzędzia, które doskonalą zwarty schemat myślowy zaproponowany w jednym z pierwszych tego typu rozwiązań, zaproponowany przez Sommerville'a oraz Sawyera. Z uwagi na brak opracowania tego modelu w literaturze krajowej został on w artykule szerzej scharakteryzowany.

Umiejętne wykorzystanie narzędzi, również autorskich, wspomagających proces zarządzania wymaganiami może znacząco wpłynąć na skuteczność developera. Wykorzystanie tego typu narzędzi niestety nie jest powszechne. Aktualny krótki przegląd narzędzi można znaleźć np. w [ShSM11] .

Wydanie normy PN-EN ISO 9241-210:2011 inspirowane do przedstawienia oraz dalszego przedyskutowania pryncypiów podejścia zorientowanego na użytkownika. Podejście to rekomenduje potrzebę, metody oraz dobre praktyki głębszego uwzględnienia kontekstu użytkownika w procesie prac nad wymaganiami.

Introspektywne badania Glinza identyfikują występujący w praktyce proces poszukiwania adekwatnych do kontekstu konkretnych rozwiązań w nawiązaniu do ogólnych modeli, norm oraz praktyk . Ten kluczowy, słabo opisany, proces determinuje sukces lub porażkę projektu.

Kontekst wymagań w podejściu zwinnym jest istotny, coraz lepiej rozumiany oraz opisany w aktualnie ukazującej się literaturze. Wyłaniają się cztery poziomy szczegółościami rozwiązań:

- poruszająca, kolektywnie rozwijana wizja,
- napędzające kolejne iteracje, adekwatnie skalowane oraz wybierane do realizacji historie użytkownika,
- sprawujące pieczę nad zmianami testy,
- refaktoryzowany kod.

Wypracowane w ramach różnych podejść rozwiązania mogą być integrowane adekwatnie do potrzeb. Przykładem może być rozwijanie dojrzałości wymagań w zespołach wykorzystujących podejście zwinne.

Literatura

- [ArGN05] Arao T., Goto E., Nagata T.: „*Business Process*” *Oriented Requirements Engineering Process*. RE '04 Proceedings of the 13th IEEE International Conference on Requirements Engineering. IEEE Computer Society, Washington.
- [BeHr03] Beecham S., Hall T., Rainer A.: *Software Process Improvement Problems in Twelve Software Companies: An Empirical Analysis*. „Empirical Software Engineering” 2003, Vol. 8, No. 1.

- [Co04] Cohn M.: *User Stories Applied for Agile Software Development*. Addison-Wesley, 2004.
- [Elss10] Elssamadisy A.: *Agile wzorce wdrażania praktyk zwinnych*. Helion, Gliwice 2010.
- [FrGK06] Fricker S., Glinz M., Kolb P.: *A Case Study on Overcoming the the Requirements Tar Pit*. „Journal of Universal Knowledge Management” 2006, Vol. 1, No. 2.
- [Gaje03] Gajewski P.: *Koncepcja struktury organizacji procesowej*. TNOiK, Toruń 2003.
- [Glin00] Glinz M.: *Problems and Deficiencies of UML as a Requirements Specification Language*. Proceedings of the 10th International Workshop on Software Specification and Design (IWSSD-10), San Diego, November 2000.
- [GoTe02] Gorschek T., Tejle K.: *A Metod for Assessing Requirements Engineering Process Maturity in Software Projects*. Blenking Institute of Technology, Master Thesis Computer Science no. MSC-2002:2, 2002.
- [Heum03] Heumann J.: *The Five Levels of Requirements Management Maturity*, http://www.therationaledpe.com/conIenI/feb_03/f_managementMaturity_jh.jsp [dostęp: 28.11.2012].
- [HoPo00] Houdek F., Pohl K.: *Analyzing Requirements Engineering Process a Case Study*. Proceedings of the 11th International Workshop on Database and Expert Systems Applications, Greenwich, UK 2000.
- [KaAK02] Kauppinen M., Aaltio T., Kujala S.: *Lessons Learned from Applying the Requirements Engineering Good Practice Guide for Process Improvement*. In: Proceedings of the 7th European Conference on Software Quality, Helsinki.
- [Keit10] Keith C.: *Agile Game Development with Scrum*. Addison-Wesley, Boston 2010.
- [KoSo98] Kotonya G., Sommerville I.: *Requirements Engineering – Processes and Techniques*. John Wiley & Sons. UK 1998.
- [Lams09] Lamsweerde A.: *Requirements Engineering*. John Wiley & Sons, 2009.
- [Leff11] Leffingwell D.: *Agile Software Requirements*. Pearson Education, 2011.
- [Maca96] Macaulay L.A.: *Requirements Engineering*. Springer-Verlag, Berlin, Heidelberg, New York 1996.
- [MaMa08] Martin R., Martin M.: *Agile programowanie zwinne*. Helion, Gliwice 2008.
- [Nguy10] Nguyen T.: *The Creation of Uni-REPM*. Blenking Institute of Technology, Master Thesis Computer Science no. MSE-2010:27, 2010.
- [Paulk95] Paulk M., Weber C., Curtis B., Chrissis M.: *The Capability Maturity Model: Guidelines for Improving the Software, Process*. Addison-Wesley, Reading-MA, Boston 1995.
- [Pich10] Pichler R.: *Agile Product Management with SCRUM*. Addison-Wesley, 2010.
- [PISO11] PN-EN ISO 9241-210:2011 Ergonomia interakcji człowieka i systemu -- Część 210: Projektowanie ukierunkowane na człowieka w przypadku systemów interaktywnych.

- [Rog98] Rogoway P.: *How to Reap the Business Benefit from SPI: Adding SPICE while Preserving the CMM (Motorola): SPI NEWSPAPER*, http://www.iscn.at/select_newspaper/assessments/motorola.html [dostęp: 28.11.2012].
- [SAJP02] Sacha M., Aurum A., Jeffery J., Paech B.: Proceedings of the Seventh Australian Workshop on Requirements Engineering Process Models in Practice, <http://www.deakin.edu.au/events/awre02/Melbourne>, 2002, <http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS//Vol-69> [dostęp: 28.11.2012].
- [SAJP02] Sacha M., Aurum A. Jeffery R., Paech B.: *Requirements Engineering Process Models in Practice. The seventh Australian Workshop on Requirements Engineering: proceedings*, Melbourne, Victoria, School of Information Systems, Deakin University, 2002.
- [SDYS09] Shrivastava A., Darhan M., Yagyasen D., Singh V.: *An Efficient Evaluation of Requirements Engineering Maturity Measurement Framework For Medium and Small Scale Software Companies*. Proceedings of the 3rd National Conference; INDIA Com-2009, Computing For Nation Development, February 26-27, 2009.
- [ShSM11] Shahid M., Ibrahim S., Mahrin M.: *An Evaluation of Requirements Management and Traceability Tools*. World Academy of Science, Engineering and Technology 78, 2011, <http://www.waset.org/journals/waset/v54/v54-117.pdf> [dostęp: 28.11.2012].
- [Somm03] Sommerville I.: *Inżynieria oprogramowania*. WNT, Warszawa 2003.
- [Somm05] Sommerville I.: *Integrated Requirements Engineering: A Tutorial*. IEEE Software, 2005.
- [SoRa05] Sommerville I., Ransom J.: *An Empirical Study of Industrial Requirements Engineering Process Assessment and Improvement*. ACM Transactions on Software Engineering and Methodology. 14(1).
- [SoSa97] Sommerville L Sawyer P.: *Requirements Engineering: A Good Practice Guide*. Wiley, Chichester, 1997.
- [StGr04] Standish Group: *Charting the Seas of Information Technology – Chaos*. The Sandish Group International, West Yarmouth 1994.
- [Tra108] Tripathi S., at all: *An Efficient Evaluation of Requirements Engineering Process Maturity Assessment and Improvement*. Proceedings of the 2nd National Conference; INDIA Com-2008, Computing For Nation Development, February 08-09, 2008.
- [Wieg03] Wiegers K.: *Software Requirements*. Microsoft Press, 2003.
- [WWW1] <http://www.iag.biz/business-analysis-resources/landing/self-assessment-tool.html> [dostęp: 28.11.2012].
- [WWW4] <http://www.agilealliance.org> [dostęp: 28.11.2012].
- [XuSS06] Xu H., Sawyer P., Sommerville I: *Requirement Process Establishment and Improvement from the Viewpoint of Cybernetics*. „The Journal of Systems and Software” 2006, 79.
- [Youn08] Young I.: *Mental Models. Aligning Design Strategy with Human Behavior*. Rosenfeld Media, 2008.

Załącznik 1

Zestawienie wybranych technik inżynierii wymagań

Technika	Mocne strony	Słabe strony
<p>Wywiad</p> <p>Dla zewnętrznego obserwatora przypomina rozmowę, jednak rolę są różnicowane. Osoba przeprowadzająca wywiad – ankieter dąży do pozyskania informacji oraz może posługiwać się wcześniej przygotowanymi pytaniami tzw. wywiad ustrukturalizowany lub jedynie dyspozycjami do wywiadu, czyli luźno sformułowanymi problemami – tzw. wywiad swobodny, nieustrukturalizowany (sytuacja bardziej zbliżona do naturalnej rozmowy). Osoba odpowiadająca – respondent stara się przekazać wiedzę w odpowiedzi na pytania. Pytania mogą być zamknięte, gdy zawierają określone możliwości odpowiedzi tzw. kafeterie (konkretne lub dysjunktywne) lub otwarte gdy respondentowi daje się swobodę odpowiedzi, a ankieter przytacza odpowiedź dosłownie.</p>	<ul style="list-style-type: none"> • możliwość pozyskania bogatych oraz szczegółowych danych; wywiad penetruje zakamarki zagadnienia • pozwala na rozwinięcie / uzupełnienie wiedzy ankietera • u respondenta rozwija się zaangażowanie oraz zrozumienie projektu • naturalna komunikacja z uwzględnieniem elementów niewerbalnych ułatwia podejmowanie / zrozumienie trudnych problemów, otwartość oraz szczerą wypowiedź. 	<ul style="list-style-type: none"> • wymaga znaczącego zaangażowania czasu oraz wysiłku, głównie ankietera, ale także respondenta • wymaga zaangażowania osób o wysokich umiejętnościach w obszarze komunikacji interpersonalnej • trudna do zastosowania w przypadku dużej ilości zróżnicowanych interesariuszy – potrzeba wspomaganie poprzez ankiety • efekty uboczne, takie jak społecznych oczekiwani, halo, Rosenthala, ...
<p>Wywiad grupowy zogniskowany</p> <p>Druga co do powszechności wykorzystania, po przypadkach użycia, technika zaliczana do jakościowych metod gromadzenia danych, polegająca na przeprowadzeniu i analizie kontrolowanej dyskusji „kolektywnej konwersacji” na dany temat pomiędzy uczestnikami małych (6-12 osób) starym selekcyjnym grup. Zwykle nie dążymy do uzyskania konsensusu, raczej chodzi o prezentację/negocjacje znaczeń, idei, opinii, oczekiwań, obserwacji, postaw oraz zachowań.</p>	<ul style="list-style-type: none"> • wspierają kreowanie wizji, formułowanie pytań oraz weryfikowanie koncepcji, gdy brak jest idei, hipotez, wiedzy, danych • możliwość obserwowania oraz wpływanie na zachowania w grupie • możliwość bardziej efektywnego wykorzystania czasu niż w przypadku wywiadu, tematy bardziej wszechstronnie naświetlone 	<ul style="list-style-type: none"> • wysokie kwalifikacje osób prowadzących oraz interpretujących wyniki • rezultaty różnią się zazwyczaj od wcześniejszych oczekiwań, mogą być specyficzne • wkład uczestników jest często różnicowany • szczegółowe, specyficzne, sensoryczne zagadnienia mogą okazać się nieodpowiednie

¹ tzw. fokus od ang. Focus Group Interview FGI

cd. załącznik I

<p>Etnografia</p> <p>Rozległy obszar wiedzy o kontekstualizacji, badania prowadzone są na podstawie zapisywania wyników bezpośrednich obserwacji. Technika uznawana jako niekompleksowa, zazwyczaj łączona z innymi technikami np. wywiadem, prototypowaniem lub analizą przypadków użycia. Opisano wiele rozszerzających strategii, takich jak np. obserwacji (jawnej, niejawnej, uczestniczącej, nieuczestniczącej, shadowing) i fotografii socjologicznej, fotografii dnia roboczego, obserwacje migawkowe, grupy fokusowe z elementami badania etnograficznego – wywiady grupowe, na które respondenci przynoszą przedmioty, które stają się bodźcem do dyskusji.</p>	<ul style="list-style-type: none"> ● pozwala na weryfikację faktów oraz założeń ● umożliwia precyzyjne pomiary ● pomaga odkrywać niejawne oddziaływania oraz wymagania (wiedza ukryta) ● ułatwia analitykom pracę nad wymaganiami (wizualizowanie, zwiększenie zainteresowania, zrozumienia, procesy pamięci) ● zwykle ponosimy relatywnie niewielkie koszty 	<ul style="list-style-type: none"> ● ludzie mogą świadomie lub nieświadomie realizować inaczej swoje działania, gdy są obserwowani ● zaobserwowane przypadki mogą nie być typowe ● obserwując złożone obszerne procesy można wyciągnąć mylne, niekompletne wnioski ● w pewnych przypadkach niemożliwa lub niepraktyczna
<p>Warsztat wymagań</p> <p>Doświadczenia praktyki wskazują na potrzebę zorganizowania warsztatu wymagań, podczas którego kluczowi interesariusze spotykają się „pod przywództwem” moderatora, aby intensywnie pracując, z wykorzystaniem technik wspomagania kreatywności, w ciągu 1-2 dni wypracować konsensus będący podstawą definicji wymagań. W wyniku warsztatów powstają ściśle zawczasu określone rezultaty.</p>	<ul style="list-style-type: none"> ● pomaga w budowaniu zespołu, którego członkowie zobowiązali się dążyć do sukcesu projektu ● wszyscy interesariusze mają możliwość wypowiedzenia się, kształtuje się polityka realizacji projektu, rozwija się zrozumienie wspólnie przyjętej definicji systemu oraz modeli wymagań, zwiększa się innowacyjność projektu 	<ul style="list-style-type: none"> ● czynniki krytyczne związane z przygotowaniem warsztatów oraz oddziaływaniem osoby prowadzącej ● dwa wątki osób reprezentujących stronę biznesową oraz osób tworzących oprogramowanie ● niezbyt przydatne w przypadku małych projektów o niewielkim ryzyku
<p>Przypadki użycia</p> <p>Koncepcję przypadków użycia stworzył Ivar Jacobson w latach 80. Podstawą konstrukcji jest scenariusz interakcji, a ponadto identyfikator, nazwa, specyfikacja aktorów, wyjątków i rozszerzeń – w wersji podstawowej. W latach 90. przypadki użycia zostały włączone do języka UML oraz uzyskały atrakcyjną prostą reprezentację graficzną. Cieszą się w dalszym ciągu wielkim zainteresowaniem (są używane w ponad 50% projektów) oraz rozwijają się w nawiązaniu do rozwoju koncepcji user stories oraz orientacji aspektowej, o czym Ivar Jacobson pisze na swojej stronie internetowej.</p>	<ul style="list-style-type: none"> ● możliwy jest prosty opis wymagań, nawet dla dużych systemów, zróżnicowani interesariusze mogą uczestniczyć oraz wnieść własny wkład ● dobrze komponują się w proces rozwoju systemu, stopniowo są uszczegóławiane oraz doskonalone, szczególnie dobrze integrują się z testowaniem oraz szacowaniem i śledzeniem prac 	<ul style="list-style-type: none"> ● z przypadkami użycia wiąże się wiele mitycznych opinii, które utrudniają ich stosowanie, takich jak: że są techniką dekompozycji, że są odpowiednią i wystarczającą techniką do opisu wszystkich wymagań, że służą tylko do analizy wymagań lub że ta analiza musi być nadmiernie pracochłonna

cd. załącznik 1

<p>Prototypy</p> <p>Podkreśliły różnorodność, szeroki zakres badań i doświadczeń nad wykorzystaniem wstępnych wersji, pierwowzorów przyszłego rozwiązania. Kontinuum prototypów rozważa się w czterech wymiarach: zakres, funkcjonalność, interakcyjność, dizajn. Prototypy o niewielkim zaawansowaniu, określone terminem low-fidelity, obrazują koncepcję, alternatywne rozwiązania, układ ekranu. Ważne znaczenie mają w tej grupie tzw. papierowe prototypy tworzone przy pomocy szkieletów odręcznych, naklejek, oprogramowania wspomagającego. Z kolei prototypy high-fidelity są zazwyczaj w pełni interaktywne, stanowią funkcjonalności produktu podstawowego, często są rozwijane z wykorzystaniem wyspecjalizowanych narzędzi (Smalltalk, Visual Basic, wzorce, SharePoint). Podział na prototypy horyzontalne oraz wertykalne jest z kolei związany z układem dwuwymiarowym (zakres, funkcjonalność). Szersza perspektywa podejmuje problem interakcji wzajemnych oraz szczegóły dizajnu.</p>	<ul style="list-style-type: none"> ● dostarcza konkret, wokół którego ogniskuje się dialog interesariuszy ● pozwala na koncentrowanie się na wybranych aspektach wymagań ● zwiększa zaangażowanie użytkownika, przeciwdziała efektom ubocznym utrudniającym wywiad, w szczególności papierowe prototypy pozwalają na aktywne doskonalenie założeń przez użytkownika ● szczególnie odpowiedni w przypadku tzw. „trudnego użytkownika”, który nie jest w stanie sformułować precyzyjnych, abstrakcyjnych założeń ● prototypy low-fidelity mogą być rozwijane przy niewielkich kosztach 	<ul style="list-style-type: none"> ● zwiększa wymagania wobec projektantów, w szczególności działania muszą być efektywne (trudne bo informatycy z reguły są perfekcjonalistami), narzuca reżim krótkich wydań, znajomości różnorodnych narzędzi, umiejętności komunikacyjne ● często spotykamy efekt „to mi już wystarczy”, związany z inspirowaną zwykłe przez użytkownika chęcią porzucenia na aktualnym rozwiązaniu, bez zrozumienia, że prototyp nie spełnia wymagań np. skalowalności i rozwojowości ● przy stosowaniu prototypów high-fidelity dochodzą nowe wymiary ryzyka
<p>Modelowanie biznesowe</p> <p>Mimo swojej powszechności pogląd, że możliwe jest wykorzystanie technologii do wyeliminowania niedoborów organizacji nie kwestionując struktury organizacji jako całości, okazuje się zażwyczaj fałszywy. Jest to technika symetryczna do prototypowania, w tym sensie, że wspiera zwiększenie zrozumienia i aktywności przez analityków współpracujących z użytkownikami.</p>	<ul style="list-style-type: none"> ● zwiększa się zrozumienie problematyki, problemów oraz potrzeb interesariuszy ● uproszczone, wyeliminowane lub uzasadnione zostają procedury co do których mogłyby powstać wątpliwości ● wyłaniają się liderzy chętni i zdolni do współpracy w kolejnych etapach 	<ul style="list-style-type: none"> ● słabo rozwinięte są podstawy metodyczne w zakresie integrowania modelu organizacyjnego z aktywnościami inżynierii wymagań ● rozległa reinkynieria poprzedzająca prace projektowe jest złożonym przedsięwzięciem opartym na znacznym ryzykiem

Załącznik 2

Dobre praktyki inżynierii wymagań z podziałem na obszary oraz poziomy

Obszary dobrych praktyk	Podstawowy (36)	Pośredni (21)	Zaawansowany (9)
I. Dokument wymagań	2 <ul style="list-style-type: none"> • zdefiniuj strukturę standardowego dokumentu specyfikacji wymagań • wyjaśnij jak korzystać z dokumentu • dołącz streszczenie wymagań • utwórz uzasadnienie biznesowe • zdefiniuj specjalistyczne terminy • zadbaj o czytelność dokumentu • pomóż czytelnikowi znaleźć informacje • uczyni dokument łatwym do zmiany 	3 -	4 -
II. Kreowanie wymagań	<ul style="list-style-type: none"> • oceń wykonalność systemu • bądź wrażliwy na kwestie organizacyjne i polityczne • zidentyfikuj i konsultuj się z udziałowcami • utrzymuj źródła wymagań • zdefiniuj środowisko operacyjne • używaj listy kontrolne przy odkrywaniu wymagań 	<ul style="list-style-type: none"> • wyszukaj ograniczenia dziedzinowe • utrzymuj uzasadnienia wymagań • wyszukując wymagania patrz z różnych punktów widzenia • prototypuj słabo zrozumiałe wymagania • wykorzystuj scenariusze przy odkrywaniu wymagań • definiuj procesy operacyjne 	<ul style="list-style-type: none"> • wielokrotnie używaj te same wymagania

cd. załącznik 2

1	2	3	4
III. Analiza i negocjowanie wymagań	<ul style="list-style-type: none"> • zdefiniuj granice systemu • używaj listy kontrolne do analizy wymagań • zaoptymizuj się w oprogramowanie do wspomaganie negocjacji • zaplanuj postępowanie w przypadku konfliktów i ich rozwiązywanie • nadaj priorytety wymaganiom 	<ul style="list-style-type: none"> • klasyfikuj wymagania wykorzystując wielowymiarowe podejście • użyj macierzy interakcji, aby wykryć konflikty i nakładające się wymagania 	<ul style="list-style-type: none"> • przypisz ryzyko do wymagań
IV. Opis wymagań	<ul style="list-style-type: none"> • zdefiniuj standardowe szablony do opisu wymagań • używaj prostego i ścisłego języka • adekwatnie używaj diagramów • uzupełnij opis w języku naturalnym innymi sposobami opisu 	<ul style="list-style-type: none"> • specyfikuj wymagania ilościowo 	<p style="text-align: center;">-</p>
V. Modelowanie systemu	<ul style="list-style-type: none"> • utwórz komplementarne modele systemu • modeluj otoczenie systemu • modeluj architekturę systemu 	<ul style="list-style-type: none"> • używaj metod strukturalnych do modelowania systemu • używaj słownika danych • dokumentuj powiązania pomiędzy interesariuszami a modelami systemu 	<p style="text-align: center;">-</p>
VI. Walidacja wymagań	<ul style="list-style-type: none"> • sprawdź zgodność dokumentu wymagań z twoimi standardami • zorganizuj formalne inspekcje wymagań • wykorzystuj interdyscyplinarne zespoły do przeglądu wymagań • zdefiniuj listy kontrolne dla walidacji wymagań 	<ul style="list-style-type: none"> • użyj prototypowania, aby animować wymagania • napisz szkic podręcznika użytkownika • zaproponuj przypadki testowe dla wymagań 	<ul style="list-style-type: none"> • stosowanie notacji formalnych • parafrazuj modele systemu

cd. załącznik 2

1	2	3	4
<p>VII. Zarządzanie wymaganiami</p>	<ul style="list-style-type: none"> • jednoznacznie identyfikuj wymagania • zdefiniuj politykę zarządzania wymaganiami • zdefiniuj politykę śledzenia • utrzymuj przewodnik śledzenia 	<ul style="list-style-type: none"> • użyj bazy danych do zarządzania wymaganiami • zdefiniuj zarządzanie zmianą • zidentyfikuj globalne wymagania systemowe 	<ul style="list-style-type: none"> • identyfikuj wymagania ulotne • przechoduj odrzucone wymagania
<p>VIII. Inżynieria wymagań dla systemów krytycznych</p>	<ul style="list-style-type: none"> • utwórz listę kontrolną dla wymagań bezpieczeństwa • zaangażuj zewnętrznych recenzentów w procesie walidacji 	<ul style="list-style-type: none"> • identyfikuj i analizuj zagrożenia • z analizy zagrożeń wyprowadź wymogi bezpieczeństwa • sprawdź powtórnie wymagania operacyjne i funkcjonalne wobec wymagań bezpieczeństwa 	<ul style="list-style-type: none"> • specyfikuj system przy użyciu metod formalnych • zbieraj opisy incydentów • wyciągaj wnioski z opisów incydentów • dbaj o rozwój kultury bezpieczeństwa w organizacji

Źródło: Na podst. [SoSa97].

Załącznik 3

Przypomnienia pryncypiów podejścia zorientowanego na użytkownika

- I. **Projekt opiera się na dogłębnym zrozumieniu użytkowników, zadań oraz otoczenia.** Powinni zostać zidentyfikowani interesariusze związani z projektem, uwzględniając tych, na których system będzie oddziaływał (bezpośrednio lub pośrednio). Charakterystyki użytkowników, zadań oraz otoczenia są nazywane **kontekstem wykorzystania**¹. Kontekst wykorzystania jest głównym źródłem informacji dla ustalenia wymagań, dostarcza zasadniczy wkład w procesie projektowania. W komentarzach do tego pryncypium często wskazuje się na trójkę {użytkownik, system, otoczenie}, którą trzeba rozpoznać w trakcie prac projektowych.
- II. **Użytkownicy biorą udział w całym procesie projektowania i rozwoju.** Zaangażowanie użytkowników w projektowanie pozwala na pozyskanie wartościowej, źródłowej wiedzy o kontekście wykorzystania, o zadaniach, a także jak użytkownicy mogą pracować w przyszłości z produktem, systemem lub usługą. Powinno ono być aktywne niezależnie czy użytkownik uczestniczy w projektowaniu, czy jest źródłem istotnych danych lub czy ocenia rozwiązania. Poprzez zaangażowanych w całym procesie projektowania i rozwoju użytkowników organizacja zamawiająca system ma możliwość oddziaływania na rozwiązanie. Takie zaangażowanie oraz współudział może mieć wpływ na poziom akceptacji oraz późniejszego wykorzystania.
- III. **Projekt jest prowadzony i udoskonalany z wykorzystaniem ocen zorientowanych na użytkownika.** Oceny projektów oraz ich poprawianie na podstawie otrzymanego od użytkowników sprzężenia zwrotnego pozwalają minimalizować ryzyko, polegające na tym, że wykonany system nie będzie zaspakajał potrzeb użytkownika lub potrzeb organizacji (włączając wymagania ukryte lub trudne do jawnego wyspecyfikowania). Wstępne rozwiązania są weryfikowane w ramach zbliżonych do sytuacji rzeczywistych scenariuszy, wyniki są podstawą progresywnie poprawianych rozwiązań. Zorientowane na użytkownika ocenianie powinno być również wykorzystane w ramach testowania, końcowej oceny projektu, a także na etapie eksploatacji, dając podstawę dla przyszłych projektów.

¹ Ang. *context of use*.

cd. załącznik 3

IV. Proces jest iteracyjny. Bez iteracji nie można zazwyczaj osiągnąć najbardziej odpowiedniego projektu dla systemu interaktywnego. Wiele potrzeb oraz oczekiwań u użytkowników oraz u innych interesariuszy, pojawia się jedynie w czasie rozwoju, gdy projektanci doskonalą swoje zrozumienie użytkowników i ich zadań, gdy w obliczu potencjalnych rozwiązań u użytkowników doskonalili się zdolności wyrażania potrzeb. W celu minimalizacji ryzyka, że rozwijany system nie spełni oczekiwań użytkownika, gdy nowe informacje są otrzymywane, opisy, specyfikacje i prototypy są zmieniane oraz poprawiane. Tak więc iteracje są wykorzystywane do stopniowej eliminacji niepewności oraz ryzyka. W humorystycznych komentarzach do tego pryncypium wskazuje się, że aby dowiedzieć się czego użytkownik potrzebuje można mu na początek dać to czego zapewne nie potrzebuje (nasz pierwszy projekt) oraz obserwować jego reakcję.

V. Projekt odnosi się do całego doświadczenia użytkownika. Kontekst użyteczności rozpatrujemy szerzej niż jedynie łatwość użycia produktu. Patrząc z punktu widzenia osobistych celów użytkownika, może on obejmować zarówno percepcyjne oraz emocjonalne aspekty zazwyczaj połączone z doświadczeniami użytkownika, jak również zagadnienia, takie jak satysfakcja z pracy oraz eliminowanie monotonii. Mocne strony użytkowników, ich ograniczenia, preferencje i oczekiwania powinny być brane pod uwagę podczas określania, jakie działania są przeprowadzane przez użytkowników i które funkcje są realizowane przez technologię. Tego pryncypium w ISO 13407 nie było. Dołączono je zapewne, dla podkreślenia, konieczności jeszcze bardziej szczegółowego analizowania kontekstu użytkownika.

VI. Zespół projektowy wykorzystuje umiejętności oraz spojrzenie interdyscyplinarne. Zorientowany na użytkownika zespół projektowy nie musi być liczny, ale powinien być wystarczająco różnorodny, aby w wymaganym czasie wypracowywać kompromisowe decyzje dotyczące projektu oraz jego wdrożenia. W czasie współpracy w zespołach, których członkowie mają rozległy zestaw umiejętności wyzwała się dodatkowa kreatywność oraz pomysły, które są wykorzystywane w projekcie. Dodatkowa korzyść z interdyscyplinarności oraz ze spojrzenia z różnych perspektyw² polega na tym, że członkowie zespołu stają się bardziej świadomi ograniczeń oraz realiów innych dyscyplin; dla przykłady eksperci techniczni stają się bardziej wyczuleni na problemy użytkowników, zaś użytkownicy mogą zyskać większą świadomość ograniczeń technicznych.

² Ang. *Multi-perspective approach*.

ANALYSIS OF SOME CONCEPTS IN THE FIELD OF DESIGN REQUIREMENTS

Summary

On the eve of the new millennium, the design requirements continue to make significant difficulties to designers. This article aims to analyze three research trends: software engineering, user-oriented approach and agile approach. The paper is to identify, describe, and to discuss mechanisms selected from among those trends.

Stanisław Stanek*
Edyta Abramek
Radosław Kowal

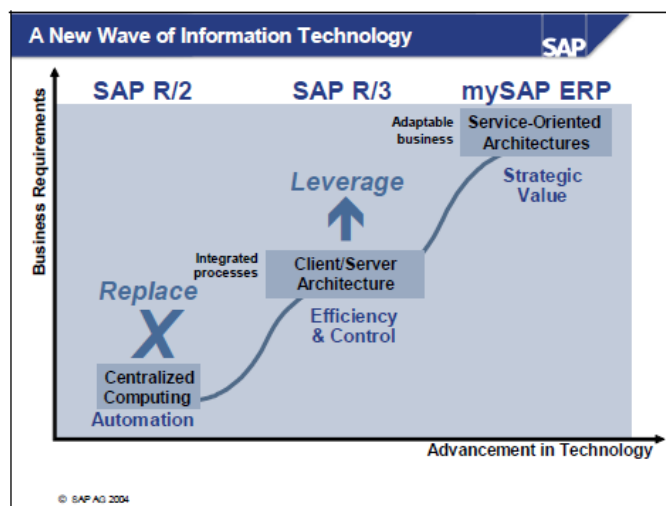
PROJEKTOWANIE SYSTEMÓW INTERORGANIZACYJNYCH

Wprowadzenie

Projekty integracyjne, realizowane już od najwcześniejszych zastosowań informatyki w biznesie, cieszą się niejednokrotnie znaczącym zainteresowaniem zarządzających i technologów z uwagi na szczególną istotność oraz złożony charakter. Udane aplikacje umożliwiają uzyskanie cennego efektu synergii. Tadeusz Kasprzak rozwój serii publikacji podejmujących problematykę integracji charakteryzuje następująco: „Zapoczątkowana publikacjami z początku lat dziewięćdziesiątych ubiegłego wieku historia serii wydawniczej *Studia Informatyki Gospodarczej* jest zarazem historią integracji systemów informacyjnych, która początkowo koncentrowała się wokół hasła komputerowo zintegrowanego wytwarzania (CIM – Computer Integrated Manufacturing), a następnie zajęła się re-inżynierią procesów łamiącą tradycyjne wewnętrzne struktury przedsiębiorstw. (...) O ile w pierwszej fazie integracji komputerowych systemów informacyjnych dominował kierunek zgodny z duchem sterowania i planowania produkcji, a więc integracja pionowa, aktualnie dzięki korzystaniu z technologii Internetu wkraczamy w fazę integracji poziomej – między firmami.” [Kasp06, s. 9-10]. Znamienna jest również opinia dotycząca cenności udanego rozwiązania integracyjnego prezentowana w materiałach firmy SAP. Platforma integracyjna Netweaver wspierająca koncepcję otwartej architektury ESA (Enterprise Service Architecture) powstała w nawiązaniu do rozwoju wymagań użytkowników funkcjonujących w globalnej, dynamicznie zmieniającej się gospodarce, potrzebujących narzędzi bardziej elastycznych, skalowalnych wspierających innowacyjność, gospodarkę wirtualną oraz redukujących koszt funkcjonowania TCO

* Prof. nadzw. dr hab. inż. Stanisław Stanek, Wyższa Szkoła Oficerska Wojsk Lądowych im. generała Tadeusza Kościuszki, 51-150 Wrocław, ul. Czajkowskiego 109, e-mail: stan_stanek@neostrada.pl

(Total Cost of Ownership). Wcześniejsze doświadczenia wskazywały, że tak znaczące zmiany w technologii mogą spowodować konieczność całkowitej wymiany podstawowej infrastruktury IT. W przypadku przejścia od SAP R/3 do mySAP ERP nie okazało się to konieczne. Wykorzystanie architektury SOA (por. rys. 1) pozwoliło na włączenie już zrealizowanych inwestycji w R/3 o wartości 200 mld \$ oraz na dalszy ewolucyjny rozwój systemu [por. SAP004].



Rys. 1. Integracja z wykorzystaniem SOA w mySAP ERP

Źródło: [SAP004].

Problematykę podstaw teorii integracji systemowej prezentuje Jacek Ochman, wskazując m.in. na to, że „(...) integracja polega na łączeniu w całość. Istotą integracji jest zatem utworzenie nowej całości, której elementy są połączone określonymi relacjami i są powiązane odpowiednim stopniem zależności od całości. (...) Z problemów integracji najbardziej istotne są jej cele i zadania oraz procesy integracyjne.” [Ochm92, s. 11]. Badania nad systemami interorganizacyjnymi (IOS) zostały zapoczątkowane artykułem Felixa Kauffmana [Kauf66] w Harvard Business Review, w którym autor zauważył, że powiązania wzajemne między systemami komputerowymi różnych organizacji mogą powodować rozległe zmiany w systemach operacyjnych oraz mogą przyczynić się do podnoszenia wydajności organizacji. Termin system interorganizacyjny¹ IOS rozumiany jako zautomatyzowany system informacyjny współdzielony przez dwie lub więcej firm wprowadzili Barret oraz Konsynski [BaKo82]. Czego

¹ InterOrganizational Systems IOS.

można się nauczyć z przeszłości, co pomoże nam zrozumieć, jak budować udane cyfrowe ekosystemy dzisiaj? Lynda Applegate w hbs@work podsumowuje doświadczenia następująco:

- Wykorzystaj wewnętrzne zasoby. Udana cyfrowe ekosystemy tworzą więzi z klientami, dostawcami i partnerami jako rozszerzenie platformy IT i procesów biznesowych, które są wykorzystywane do koordynacji i zarządzania działalnością wewnątrz firmy.
- Wykorzystaj gospodarczą wartość cyfryzacji. Podczas gdy aktywa rzeczowe często tracą wartość w trakcie użytkowania, wartość zasobów cyfrowych rośnie. Wyłaniający się otwarty standard, „na żądanie” architektury IT i modeli biznesowych zapewnia, że wszyscy członkowie cyfrowego ekosystemu mogą tworzyć, udostępnić i w pełni wykorzystywać wartości ekonomiczne zasobów cyfrowych.
- Wyłaniaj propozycje korzystne/wygrywające dla wszystkich. Podstawą biznesu elektronicznego jest przekonanie, że KAŻDY członek uzyska korzyści przewyższające wkłady, niezależnie od tego czy sprzedaje książki klientom, czy współpracuje przy budowie najbardziej zaawansowanego samolotu.
- Sprawne odkrywanie oraz wzmacnianie uzdolnień. Rozszerzone przedsiębiorstwa i sieci dostarczają wartości przez umożliwienie każdemu członkowi koncentrowania się na tym co robi najlepiej, przyczyniając się przy tym do większego dobra ogółu. Ze wzrostem kapitału społecznego rośnie również wartość ekonomiczna całości [WWW1].

Jako katalizator oddziałujący na projektowany cyfrowy ekosystem należy z pewnością analizować infrastruktury informacyjne państw (por. np. [Olen06]). Jest to istotny wyróżnik rozważanej klasy systemów, który w tym artykule będzie traktowany jako parametr procesów decyzyjnych. W nawiązaniu do problematyki IOS, przedstawionej w artykule [Stan09], zostały podniesione tutaj problemy celów i zadań systemów tej klasy. Niniejszy artykuł omawia doświadczenia oraz rekomendacje dotyczące technologii oraz procesów. Problematyka okazała się tak obszerna, że zagadnienie planowane pierwotnie na jeden artykuł, zaprezentowano w dwóch².

1. Aspekty istotne dla tworzenia systemów interorganizacyjnych

Systemy interorganizacyjne to takie systemy, które wychodzą poza granice danej organizacji gospodarczej. Ich zadaniem jest łączenie przedsiębiorstw ze sobą (minimum dwóch) w celu umożliwienia współdzielenia zasobów, sprawnego przepływu danych lub informacji pomiędzy nimi, lepszej koordynacji działań itp.

² S. Stanek, E. Abramek, R. Kowal: *Projektowanie systemów interorganizacyjnych* oraz R. Tyrała: *Elektroniczna wymiana danych na przykładzie systemu BPSC Impuls 5*.

Projektując tego rodzaju systemy, należy brać pod uwagę różne poziomy (konteksty) integracji, jak np. interorganizacyjny i organizacyjny oraz różne aspekty integracji: prawne, społeczne, technologiczne oraz ekonomiczne, czyli analizować koszty i korzyści wynikające z wdrożenia tego typu systemów, na co zwrócono uwagę w artykule K. Bouchbouta i Z. Alimazighiego (por. [BoAl08, s. 348]).

Na poziomie interorganizacyjnym należy rozważyć: jakiego rodzaju dane lub dokumenty wymieniają pomiędzy sobą przedsiębiorstwa; które systemy dostarczają, generują te dane lub dokumenty, a które systemy będą tych danych lub dokumentów potrzebowały. Poziom organizacyjny odnosi się do tego, w jaki sposób przebiega proces wspierany przez system interorganizacyjny; jakie jest wsparcie procesu ze strony obu organizacji; czy rozwiązanie zapewni poprawę obsługi klientów. Istotne jest również to, jak zostaną rozwiązane aspekty prawne, kwestie dotyczące własności intelektualnej.

Aspekt technologiczny dotyczy tego, jaka technologia zostanie wykorzystana w celu zapewnienia integracji; jakie zostaną zastosowane standardy; czy zostaną wykorzystane rozwiązania autorskie. Ponadto, konieczne jest określenie jaki sprzęt jest wymagany do budowy systemu interorganizacyjnego. Jakie są wymagania w odniesieniu do sieci komputerowej, bezpieczeństwa danych. Czy rozwiązanie zapewni, przykładowo, redukcję czasu wyszukiwania niezbędnych informacji. Należy również uwzględnić aspekty rozwojowe systemu – czy system może być dalej rozbudowywany w następnych przedsiębiorstwach, o kolejne systemy, podsystemy, usługi.

Warto również zwrócić uwagę na to, czego dotyczy sama integracja. Czy jest to integracja na poziomie sprzętowym, na poziomie danych, na poziomie systemów informatycznych (oprogramowania systemowego, aplikacyjnego lub komunikacyjnego), czy też na poziomie interfejsu systemu informatycznego.

Aspekt finansowy dotyczy tego, jakiego rodzaju inwestycje muszą poczynić firmy, np. zakup sprzętu i oprogramowania, koszty związane ze szkoleniami i utrzymaniem systemu.

Aspekt ludzki dotyczy tego, jakie osoby po stronie każdej z organizacji będą odpowiedzialne za proces integracji, czy konieczne będą specjalistyczne szkolenia tych osób.

Istotny jest również aspekt ekonomiczny. Systemy interorganizacyjne z założenia są budowane po to, aby dostarczać korzyści gospodarczych. Korzyści te są wynikiem relacji międzyorganizacyjnych postrzeganych jako nowy zasób strategiczny przedsiębiorstwa (por. [Maci09, s. 27]).

Pojęcie integracji może odnosić się do systemów zarządzania, procesów biznesowych, systemów informatycznych, ich podsystemów, metod, dyscyplin naukowych, powiązań człowieka z narzędziem pracy. Świadczy to o interdyscyplinarności zagadnienia (por. [OlSr01, s. 35]).

W odniesieniu do systemów informatycznych wyróżnia się wiele różnych podziałów i rodzajów integracji. Przykładowo, w modelu Francois Vernadata (por. [Vern99]) występują trzy poziomy integracji: fizyczna, aplikacji i biznesowa.

Integracja na poziomie fizycznym dotyczy komunikacji między systemami informatycznymi, reguł wymiany danych za pomocą interfejsów programistycznych, sieci komputerowych, protokołów komunikacyjnych. Wiąże się z utworzeniem standardu OSI/ISO oraz protokołów sieciowych, stanowiących podstawę funkcjonowania sieci komputerowych, w tym Internetu.

Integracja na poziomie aplikacji i danych dotyczy współdziałania aplikacji na różnych platformach sprzętowych i programowych. Dotyczy problemu wspólnego użytkowania danych (zasobów danych) przez różne aplikacje. Do standardów wymiany danych należą STEP, EDI, COM/DCOM, ActiveX, CORBA, .NET Remoting, Enterprise JavaBeans. Ich słabą stroną jest brak wzajemnej kompatybilności oraz wykorzystywanie binarnego formatu wymiany danych, który nie sprawdza się w Internecie. W odpowiedzi na te trudnienia opracowano standard Web services. Z integracją na poziomie aplikacji są związane zagadnienia wspólnych usług sieciowych (integracja na poziomie usług sieciowych), przetwarzanie rozproszone oraz budowanie aplikacji przenośnych.

Trzeci, najwyższy poziom integracji stanowi integracja biznesowa (integracja na poziomie procesów biznesowych). Oznacza scalanie i koordynację na poziomie procesów biznesowych w przedsiębiorstwie oraz w jego otoczeniu. Z tym rodzajem integracji są związane zagadnienia pracy grupowej, kooperacji lub koopetycji oraz wirtualizacji procesów biznesowych.

W odniesieniu do systemów informatycznych można zaproponować wiele innych sposobów rozróżniania rodzajów integracji. Jeden z nich obejmuje podział na:

- integrację na poziomie fizycznym, m.in. integracja z wykorzystaniem API,
- integrację na poziomie danych (w tym przesyłania danych), np. ETL i aplikacji np. EAI,
- integrację na poziomie usług sieciowych/zadań/sieci, np. brokerzy wiedzy, SOA, ESB,
- integrację na poziomie procesów biznesowych,
- integrację na poziomie warstwy prezentacji (na poziomie interfejsu użytkownika),
- integrację w znaczeniu organizacyjnym (integracja funkcji, integracja geograficzna i integracja międzyorganizacyjna).

API (Application Programming Interface) to interfejs programistyczny aplikacji. Umożliwia tworzenie rozszerzeń do danego systemu, wykorzystując jego zasoby i logikę. Zawiera zestaw reguł, które określają, w jaki sposób programy komunikują się ze sobą. API definiuje się na poziomie kodu źródłowego. Przy-

kładowo, w API jest wyposażone oprogramowanie dla małych i średnich firm o nazwie SAP Business One. Pozwala ono na import lub eksport danych z systemu SAP. Rozszerzeniami systemu SAP BO zajmuje się m.in. firma Altab. Firma ta w swojej ofercie posiada program Altbank, który jest rozszerzeniem systemu SAP Business One przeznaczonym do zbierania informacji o zobowiązaniach firmy i na ich podstawie eksportowania przelewów do systemów bankowości elektronicznej. Inny program o nazwie AltEDI łączy SAP Business ONE z systemami dostawców lub odbiorców. Jego celem jest wyeliminowanie wielokrotnego wprowadzania danych oraz przyspieszenie i zwiększenie dokładności przepływu informacji pomiędzy systemami informatycznymi firm uczestniczących w wymianie [WWW2]. Innym przykładem jest firma kurierska Opek Sp. z o. o., która wdrożyła system klasy ERP – Microsoft Dynamics NAV w celu prowadzenia zaawansowanych rozliczeń i usprawnienia zarządzania transakcjami. Jednym z rozszerzeń tego systemu jest wprowadzenie automatyzacji rozliczania faktur, zrealizowane przez firmę IT.Integro z Poznania. System komunikuje się z oprogramowaniem banku i wspomaga w zakresie rozliczeń z bankiem: przelewy, potwierdzenie salda, kojarzenie płatności z fakturą i listem przewozowym³ [WWW3].

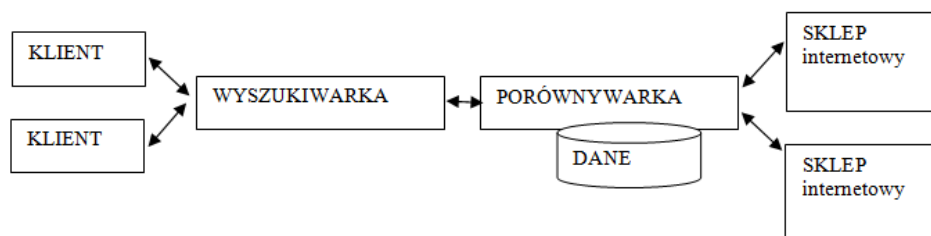
Do innych technik integracyjnych należą ponadto, MOM/JMS (Java Message Oriented Middleware API), JCA (Java Cryptography Architecture) oraz XML / Web services. MOM (Message Oriented Middleware) to architektura oparta na koncepcji asynchronicznej wymiany jednostek danych. Jednym ze standardów realizujących MOM jest m.in. JMS (Java Messaging System).

Standardem transformacji danych jest ETL (Extract, Transform and Load). ETL jest to narzędzie wspomagające pozyskanie danych dla baz danych, w szczególności dla hurtowni danych. Przykładowymi narzędziami ETL są: SAP Business Objects Data Integrator (BODI), Oracle Data Integrator (ODI), SAS ETL Studio, SQL Server Integration Services (SSIS) firmy Microsoft.

Integracja na poziomie aplikacji jest określana jako EAI (Enterprise Application Integration). Rozwiązanie z założenia miało zredukować ilość połączeń w systemie rozproszonym poprzez wprowadzenie jednego centralnego punktu (tzw. pośrednika). Rezultatem EAI są zintegrowane procesy biznesowe. EAI zapewnia łączność pomiędzy procesami biznesowymi różnych aplikacji połączonych ze sobą. Zapewnia współdzielenie danych między wieloma aplikacjami. Ideę EAI realizuje m.in. Microsoft BizTalk Server. Na podstawie Microsoft BizTalk 2002 Server w firmie PSE SA zintegrowano systemy EMPAC (Enterprise MPAC) i SAP (mySAP.com). Integracja przyniosła następujące korzyści: automatyzację wymiany danych w celu wyeliminowania wielokrotnego wprowadzania tych samych danych w różnych systemach używanych w firmie, ogra-

³ Ponad 70 tysięcy rozliczonych faktur miesięcznie dzięki Microsoft Dynamics NAV – Microsoft Case Studium.

niczenie liczby błędów, możliwość definiowania łańcuchów czynności realizowanych kolejno przez połączone aplikacje oraz szybszy dostęp do informacji i analiz sytuacji firmy [WWW4]. Innym przykładem integracji na poziomie aplikacji są porównywarki cen. Są to systemy, które umożliwiają porównywanie cen z dostępnych w Internecie sklepów internetowych na podstawie zawartej z nimi umowy o współpracy. Przykładowa porównywarka cen otrzymuje oferty produktów wraz z cenami. Zadaniem oprogramowania jest kojarzenie ofert cenowych tych samych produktów z różnych sklepów internetowych. Porównywanie ofert bez integracji systemów informatycznych byłoby niemożliwe. Z porównywarek korzystają z jednej strony klienci poszukujący określonego produktu, z drugiej strony sklepy internetowe zamieszczające swoje oferty w serwisie porównywarki (rys. 2). Porównywarka cen pobiera aktualne oferty produktów i ich cen od różnych partnerów handlowych. Klient końcowy otrzymuje listę cen danego towaru w różnych sklepach.



Rys. 2. Schemat poglądowy porównywarki cen

Źródło: [FrSa08].

Ponad warstwą danych i aplikacji znajduje się warstwa usług sieciowych. SOA (Service Oriented Architecture) jest to architektura zorientowana na usługi. Pojawienie się architektury zorientowanej na usługi spowodowało zmianę w zakresie wymiany danych i integracji oprogramowania. Podstawową zasadą rządzącą tym podejściem jest luźne powiązanie elementów systemu, a nie jak do tej pory silne powiązanie ze sobą podsystemów danego systemu. SOA może korzystać z ESB (Enterprise Service Bus). ESB jest to dodatkowa warstwa pośrednia w wielowarstwowej architekturze systemów informatycznych umożliwiająca zastosowanie koncepcji SOA w środowisku korporacyjnym. Umożliwia dynamiczne przyłączanie i odłączanie usług wchodzących w skład systemu. ESB powstało na podstawie koncepcji MOM i EAI. Przykładem środowiska realizującego ideę ESB jest JBoss lub WebSphere.

Dla porównania z SOA, Web service jest to zbiór technologii, takich jak np. SOAP, UDDI, WSDL, XML użytych do realizacji SOA. SOAP to standard wymiany danych, UDDI to technologia pozwalająca na publikację, wyszukiwanie i wiązanie informacji o usługach Web service, WSDL to język stosowany do

opisu interfejsów usług, a XML to język opisu danych. Wyróżnia się oprócz tego standardy dostępu do danych i standardy transformacji danych (np. ETL w hurtowniach danych).

Podczas gdy za pomocą języka WSDL można zdefiniować usługi, interakcja pomiędzy usługami odbywa się za pomocą języka BPEL (Business Process Execution Language). Istnieje wiele graficznych narzędzi ułatwiających modelowanie procesów biznesowych w języku BPEL. Warstwa procesów biznesowych znajduje się ponad warstwą usług sieciowych. Dotyczy integracji na poziomie procesów biznesowych. Klasyczna koncepcja tworzenia systemów informatycznych przebiega następująco – pracownicy działu IT muszą zrozumieć procesy w przedsiębiorstwie, a następnie wdrożyć system informatyczny wspomagający ich realizację. Obecnie dąży się do tego, aby pracownicy firmy mogli samodzielnie projektować lub modyfikować istniejące w firmie procesy biznesowe. Ciężar tworzenia aplikacji jest rozkładany pomiędzy dział IT a użytkowników końcowych systemu.

Integracja dotyczy również warstwy prezentacji, inaczej interfejsu użytkownika. Z tego rodzaju integracją można spotkać się w sytuacji, gdy serwis internetowy posiada kontrolkę flash informującą np. o stanie pogody.

Integracja w znaczeniu organizacyjnym obejmuje integrację funkcji w przedsiębiorstwie, integrację geograficzną i integrację międzyorganizacyjną (interorganizacyjną). Korzystne rezultaty przynosi integracja funkcji systemów ERP i CRM, a systemu CRM z produktami klasy CTI (Computer Telephony Integration) lub modułu „Produkcja” systemu ERP z programami CAD/CAM podczas projektowania produktu, dodatkowo z wykorzystaniem technik PLM (Product Lifecycle Management). Współcześnie istotne znaczenie ma również integracja systemów ERP z systemami SCM (Supply Chain Management), ponieważ o uzyskaniu przez przedsiębiorstwo przewagi konkurencyjnej coraz częściej decydują czynniki związane z jakością funkcjonowania łańcucha wartości. Przedsiębiorstwo może udostępnić wybrane funkcje systemu ERP pozostałym uczestnikom łańcucha wartości (klientom, punktom sprzedaży, dostawcom). Klienci mogą składać swoje zamówienia bezpośrednio u wytwórcy i śledzić stopień ich wykonania. Ponadto, klient może składać zamówienie w dowolnym miejscu świata poprzez Internet. Dzięki temu można mówić o integracji geograficznej.

2. Analiza przypadku

EDI (Electronic Data Interchange) to termin wywodzący się z końca lat 70. i początku 80. XX wieku. Oznacza wymianę sformatowanych wiadomości, reprezentujących dokumenty handlowe, przesyłane między kontrahentami z wy-

korzystaniem środków komunikacji elektronicznej⁴. Od początku elektroniczna wymiana danych miała na celu usprawnienie komunikacji między jednostkami gospodarczymi w kanale dystrybucji. Dostawcy otrzymywali zamówienia od kupujących w postaci pliku danych o odpowiednim formacie oraz odpowiadali wystawiając faktury i dokumenty magazynowe. Te dokumenty były dostarczane z wykorzystaniem początkowo sieci telekomunikacyjnej, a później z wykorzystaniem Internetu.

```
<?xml version="1.0" encoding="utf-8" ?>
<Document-Invoice>
  <Invoice-Header>
    <InvoiceNumber></InvoiceNumber>
    <InvoiceDate></InvoiceDate>
    <SalesDate></SalesDate>
    <InvoiceDuplicateDate></InvoiceDuplicateDate>
    <InvoiceCurrency></InvoiceCurrency>
    <InvoicePaymentDueDate></InvoicePaymentDueDate>
    <InvoicePaymentTerms></InvoicePaymentTerms>
    <InvoicePostDate></InvoicePostDate>
    <DocumentFunctionCode></DocumentFunctionCode>
  <Order>
    <BuyerOrderNumber></BuyerOrderNumber>
    <SupplierOrderNumber></SupplierOrderNumber>
    <BuyerOrderDate></BuyerOrderDate>
  </Order>
  <Delivery>
    <DeliveryLocationNumber></DeliveryLocationNumber>
    <DeliveryDate></DeliveryDate>
    <DespatchNumber></DespatchNumber>
    <DespatchAdviceNumber />
  </Delivery>
</Invoice-Header>
<Invoice-Parties>
  <Buyer>
    <ILN></ILN>
    <TaxID></TaxID>
    <AccountNumber></AccountNumber>
    <Name></Name>
    <StreetAndNumber></StreetAndNumber>
    <CityName></CityName>
```

Rys. 3. Fragment dokumentu „Faktura” w standardzie EDIFACT

⁴ Na podstawie definicji National Institute of Standards and Technology (NIST).

W początkowym okresie nie istniały ogólnie przyjęte standardy dla tworzonych dokumentów. Prowadziło to jednak do powstawania kłopotów z wymianą komunikatów między partnerami handlowymi, którzy nie stosowali tego samego sposobu kodowania. Do rozwiązania tego problemu powołano specjalny komitet w ramach American National Standards Institute (ANSI). Prace komitetu zaowocowały standardem ASC X12. Inny, kluczowy dla EDI standard – United Nations/Electronic Data Interchange For Administration, Commerce and Transport (UN/EDIFACT), został opracowany przez grupę United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT), powołaną pod auspicjami ONZ. Jeszcze innym standardem, używanym głównie w Wielkiej Brytanii, jest TRADACOMS.

Wszystkie te standardy powstały na początku lat 80. XX wieku. Każdorazowo ich celem było ustalenie rodzajów komunikatów, które mogą być przesyłane, listy pól obowiązkowych dla poszczególnych komunikatów, sposobu formatowania dokumentów oraz sposobu kodowania tekstu w dokumencie.

Oprócz standardów komunikatów do wykorzystania EDI, konieczne jest ustalenie sposobu ich przesyłania. Można założyć, że większość komunikacji w ramach EDI wykorzystuje jako medium Internet. Ponadto możliwe są trzy scenariusze, opisujące sposób przekazywania komunikatów między zainteresowanymi stronami:

1. Bezpośrednio – komunikaty wędrują z jednej firmy do drugiej.
2. Z wykorzystaniem platformy pośredniczącej (serwera komunikatów) – w tym przypadku strony korzystają z pośrednika, który od nadawcy odbiera komunikat, sprawdza jego poprawność, jeżeli trzeba dokonuje konwersji na inny format, informuje o wykryciu błędów i finalnie przekazuje do odbiorcy.
3. Z wykorzystaniem dwóch platform pośredniczących – nadawca i odbiorca nie korzystają z tego samego serwera komunikatów. Nadawca wysyła komunikat do własnego serwera komunikatów, a ten z kolei przekazuje do platformy odbiorcy, która dostarcza go do adresata.

Jednym z elementów wdrożenia systemu SAP Business One w firmie Enan Sp. z o.o. jest rozwiązanie dodatkowe, powiększające zestaw standardowych funkcjonalności systemu o obsługę elektronicznej wymiany danych (EDI). Wymóg stworzenia tego typu rozwiązania wynikał z umów handlowych, podpisanych przez firmę Enan z kilkoma sieciami handlowymi, m.in. Auchan, Metro Group (Real, MediaMarkt). Siłą EDI jest umożliwienie partnerom handlowym przesyłania informacji na temat zachodzących w ich ekosystemie zdarzeniach gospodarczych. Sieci handlowe wymagały ponadto, aby komunikaty EDI były odbierane i wysyłane z wykorzystaniem platformy EWA dostarczanej przez firmę EDISON S.A. W celu usprawnienia korzystania z platformy firma EDISON oferuje program o nazwie Connector Plus. Jego zadaniem jest obsługa procesu wysyłania i pobierania dokumentów do i z platformy EWA. Dodatkowo Connector Plus pozwala na monitorowanie statusu poszczególnych komunikatów. Przykładową listę dokumentów odebranych i wysłanych przedstawia rys. 4.

Numer dokumentu	ID	Partner	Data stworzenia	Data odbioru	Data wykończenia	Data dokumentu	Rodzaj dokumentu	Status
604769	1099	BOWREX	29-12-2010 15:29:29	31-01-2011 08:26:35		30-12-2009 00:00:00	Faktura (INVOIC)	D O
8073_33736	8073_337368700	BOWREX	16-12-2010 05:08:23	24-12-2010 02:46:16	16-01-2011 13:54:34	09-09-2010 00:00:00	Faktura (B10)	D O W
8073_33735	8073_337360200	BOWREX	16-12-2010 05:08:22	24-12-2010 02:46:16		09-09-2010 00:00:00	Faktura (B10)	D O

Rys. 4. Lista dokumentów w programie Connector Plus

Źródło: Instrukcja obsługi programu Connector Plus firmy EDISON S.A.

W przypadku firmy Enan wymóg przesyłania informacji dotyczył zamówień od klientów (sieci handlowe) oraz wystawianych na ich podstawie faktur i faktur korygujących. Konieczne było zatem utworzenie mechanizmu pozwalającego na pobieranie komunikatów EDI przychodzących z platformy EWA i zapisanie ich do bazy SAP Business One w postaci dokumentów typu „Zlecenie sprzedaży”. Należało ponadto stworzyć mechanizm, który po dodaniu w SAP Business One dokumentów typu „Faktura sprzedaży” i „Faktura korygująca sprzedaży” automatycznie zapisuje je jako komunikaty EDI, w określonym standardzie. Proces tworzenia dokumentów w akceptowalnym przez Connector Plus formacie nie wymaga interakcji z użytkownikiem. Po prostu po dodaniu określonego dokumentu dla wybranych kontrahentów dokument dla EDI jest tworzony automatycznie w tle.

Problem komunikacji z platformą EWA został rozwiązany poprzez ustalenie, że program Connector Plus sam pobierze dokumenty z określonej lokalizacji w systemie plików, a także zapisze w ustalonej lokalizacji dane, które mają zostać zapisane w bazie SAP Business One.

Do pobierania danych zapisanych przez EWA wykorzystano usługę systemu MS Windows. Usługa ta reaguje na pojawienie się w określonej lokalizacji nowych plików. Po wychwyceniu powstania w systemie plików nowego pliku usługa rozpoczyna konwersję danych zawartych w pliku z formatu EDI na format XML wykorzystywany w SAP Business One.

Po poprawnym zapisaniu dokumentów nazwa pliku zostaje zmieniona. W momencie wystąpienia błędu osoby zaangażowane w realizację procesu biznesowego są z kolei powiadamiane o błędzie przy użyciu poczty elektronicznej.

System SAP Business One posiada własny zestaw obiektów do zapisywania/odczytywania dokumentów z bazy danych. Programista nie jest uprawniony do dokonywania zmian bezpośrednio w tabelach bazy danych. Zamiast tego wykorzystuje tzw. Software Development Kit (SDK), dostarczany wraz z systemem. Wspomniana usługa wykorzystuje obiekty z SDK, poprzez które dokumenty są zapisywane w systemie SAP Business One.

```

<?xml version="1.0" encoding="UTF-16"?>
<BOM>
  <BO>
    <AdmInfo>
      <Object>13</Object>
    </AdmInfo>
    <OINV>
      <row>
        <DocType>I</DocType>
        <CANCELED>N</CANCELED>
        <Handwrkten>N</Handwrkten>
        <Printed>N</Printed>
        <DocStatus>O</DocStatus>
        <InvntSttus>O</InvntSttus>
        <Transferred>N</Transferred>
        <ObjType>13</ObjType>
        <DocDate>20091105</DocDate>
        <DocDueDate>20091105</DocDueDate>
        <CardCode>000049</CardCode>
        <CardName>SATURN PLANET SP. Z O.O. POZNAŃ I SPÓŁKA KOMANDYTOWA</CardName>
        <Address>Al. Jerozolimskie 179
02-222 Warszawa
Poland</Address>
        <NumAtCard>
</NumAtCard>
        <VatPercent>0.000000</VatPercent>
        <VatSum>43.780000</VatSum>
        <VatSumFC>0.000000</VatSumFC>
        <DiscPrct>0.000000</DiscPrct>
        <DiscSum>0.000000</DiscSum>
        <DiscSumFC>0.000000</DiscSumFC>|
        <DocCur>PLN</DocCur>
      </row>
    </OINV>
  </BO>
</BOM>

```

Rys. 5. Fragment dokumentu „Faktura” w formacie XML z SAP Business One

W przypadku utworzenia nowego dokumentu faktury (lub faktury korygującej) informacja o niej jest zapisywana do specjalnie utworzonej tabeli. Tabela ta pełni rolę kolejki, w której są zapisywane informacje o dokumentach, które mają zostać przesłane na platformę EWA. Każdy dokument posiada swój identyfikator oraz status, opisujący czy został już przesłany, czy też nie. Do przeglądania kolejki i zapisywania dokumentów w formacie EDI zostało utworzone zadanie usługi SQL Agent. Zadanie jest uruchamiane co 10 minut. Jednym z elementów zadania jest wywoływanie aplikacji konsolowej z odpowiednimi parametrami. Parametry opisują jaki dokument i gdzie w systemie plików ma zostać utworzony. Aplikacja ma za zadanie pobrać szablon dokumentu EDI, wykonać mapowanie z formatu SAP na EDI i zapisać dokument w odpowiedniej lokalizacji. Po poprawnym wykonaniu zadania jest zmieniany status dokumentu w tabeli zawierającej kolejkę dokumentów do przetworzenia.

Dzięki zastosowaniu opisanego powyżej rozwiązania opracowywanie dokumentów i przesyłanie ich między systemami sieci handlowych i SAP Business One przebiega w sposób w pełni zautomatyzowany. Użytkownicy pełnią jedynie funkcję kontrolną, sprawdzając przyczyny ewentualnych błędów.

Podsumowanie

Potrzebny dalszy rozwój architektury korporacyjnej oraz oprogramowania powinien uwzględniać wymagania nowej gospodarki oraz dotychczasowe doświadczenia we wdrażaniu rozwiązań, wskazujące m.in. na: rosnące znaczenie adaptowalności zmian zachodzących w skali tygodni oraz dni zamiast wcześniejszych lat lub miesięcy, potrzebę koherentnego zespolenia procesów integracji technologii oraz integracji biznesu, wspieranie przemian w organizacjach funkcjonujących w ekosystemie technologii XXI wieku ze zdeorganizowanymi procesami biznesowymi.

W odniesieniu do systemów informatycznych wyróżnia się wiele różnych podziałów i rodzajów integracji.

Projektując systemy interorganizacyjne, należy brać pod uwagę różne poziomy integracji (integracja pionowa i pozioma) oraz różne aspekty integracji: prawne, społeczne, technologiczne oraz ekonomiczne, czyli analizować koszty i korzyści wynikające z wdrożenia tego typu systemów.

Wirtualizacja, globalizacja oraz rosnąca rola przemian strukturalnych w biznesie wpływają na konkurencyjną niezbędność oraz rosnące zapotrzebowanie na IOS. Potencjał IOS może się realizować jedynie, gdy system ten jest w pełni zintegrowany z innym wewnętrznym systemem informacyjnym. Zarówno duże, jak i małe organizacje są zdolne do uzyskania korzyści z IOS.

Dalsze analizy przypadków przedstawiające różne sposoby integracji systemów informatycznych zostały opracowane na podstawie rozwiązań firmy BPSC z Chorzowa oraz przedstawione w następnym artykule⁵.

Ostatnia dekada to rozwijający się postęp we wprowadzaniu IT do zarządzania łańcuchami dostaw, elektronicznej wymiany dokumentacji (EDI), różnorodnych form IOS, takich jak ekstranety, B2B, P2P oraz rynków elektronicznych. Umożliwiają one integrację między partnerami z wykorzystaniem powiązanych technologii (XML, SOAP, WSDL, ESB, ETL, dynamiczny ETL, Cloud Computing).

⁵ Powstały dzięki współpracy Katedry Informatyki Uniwersytetu Ekonomicznego z firmą BPSC SA, z siedzibą w Chorzowie. Autorem analizy przypadku jest Pan Rafał Tyrała z firmy BPSC (artykuł pt. Elektroniczna wymiana danych na przykładzie systemu BPSCV Impuls 5). Przykłady zostały opracowane z uwzględnieniem przedstawionych w niniejszym rozdziale aspektów istotnych z punktu widzenia tworzenia systemów interorganizacyjnych.

Literatura

- [BaKo82] Barret S., Konsynski B.R.: *Interorganizational Information Sharing Systems*. „MIS Quarterly” 1982, Vol. 6, No. 1.
- [BoAl08] Bouchbout K., Alimazighi Z.: *Framework for Identifying the Critical Factors Affecting the Decision to Adopt and Use Interorganizational Information Systems*. „Engineering and Technology” 2008, Vol. 11.
- [FrSa08] Fryźlewicz Z., Salamon A.: *Podstawy architektury i technologii usług XML sieci Web*. MIKOM/Wydawnictwo Naukowe PWN, Warszawa 2008.
- [Kasp06] Kasprzak T.: *W kierunku rozszerzonego przedsiębiorstwa*. Difin, Warszawa 2006.
- [Kauf66] Kauffman F.: *Data Systemsthat Cross Company Boundaries*. „Harvard Business Review” 1966, Vol. 44, No. 1.
- [Maci09] Macias J.: *Relacje międzyorganizacyjne jako nowy zasób strategiczny przedsiębiorstwa*. „Przegląd Organizacji” 2009, nr 12.
- [Ochm92] Ochman J.: *Integracja w systemach informatycznych zarządzania*. PWE, Warszawa 1992.
- [Olen06] Olenski J.: *Infrastruktura informacyjna państwa w globalnej gospodarce*. UW WNE, Warszawa 2006.
- [OISr01] Olszak C., Sroka H.: *Zintegrowane Systemy Informatyczne w Zarządzaniu*. Wydawnictwo AE, Katowice 2001.
- [SAP004] TERP10: ERP Business Process Integration based on R/3 Enterprise, SAP Material number: 50069275, 2004.
- [Sta09] Stanek S.: *Elementy koncepcji i zastosowań interorganizacyjnych systemów informatycznych IOS*. W: *Zintegrowane Systemy Zarządzania ERP w gospodarce wirtualnej*. Red. H. Sroka. Wydawnictwo AE, Katowice 2009.
- [Vern96] Vernadat F.: *Enterprise Modeling and Integration: Principles and Applications*. Springer, London 1996.
- [WWW1] <http://www.exed.hbs.edu/assets/Pages/digital-ecosystems.aspx>
- [WWW2] <http://www.altab.pl>
- [WWW3] http://www.it.integro.pl/referencje/wdrozenie_opek.pdf [dostęp: 28.11.2012].
- [WWW4] <http://www.docstoc.com/docs/40279351/BizTalk-Server-Udana-Integracja-Systemw-EMPAC-SAP> [dostęp: 28.11.2012].

DESIGNING INTERORGANIZATIONAL SYSTEMS

Summary

The chapter comprises discussion on interorganizational system development problems. At first, authors define what interorganizational information system is and in what aspects they could be analyzed. The second part includes consideration of integration problems on different organizational levels. Then authors present firms' IT solutions applicable for integration of information systems. The purpose of this article is to present experiences and recommendations on IOS technologies and processes.

Rafał Tyrała*

ELEKTRONICZNA WYMIANA DANYCH NA PRZYKŁADZIE SYSTEMU BPSC IMPULS 5

Wprowadzenie

Idea stosowania elektronicznej wymiany danych polega na zastąpieniu, w celu efektywniejszej współpracy między dwiema firmami, przesyłu papierowych dokumentów i ich ręcznego wprowadzania w obu systemach, elektronicznym przesyłem informacji. Klasyczna wymiana dokumentów prowadzi niejednokrotnie do powstawania trudnych do wykrycia błędów, a czas obiegu takich dokumentów może trwać nawet do kilku dni. Również użycie szybkich środków przekazu (np. poczta elektroniczna) nie uwalnia użytkowników w obu firmach od konieczności dostosowania otrzymywanych informacji do formatu wymaganego przez ich system informatyczny. Z pomocą przychodzą coraz szerzej stosowane rozwiązania informatyczne, których kilka możliwych wariantów przedstawiono poniżej.

1. Moduł EDI systemu Impuls 5

Moduł EDI pośredniczy w wymianie danych między systemem Impuls 5 [WWW1] i dowolnymi aplikacjami zewnętrznymi wykorzystującymi, przy wymianie informacji, standard komunikatów EDIFACT przyjęty przez Grupę Roboczą EDI przy Klubie ECR Polska [WWW3]. Firmy przekazują komunikaty za pośrednictwem operatora EDI (na polskim rynku funkcjonuje kilku operatorów, z czego najczęściej spotykani we wdrożeniach to np.: Comarch, Edison, Infinite [WWW2; WWW4; WWW5]); wymieniając dokumenty w pełni automatycznie, a różnice techniczne używanego oprogramowania stają się nieistotne. Poniżej przedstawiono schemat działania elektronicznej wymiany danych modułu EDI z zewnętrzną aplikacją. Dla przejrzystości w niniejszym przykładzie komunikacja dotyczy tylko dwóch firm, jednak w ogólnym przypadku mechanizm ma zastosowanie do dowolnej ilości współpracujących ze sobą jednostek.

* Instytut Matematyki Uniwersytetu Śląskiego, 40-007 Katowice, ul. Bankowa 14, e-mail: rtyrala@ux2.math.us.edu.pl.



Rys. 1. Schemat funkcjonowania Elektronicznej Wymiany Danych

Źródło: [BPSC11].

Elektroniczna wymiana danych między firmą A a firmą B jest realizowana w następujący sposób:

- firma A tworzy dokument w systemie IMPULS 5,
- dokument ten jest przekazywany do modułu EDI odpowiedzialnego za przetworzenie dokumentu do formatu EDI akceptowalnego przez Operatora EDI,
- dokument zostaje przesłany za pomocą Operatora EDI do firmy B,
- moduł konwersji dokonuje przekształcenia formatu otrzymanego dokumentu na format systemu firmy B,
- dokument zostaje zaewidencjonowany w systemie B.

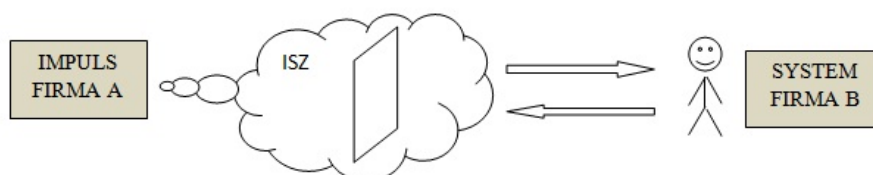
W analogiczny sposób informacja jest przekazywana z firmy B do firmy A:

- firma B tworzy dokument w posiadanym systemie,
- dokument ten jest przekazywany do modułu konwersji odpowiedzialnego za odpowiednie przetworzenie dokumentu do formatu EDI akceptowalnego przez Operatora EDI,
- dokument zostaje przesłany za pomocą Operatora EDI do firmy A,
- moduł konwersji dokonuje przekształcenia formatu otrzymanego dokumentu na format systemu Impuls 5,
- dokument zostaje zaewidencjonowany w systemie Impuls 5.

Komunikaty wymieniane przez systemy są opisane szczegółowymi specyfikacjami. Ze względu na swój charakter, najczęściej we wdrożeniach spotyka się komunikaty związane z procesem zamawiania towarów/usług (komunikat ORDERS [Ord09] – Firma B składa zamówienie w Firmie A używającej systemu Impuls 5) i komunikaty związane z dokumentami sprzedaży (INVOICE / INVOICE CORR [Inv12; InvCorr12] – firma A przekazuje faktury sprzedaży dotyczące zrealizowanych zamówień Firmie B). Wymiana informacji może być znacznie szersza i zawierać komunikaty niosące np. dane o cennikach, dokumentach magazynowych, potwierdzeniach i rozbieżnościach w dostawach. Złożoność komunikacji jest uzależniona w głównej mierze od procesów, które mają odzwierciedlenie w systemach informatycznych stosowanych przez współpracujące firmy. Najczęściej w roli zamawiającego występują duże sieci handlowe (np.: Leroy-Merlin, Selgros, Real, Tesco, Jeronimo Martins (Biedronka), Makro Cash and Carry, Auchan).

2. Moduł Internetowy System Zamówień Impuls 5

Pewną odmianą komunikacji B2B jest Internetowy System Zamówień [BPSC11]. W tym przypadku komunikacja nie przebiega w sposób w pełni automatyczny, jak to przy zastosowaniu modułu EDI, niemniej jednak informacje przekazywane w naturalny sposób (poprzez witrynę internetową) do/z systemu są zgodne z wewnętrznymi strukturami systemu Impuls 5. Dzięki temu w łatwy sposób przedstawiciel zewnętrznej firmy B może zarejestrować bezpośrednio w systemie Impuls 5 (działającym w firmie A) zamówienie, mimo iż sam nie wykorzystuje tego oprogramowania. Tą samą drogą może również uzyskać informację zwrotną o stanie realizacji swoich zamówień lub stanie swoich rozrachunków. Taki sposób komunikacji ma zastosowanie zwłaszcza wtedy, gdy system firmy B nie daje możliwości zintegrowania się lub nie obsługuje niektórych procesów (np. składania zamówień do dostawcy).



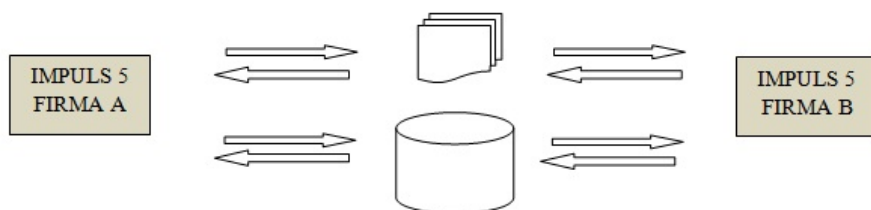
Rys. 2. Internetowy System Zamówień systemu Impuls 5

Oba mechanizmy – EDI i Internetowego Systemu Zamówień – często są stosowane równolegle (do komunikacji z innymi grupami partnerów biznesowych).

3. Integracje bezpośrednie systemu Impuls 5

Częstym przypadkiem integracji z systemami zewnętrznymi jest bezpośrednio komunikacja dwóch systemów. Mowa tu o sytuacji, w której firmy współpracujące chcą wymieniać dane bez pośrednictwa operatora (po wcześniejszym uzgodnieniu wspólnego sposobu komunikacji). Takie integracje są stosowane głównie w przypadku współpracy małej ilości firm lub łączenia systemów informatycznych różnych producentów działających w tej samej firmie, również przy komunikacji między jednostkami organizacyjnymi (oddziałami)

jednej firmy, ewentualnie komunikacji oddziałów z centralą. Nośnikiem informacji może być wówczas plik o wspólnie uzgodnionej strukturze (XML, XLS, płaskie pliki tekstowe, itp.) lub połączenia między bazami danych (z użyciem tabel pośrednich, buforujących przekazywane informacje). Możliwe jest również wykorzystanie mechanizmów Webservice [FZ08].

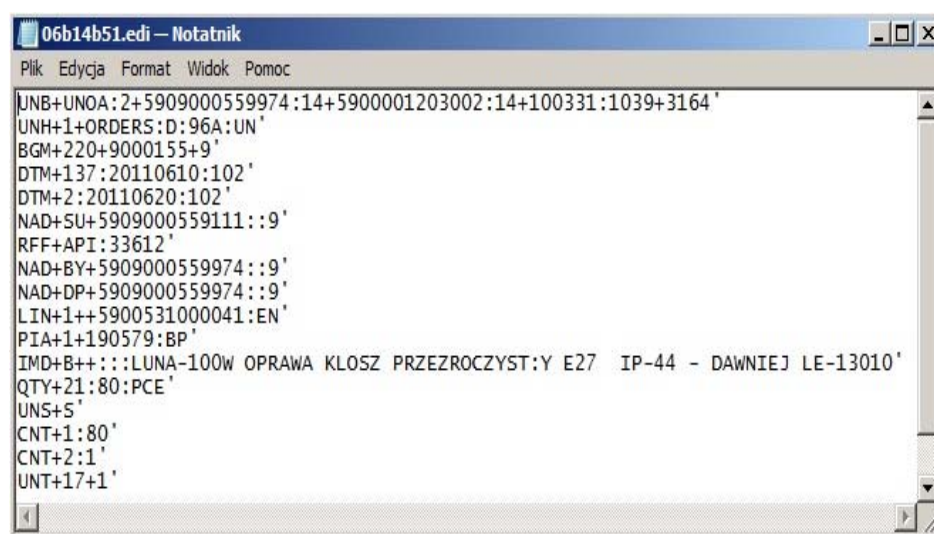


Rys. 3. Bezpośrednia wymiana informacji pomiędzy firmami

4. Przykłady zastosowań

Przypadek 1. Sieć handlowa CASTORAMA zamawia w firmie LMC asortyment oświetleniowy. System magazynowo-dystrybucyjny stosowany przez CASTORAMĘ wspiera tworzenie zamówienia zakupu do dostawcy. Tak utworzone zamówienie za pośrednictwem modułu konwersji zostaje przetworzone do postaci zrozumiałej dla operatora pośredniczącego w przekazywaniu danych między obiema firmami. Bez znaczenia jest tu producent oprogramowania używanego przez CASTORAMĘ – istotna jest jedynie współpraca tego systemu z platformą operatora. System informatyczny CASTORAMY, w celu przekazania do operatora informacji o zamówieniu, może stosować ustalone z operatorem na etapie wdrożenia mechanizmy wymiany danych (pliki o określonej strukturze, np. XML, EDIFACT, bezpośrednie podłączenie operatora do bazy systemu CASTORAMY, Webservice’y). Sposób komunikacji operatora z poszczególnymi systemami jest więc nieistotny dla pozostałych uczestników wymiany informacji i jest uzgodniony indywidualnie z każdym z nich. Tym samym również wewnętrzna reprezentacja informacji o zamówieniu (w obrębie platformy operatora) nie ma znaczenia dla uczestników wymiany informacji – do każdego z nich dane trafiają już w stosownie przetworzonym formacie.

Przekazanie informacji o zamówieniu zakupu przez CASTORAMĘ do operatora kończy pierwszy etap procesu po stronie zamawiającego. Zadaniem operatora jest skuteczne dostarczenie komunikatu zamówienia (z gwarancją niezmięnionej treści) do dostawcy (używającego dowolnego systemu informacyjnego, a w omawianym przypadku do firmy LMC używającej systemu Impuls 5). Ponownie – w zależności od systemu dostawcy – operator dostarcza komunikat zamówienia w formie uzgodnionej na etapie wdrożenia (w przypadku systemu Impuls 5 jest to plik w standardzie EDIFACT – zgodnie ze specyfikacją dostępną w na stronach ECR Polska – Grupa d.s. EDI).



```
06b14b51.edi - Notatnik
Plik Edycja Format Widok Pomoc
JNB+UNOA:2+5909000559974:14+5900001203002:14+100331:1039+3164'
UNH+1+ORDERS:D:96A:UN'
BGM+220+9000155+9'
DTM+137:20110610:102'
DTM+2:20110620:102'
NAD+SU+5909000559111::9'
RFF+API:33612'
NAD+BY+5909000559974::9'
NAD+DP+5909000559974::9'
LIN+1++5900531000041:EN'
PIA+1+190579:BP'
IMD+B+:::LUNA-100W OPRAWA KLOSZ PRZEZROCZYST:Y E27 IP-44 - DAWNIEJ LE-13010'
QTY+21:80:PCE'
UNS+5'
CNT+1:80'
CNT+2:1'
UNT+17+1'
```

Rys. 4. Fragment pliku EDIFACT z komunikatem ORDERS

Plik z zamówieniem jest za pomocą modułu Impuls 5 EDI odczytywany z określonej lokalizacji, w której umieścił go operator, a następnie interpretowany. W efekcie poprawnego przetworzenia w systemie Impuls 5 powstaje Zamówienie Klienta (zgodne ze wszystkimi regułami tworzenia zamówień obowiązującymi w module Dystrybucja systemu Impuls 5 i identyczne z tym, jakie powstałoby w przypadku ręcznego wprowadzenia danych o tym samym znaczeniu, tj. zamówienia tych samych towarów na warunkach oferowanych sieci CASTORAMA).

The screenshot displays the 'ZK Rozrachunki:0,00 pasek wg usera :)-Brutto:0,00-' window in the Impuls 5 system. The window is titled 'Zamówienia Klientów -BPSC SA Rok 2011-' and shows a summary of a client order. The order details include:

- Zamawiający:** 1271 CASTORAMA POLSKA
- Nr wg klienta:** 9000155
- Termin:** 2011-06-20
- Statut zamówienia:** EDI (Zamówienia EDI)
- Jednostka:** DE Wydział ekonomiczno - finansowy 1

The window also shows various fields for payment terms, currency, and other order parameters. The 'Szczegóły' tab is active, and the 'Zamówienie potwierdzone' checkbox is checked.

Rys. 5. Ekran systemu Impuls 5 z zaimportowanym zamówieniem klienta

The screenshot displays the 'ZK Rozrachunki:0,00 pasek wg usera :)-Brutto:0,00-' window in the Impuls 5 system, showing a detailed view of a client order position. The window is titled 'Zamówienia Klientów -BPSC SA Rok 2011-' and shows a summary of a client order. The order details include:

- Zamawiający:** 1271 CASTORAMA POLSKA
- Nr wg klienta:** 9000155
- Termin:** 2011-06-20
- Statut zamówienia:** EDI (Zamówienia EDI)
- Jednostka:** DE Wydział ekonomiczno - finansowy 1

The window also shows various fields for payment terms, currency, and other order parameters. The 'Szczegóły' tab is active, and the 'Zamówienie potwierdzone' checkbox is checked. The 'Szczegóły pozycji zamówienia' section is expanded, showing the following details:

- Indeks:** 103003
- Nazwa indeksu/usługi:** UNA-100W OPRAWA KŁOSZ PRZEZROCYSTY, E27 IP-44
- Ilość sp:** 80 szt
- Ilość do wysłania:** 80,000 szt
- Terminy:** 2011-06-20
- Statuty:** WS WS
- Cena:** 0,00

Rys. 6. Ekran Impuls 5 z pozycjami zaimportowanego zamówienia klienta

W tym momencie kończy się cały proces składania zamówienia, choć istnieją możliwości (a także dodatkowe komunikaty), które pozwalają uczynić ten proces bardziej skomplikowanym (np. poprzez wprowadzenie potwierdzenia przyjęcia zamówienia z informacją zwrotną o stanie dostępności poszczególnych towarów). Dalsza obsługa realizacji zamówienia odbywa się wyłącznie w systemie Impuls 5 (bez udziału systemów zewnętrznych i operatora) za pomocą mechanizmów zgodnych z wdrożoną funkcjonalnością modułów Dystrybucja i/lub Gospodarka Magazynowa systemu Impuls 5. Efektem końcowym jest wystawienie przez firmę LMC faktury sprzedaży dla klienta CASTORAMA za zrealizowane zamówienie.

Data sprzed.	Pełny numer faktury	Rodzaj	Wartość netto	Wartość VAT	Wartość brutto	Termin płat.	Rejestr	Waluta
2011-01-31	1/2221/01	FV	100,00	22,00	122,00	2011-02-14	PK1	
2011-06-22	1/2221/06	FV	10 160,00	711,20	10 871,20	2011-06-24	PK1	
2011-01-31	FVK/1/0001/01/2011	FVK	-299,72	-19,03	-318,75	2011-02-05	PK1	

Rys. 7. Dokument w rejestrze faktur sprzedaży modułu Dystrybucja Impuls 5

The screenshot shows the 'Kartoteka dokumentów sprzedaży' window with a list of invoices and a detailed view of invoice 1/2221/06. The list includes columns for date, invoice number, type, net value, VAT, gross value, due date, and register. The detailed view shows the following information:

- Rodzaj faktury: FV Faktura VAT 0
- Miejsce wyst.: DE Wydział ekon.
- Data wystawienia: 2011-06-22
- Pełny numer faktury: 1/2221/06
- Data sprzedaży: 2011-06-22
- Kontrahenci: Zamawiający: 1271 CASTORAMA POLSKA; Płatnik: 1271 CASTORAMA POLSKA; Nri osoby fizycznej: <brak>
- Sposób płatności: 1 Przelew; Termin płatności: 2011-06-24; dni: 2; od: 2011-06-22
- Nr konta: <brak>
- Odbiorca: 1271 CASTORAMA POLSKA; Adres dostawy: <brak>
- Upusty: upust, za obrót, za termin płat., za wart. trans.
- Zaliczki/Płaty, Informacje związane, Pozostałe, Intrastrat
- Wartość netto: 10 160,00; Wartość VAT: 711,20; Wartość brutto: 10 871,20; Zapłacono: 0,00; 10 871,20

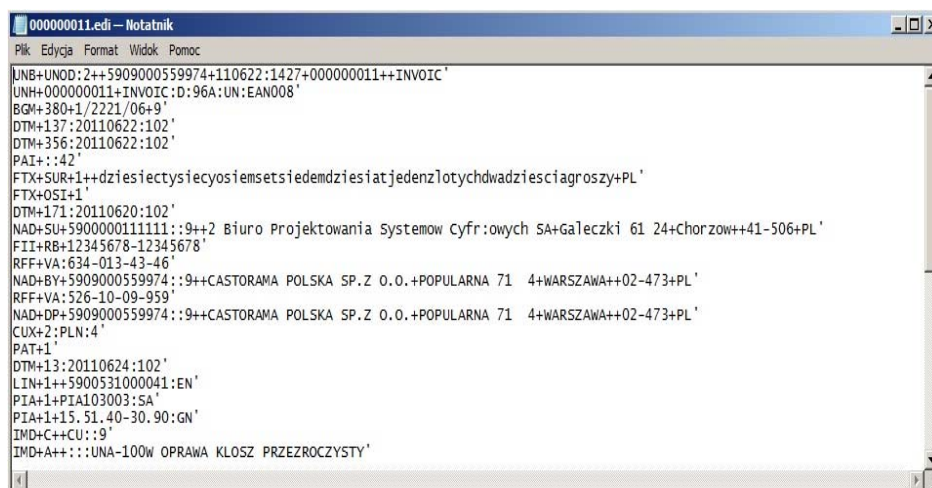
Rys. 8. Dokument sprzedaży w rejestrze faktur sprzedaży – nagłówek

The screenshot shows the 'Kartoteka dokumentów sprzedaży' window with a list of invoices and a detailed view of invoice 1/2221/06. The list includes columns for date, invoice number, type, net value, VAT, gross value, due date, and register. The detailed view shows the following information:

- Rodzaj faktury: FV Faktura VAT 0
- Miejsce wyst.: DE Wydział ekon.
- Data wystawienia: 2011-06-22
- Pełny numer faktury: 1/2221/06
- Data sprzedaży: 2011-06-22
- Szczegóły: 1. Faktura, 2. Pozycje, 3. Dekretacja, 4. Formularz
- Indeks: 103003 UNA-100W OPRAWA KŁOSZ PRZEPROZCZYSTY, E27 IF
- Magazyn: aaaaaaaaa M
- Miejsce skład.: p 1 Podstawowe
- Ilość: 80,000 szt
- Cennik: <brak>
- Cena jedn. netto: 127,0; Rabat: 0,00 % Kwota; Cena sprzed.: 127,00000
- Cena jedn. brutto: 135,9; Rodzaj: <brak>; Cena brutto: 135,89000
- Stawki/Zaliczki: Stawka: 7; Wart. zaliczki w walcie.
- Ogólne: Schemat dekr.: <brak>; Transakcja VAT sprzed.: <brak>
- Opis pozycji
- Wartość netto: 10 160,00; Wartość VAT: 711,20; Wartość brutto: 10 871,20; Zapłacono: 0,00; 10 871,20

Rys. 9. Dokument sprzedaży w rejestrze faktur sprzedaży – pozycje

Dokument sprzedaży powstaje w module Dystrybucja systemu Impuls 5 i rozpoczyna się proces przekazania (analogiczną ścieżką) komunikatu do odbiorcy (CASTORAMA). Moduł EDI konwertuje więc dane wewnętrznej reprezentacji faktury w systemie Impuls do pliku zgodnego ze specyfikacją EDIFACT INVOIC.



```
00000011.edi - Notatnik
Plik Edycja Format Widok Pomoc
UNB+UNOD:2++5909000559974+110622:1427+000000011++INVOIC'
UNH+000000011+INVOIC:D:96A:UN:EAN008'
BGM+380+1/2221/06+9'
DTM+137:20110622:102'
DTM+356:20110622:102'
PAI+:42'
FTX+SUR+1++dziesietysiecycosiemsetsiedemdziesiatjedenzlotychdwadziesciagroszy+PL'
FTX+OSI+1'
DTM+171:20110620:102'
NAD+SU+5900000111111::9++2 Biuro Projektowania Systemow Cyfr:owych SA+Galeczki 61 24+Chorzow++41-506+PL'
FII+RB+12345678-12345678'
RFF+VA:634-013-43-46'
NAD+BY+5909000559974::9++CASTORAMA POLSKA SP.Z O.O.+POPULARNA 71 4+WARSZAWA++02-473+PL'
RFF+VA:526-10-09-959'
NAD+DP+5909000559974::9++CASTORAMA POLSKA SP.Z O.O.+POPULARNA 71 4+WARSZAWA++02-473+PL'
CUX+2:PLN:4'
PAT+1'
DTM+13:20110624:102'
LIN+1+5900531000041:EN'
PIA+1+PIA103003:SA'
PIA+1+15.51.40-30.90:GN'
IMD+C+CU::9'
IMD+A++::UNA-100W OPRAWA KLOSZ PRZEZROCZYSTY'
```

Rys. 10. Fragment pliku EDIFACT z komunikatem INVOIC

Plik zostaje automatycznie umieszczony w określonej lokalizacji (ustalony katalog lokalny lub sieciowy w ramach sieci LAN firmy LMC). Z tego miejsca oprogramowanie operatora automatycznie pobiera plik (przetwarzając go do swej postaci obsługiwanej w ramach platformy operatora) i na podstawie zawartości (informacje z tzw. koperty pliku EDIFACT) przekazuje do odbiorcy. Tam ponownie, za pomocą ustalonych wcześniej procedur konwersji, komunikat zostaje wprowadzony do systemu informatycznego odbiorcy, co skutkuje zarejestrowaniem faktury zakupowej. Ponownie – tak jak w przypadku opisanego wcześniej zamówienia – jest to najprostszy przykład wymiany informacji o dokumencie faktury, który może zostać wzbogacony o obsługę innych komunikatów, np. informacje dotyczące wysyłki i odbioru dostawy towaru w magazynach.

W powyższym opisie można zauważyć niezależność technologii stosowanych w obu współpracujących systemach, a także technologii stosowanej w platformie operatora. Jedynie w punktach styku dochodzi do dwustronnych uzgodnień, w których konieczna jest wspólna decyzja o sposobie przekazywania danych. Wszelkie pozostałe operacje są wykonywane na podstawie oprogramowania i infrastruktury poszczególnych systemów, nie wymagają zatem dodatkowych nakładów prac.

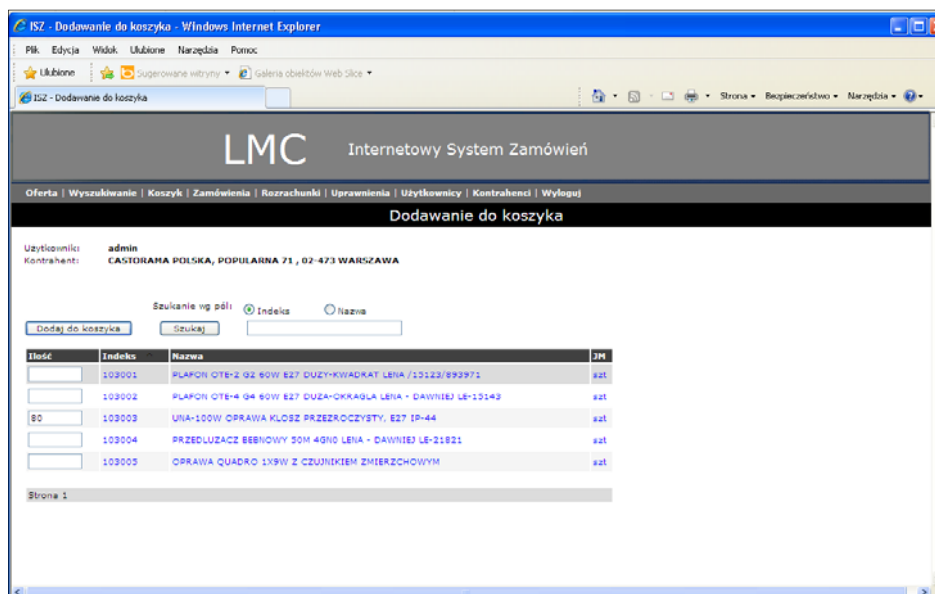
Warto też zaznaczyć, że operator często udostępnia opcję archiwizacji komunikatów, w przekazywaniu których uczestniczy, z możliwością udostępnienia archiwum do celów kontrolnych (w tym również kontroli skarbowej) w czytelnej postaci (komunikaty zinterpretowane).

Należy też zwrócić uwagę na fakt, że uruchomienie takiej modelowej komunikacji między dwiema firmami daje możliwości łatwego poszerzenia współpracy o kolejne firmy handlujące z dostawcą używającym systemu Impuls 5 – wymaga to jedynie ich woli współpracy tą drogą i udrożnienia komunikacji za pośrednictwem operatora po stronie zainteresowanych firm. W części Impulsowej konieczna jest natomiast tylko konfiguracja wersji komunikatów wymienianych z kolejnymi kontrahentami.

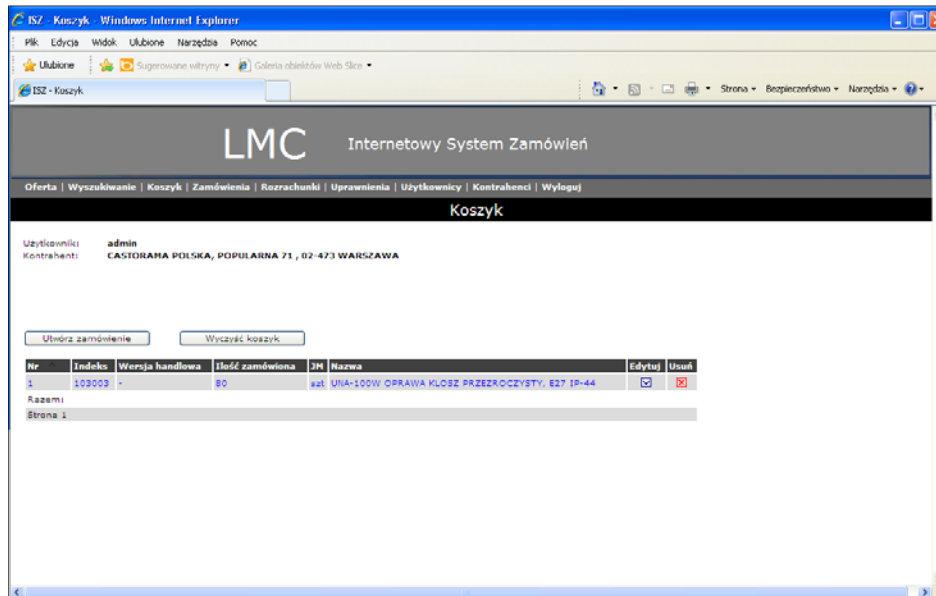
Współpraca elektroniczna realizowana opisaną metodą po etapie pilotażowym staje się praktycznie bezobsługowa – użytkownicy systemów wykonują czynności inicjujące poszczególne procesy (po stronie zamawiającego utworzenie zamówienia, po stronie dostawcy utworzenie faktury), natomiast znika konieczność wykonywania ręcznych czynności związanych z zakończeniem procesu (rejestracja zamówienia u dostawcy i przyjęcie faktury u odbiorcy). W związku z tym zakres obowiązków osób zajmujących się opisanymi procesami nie tylko nie zwiększa się, lecz zmniejsza się o czynności raz już wykonane przez osoby obsługujące system kontrahenta. Automatyzacja procesu zmniejsza ryzyko wystąpienia pomyłek powtórnej rejestracji – w tym zakresie pozostają do wykonania jedynie czynności kontrolne (weryfikacja poprawności otrzymanych automatycznie dokumentów).

Realizacja integracji metodą wymiany komunikatów między partnerami handlowymi za pośrednictwem operatora łączy w sobie cechy integracji procesowej (w pełny proces są zaangażowane obie strony transakcji) z cechami integracji danych (na styku operatora i systemów partnerów handlowych musi dojść do uzgodnienia notacji i przekazania danych składowych komunikatów). Automatyczna obsługa wymiany danych minimalizuje konieczność używania interfejsu do celów samej integracji (poza, co najwyżej, jednorazowymi czynnościami konfiguracyjnymi). Pozostaje w użyciu jedynie interfejs użytkownika obsługujący standardową funkcjonalność obu systemów służącą do wprowadzania odpowiednich dokumentów. Dane są więc prezentowane w sposób typowy dla każdego z komunikujących się systemów.

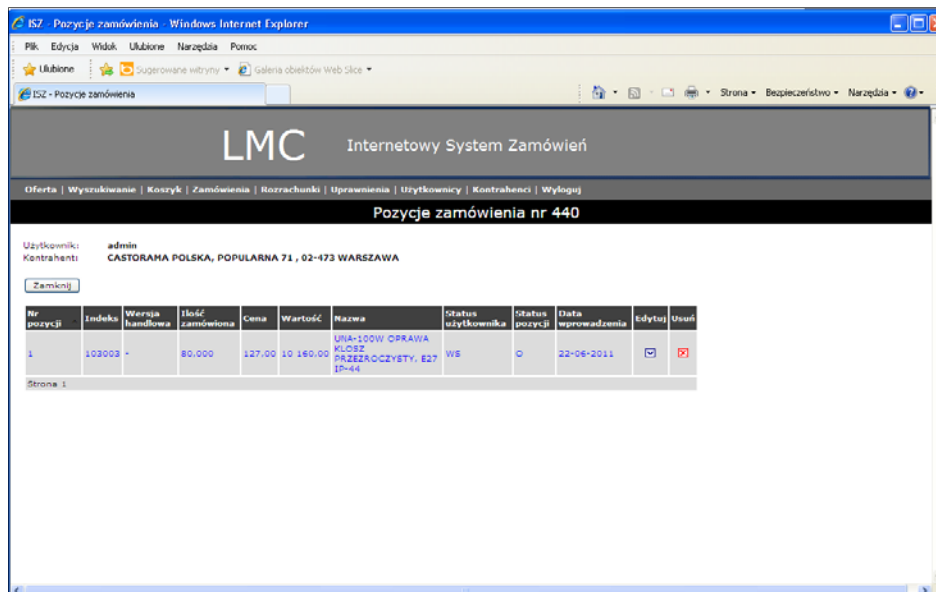
Przypadek 2. Firma LMC udostępnia swoim kontrahentom witrynę internetową umożliwiającą składanie zamówień. Zamawiający loguje się na stworzone na etapie rejestracji konto i poprzez typowe dla takich rozwiązań mechanizmy tworzy zamówienie na towary, które zamierza nabyć. Istotną cechą odróżniającą ten sposób współpracy od opisanego wcześniej przypadku stosowania automatycznej wymiany między systemami jest to, że dane zamówienia nie są wprowadzane w systemie informatycznym zamawiającego. Takie rozwiązanie ma zastosowanie w przypadku braku funkcjonalności tworzenia zamówień zakupu w systemie informatycznym zamawiającego lub innych przyczyn uniemożliwiających integrację między systemami w pełnym zakresie procesu składania i realizacji zamówień. Osoba odpowiedzialna za złożenie zamówienia zakupu z jednej strony wykonuje to niejako wprost w docelowym systemie dostawcy, z drugiej jednak strony jako przedstawiciel zewnętrznej firmy nie korzysta z funkcjonalności systemu Impuls 5 bezpośrednio (co w oczywisty sposób byłoby rozwiązaniem niedopuszczalnym), lecz rejestruje dane za pomocą dedykowanego interfejsu Web'owego.



Rys. 11. Ekran Internetowego Systemu Zamówień – wybór towarów do koszyka



Rys. 12. Ekran Internetowego Systemu Zamówień – zawartość koszyka



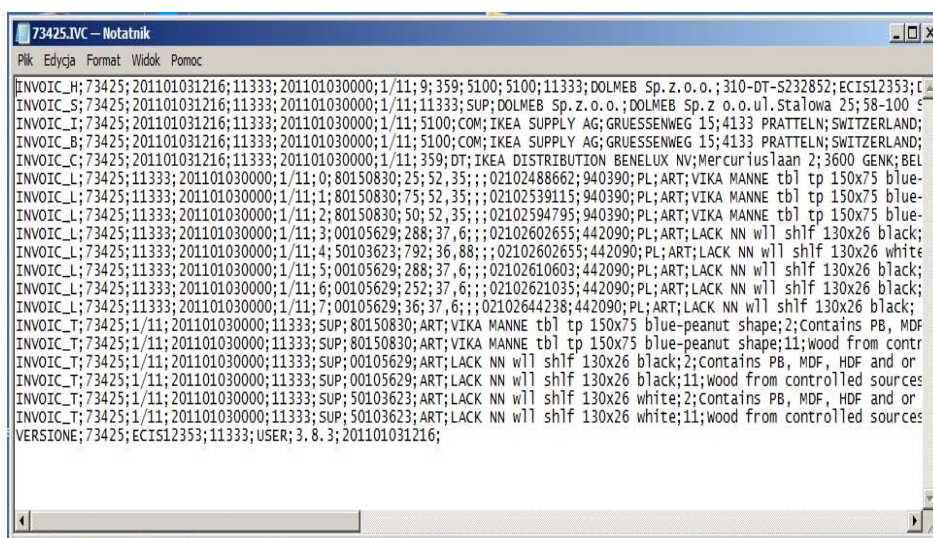
Rys. 13. Ekran Internetowego Systemu Zamówień – przygotowane zamówienie klienta

Przy zachowaniu bezpieczeństwa danych i kontroli uprawnień dla wykonywanych czynności, pracownik dostawcy nie musi ponownie wprowadzać tych samych informacji do systemu. Zamówienie złożone za pomocą Internetowego Systemu Zamówień Klienta systemu Impuls 5 trafia (tak jak w przypadku 1) do modułu Dystrybucja systemu Impuls 5, gdzie jest już przetwarzane przez uprawnionych do tego użytkowników dostawcy. Informacja zwrotna powstająca w wyniku zakończenia procesu realizacji zamówienia może być w najprostszej wersji zaprezentowana przedstawicielowi odbiorcy w postaci stanu rozrachunków. Możliwe jest także zrealizowanie funkcjonalności umożliwiającej pobieranie dokumentów końcowych (faktur sprzedaży) w plikach PDF (lub innych uzgodnionych między partnerami handlowymi). Ponownie widać tu wyraźnie automatyzację przekazywania informacji (po stronie Impuls 5 – ISZ działa na tej samej bazie danych, co sam system Impuls 5). Łącznikiem dla danych po stronie Odbiorcy pozostaje jednak człowiek (wprowadzający i interpretujący dane). Z punktu widzenia technologii, uruchomienie witryny internetowej pozwalającej na przyjmowanie zamówień i przekazywanie informacji zwrotnych (o ich realizacji, stanach magazynowych i rozliczeniach) wymaga konfiguracji dodatkowego serwera (w przypadku ISZ Impuls 5, ze względu na użycie .Net jest nim Microsoft IIS).

Przypadek 3. Niejednokrotnie standard wymiany danych jest narzucany przez jedną ze stron uczestniczących w wymianie handlowej. Jeśli firmy nie przewidują pośrednictwa operatora zarządzającego „transportem” komunikatów między partnerami handlowymi (z przyczyn ograniczenia kosztów wdrożenia przez wyeliminowanie operatora, jako ogniwa w łańcuchu komunikacji), wówczas konieczne staje się ustalenie reguł wymiany danych bezpośrednio. W praktyce często są wykorzystywane mechanizmy istniejące w jednym systemie, podczas gdy drugi system jest dostosowywany tak, aby mógł współpracować według narzuconych zasad. Omawiany przypadek pokazuje, jak firma IKEA (posiadająca we własnym systemie specyficzny mechanizm generacji faktur zakupu na podstawie składanych przez siebie zamówień do dostawców) nakłoniła partnera handlowego (firmę DFL korzystającą z systemu Impuls 5) do wykorzystania informacji przekazywanych w plikach tekstowych podczas tworzenia faktur sprzedaży u dostawcy. Proces jest o tyle nietypowy, że inicjatywa utworzenia dokumentu sprzedaży w systemie dostawcy pochodzi z systemu odbiorcy, a nie – jak w poprzednich dwóch przykładach – jest efektem zakończenia procesu realizacji zamówienia w systemie dostawcy. Treść dokumentu jest

więc niejako narzucana z góry, a dostawca musi dopasować realizację dostawy tak, by uzyskać zgodność dokumentów i tym samym móc zarejestrować fakturę sprzedaży w Impulsie, której w istocie jest wystawcą. Po raz kolejny widać tu cel nadrzędny – eliminację ryzyka popełnienia pomyłki przy wprowadzaniu danych związanych z dokumentem już raz wprowadzonym w jednym z systemów.

W realizacji tej integracji wykorzystano pliki tekstowe generowane przez system używany przez firmę IKEA. Po stronie systemu Impuls 5 nie były wymagane dodatkowe rozwiązania sprzętowe, natomiast konieczna była realizacja mechanizmu importu danych z plików tej postaci i tworzenia z nich stosownych dokumentów sprzedaży w module Dystrybucja systemu Impuls 5.



Rys. 14. Fragment pliku tekstowego z fakturą

Podobne czynności programistyczne zostały wykonane w przypadku innej integracji bezpośredniej – firmy KLG, która otrzymuje od swego odbiorcy – firmy JAGO – pliki XML z zamówieniami. Tu proces wymiany informacji jest zredukowany jedynie do importu zamówień (bez zwrotnej informacji o wystawionych fakturach). Efekt importu pliku jest identyczny jak w dwóch poprzednich przypadkach – powstaje zamówienie typowe dla modułu Dystrybucja systemu Impuls 5.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <Documents>
- <Document-Order>
- <Order-Header>
  <OrderNumber>WZ/005160/2010/002886</OrderNumber>
  <CustomerOrderNumber>Z226005110</CustomerOrderNumber>
  <OrderDate>2010-05-18</OrderDate>
  <ExpectedDeliveryDate>2010-05-19</ExpectedDeliveryDate>
  <DocumentFunctionCode>D</DocumentFunctionCode>
</Order-Header>
- <Order-Parties>
  <Buyer>
  <Seller>
    <ILN>005160</ILN>
  </Seller>
- <DeliveryPoint>
  <ILN>Z2260</ILN>
  <DeliveryPointName>Z2260</DeliveryPointName>
  <DeliveryPointPostalCode>99300</DeliveryPointPostalCode>
  <DeliveryPointCity>Kutno</DeliveryPointCity>
  <DeliveryPointAddress>Barlickiego 14</DeliveryPointAddress>
  <DeliveryPointULCode>12260</DeliveryPointULCode>
</DeliveryPoint>
- <ShipFrom>
  <ILN>5900000933351</ILN>
  <ShipFromName>JAGO</ShipFromName>
  <ShipFromCode>32065</ShipFromCode>
  <ShipFromCity>Krzyszowice</ShipFromCity>
  <ShipFromAddress>Daszyńskiego 10 A</ShipFromAddress>
  <ShipFromULCode>000002</ShipFromULCode>
</ShipFrom>
</Order-Parties>
- <Order-Lines>
```

Rys. 15. Fragment pliku XML z zamówieniem

Integracje bezpośrednie cechuje niewątpliwa zaleta – są tańsze w realizacji, wymagają bowiem jedynie modyfikacji programistycznych wynikających z dwustronnych uzgodnień. Trzeba jednak mieć świadomość faktu, iż nie jest tu w żaden sposób gwarantowana warstwa „transportowa”. Jeśli nośnikiem informacji są pliki, to w gestii użytkowników obu systemów leży konieczność wzajemnego ich przekazania (np. za pomocą poczty elektronicznej, FTP). W rozwiązaniach korzystających z Webservice’ów problem pośrednictwa w przekazywaniu danych rozwiązuje się sam ze względu na samą technologię takich rozwiązań opartą na Internecie. Ponadto w stosowaniu integracji bezpośrednich należy zachować daleko idący umiar i ostrożność, by uniknąć tzw. in-

formatycznego spaghetti, tzn. równoczesnego integrowania wielu systemów z zastosowaniem indywidualnych rozwiązań do każdej z tych integracji. Przy dużej ilości integrowanych systemów w szybkim tempie rośnie bowiem koszt utrzymania takich rozwiązań. Wówczas stanowczo jest polecane wykorzystanie pośrednictwa operatora lub wykorzystania rozwiązań opartych na platformach integracyjnych.

Podsumowanie

Dobór sposobu komunikacji między systemami informatycznymi zależy nie tylko od ich możliwości technicznych, ale również w dużej mierze od złożoności procesów biznesowych zachodzących między kontrahentami wykorzystującymi te systemy. Opisane tu przypadki zarysowują zagadnienie integracji na nieskomplikowanym, jednakże bardzo często występującym modelu biznesowym, w którym jedna strona tworzy zamówienia, a druga je realizuje poprzez sprzedaż, czemu towarzyszy generowanie odpowiednich dokumentów. Zaprezentowane przykłady nie wyczerpują wszystkich możliwości jakie dają współczesne narzędzia informatyczne (w szczególności technologie WebService'ów czy Platform Integracyjnych, takich jak Microsoft BizTalk zostały tu jedynie wspomniane bez szczegółów). Niemniej jednak przedstawione przypadki wdrożeń dają wyobrażenie o kluczowych w każdej integracji „punktach styku” systemów, a także o korzyściach płynących z dobrze zrealizowanej elektronicznej wymiany danych.

Literatura

- [BPSC11] Materiały informacyjne BPSC 2011.
- [FS08] Fryźlewicz Z., Salamon A.: *Podstawy architektury i technologii usług XML sieci Web*. Wydawnictwo Naukowe PWN, Warszawa 2008.
- [Inv12] Specyfikacja Komunikatu Faktura, INVOIC EANCOM97/EDIFACT D.96A, wyd. 8.4, 05.2012.
- [InvCorr12] Specyfikacja Komunikatu Korekta Faktury, CORRECTING INVOIC EANCOM97/EDIFACT D.96A, wyd. 7.0, 03.2012.
- [Ord09] Specyfikacja Komunikatu Zamówienie, ORDERS EANCOM97/EDIFACT D.96A, wyd. 6.2, 08.2009.
- [WWW1] <http://www.bpsc.com.pl/>.
- [WWW2] <http://www.comarch.pl/edi-integracja-i-wsparcie-sprzedazy>
- [WWW3] http://www.ecr.pl/grupy/grupa_edi/.

[WWW4] <http://www.edison.pl/>.

[WWW5] http://www.infinite.pl/Produkty-EDInet_Solutions.

[WWW6] <http://www.microsoft.com/biztalk/>.

ELECTRONIC DATA INTERCHANGE ON THE EXAMPLE OF BPSC INFORMATION SYSTEM IMPULS 5

Summary

The paper covers consideration of Electronic Data Interchange (EDI) fundamentals as well as the analyses of 3 case studies for exemplification of proposed by BPSC software company solutions. At first, author presents EDI module implemented in BPSC Impuls 5 information system. Next part includes analysis and explanation of Internet Order System developed within BPSC Impuls 5 system. In the last part, author considers 3 case studies to illustrate EDI solutions. In the case studies the implementation of EDI was done in Castorama retail network, LMC company and IKEA firm.

Mariusz Żytniewski
Piotr Zadora

MODELOWANIE PROCESÓW BIZNESOWYCH Z UŻYCIEM NOTACJI BPMN

Wprowadzenie

W teorii nauk o zarządzaniu jednym z nurtów zarządzania organizacją jest podejście procesowe, które zakłada horyzontalne oraz kompleksowe spojrzenie na strukturę organizacji. W podejściu tym wyróżnione elementy organizacji są koordynowane przez menedżera danego procesu biznesowego, którego zadaniem jest nadzorowanie i koordynowanie przebiegu danego procesu. W ostatnich latach na rynku pojawiło się wiele rozwiązań informatycznych wspomagających projektowanie i monitorowanie procesów biznesowych, takich dostawców jak: ADONIC, ARIS, Axway, BizAgi, Holocentric Modeler, iGraphix, Intalio, Provision, TIBCO. Oprogramowanie takie wspomaga proces identyfikacji zadań realizowanych przez poszczególne komórki organizacji, pozwala na definiowanie graficznej reprezentacji procesu biznesowego i symulowanie przebiegu realizacji procesu biznesowego. Oprogramowanie to zwykle zapewnia możliwość przygotowania interfejsów użytkownika w architekturze klient-serwer, które pozwalają na kontrolowanie pracy poszczególnych uczestników procesu.

Dla celów definiowania graficznej reprezentacji procesu biznesowego można wyróżnić wiele notacji wspomagających modelowanie procesów biznesowych, takich jak BPMN (Business Process Modeling Notation) oraz BPEL (Business Process Execution Language). Niniejsze opracowanie prezentuje syntetyczne ujęcie wskazanych koncepcji.

W części pierwszej zostaną zaprezentowane podstawy teoretyczne dotyczące procesów biznesowych. Zostaną wskazane różnice podejścia procesowego i funkcjonalnego, definicja procesu, cechy oraz mierniki procesu biznesowego. Następnie zostaną ukazane przykłady modelowania prostych procesów biznesowych na podstawie wybranych elementów notacji BPMN.

1. Podejście procesowe w budowie organizacji

Podejście procesowe w teorii nauk o zarządzaniu jest związane ze zmianą spojrzenia na strukturę organizacyjną z pionowego, liniowego, funkcjonalnego na poziome, horyzontalne, procesowe. Tabela 1 prezentuje różnice między obydwojema podejściami.

Tabela 1

Różnice podejścia procesowego i funkcjonalnego

Podejście funkcjonalne	Podejście procesowe
Działania przedsiębiorstwa nastawione na realizację poszczególnych funkcji, takich jak marketing, finanse	Działania przedsiębiorstwa koncentrują się na zarządzaniu procesami w celu maksymalizacji satysfakcji klienta
Nacisk na dostawę produktów i usług	Nacisk na jakość wykonywanej pracy
Funkcje na ogół nie są dostatecznie koordynowane	Występuje koordynacja między funkcjami
Zarządza się wycinkami procesów	Obowiązuje spojrzenie systemowe, tj. na cały proces i konfigurację procesów organizacji
Dominuje orientacja na wnętrze organizacji	Dominuje orientacja na klienta zarówno wewnętrznego, jak i zewnętrznego
Brak konkurencji między działaniami	Występuje wewnętrzna konkurencja w firmie
Zachowania są uwarunkowane przez opisy stanowisk pracy	Dominuje praca zespołowa
Ścisła, sformalizowana kontrola	Role i odpowiedzialność wynikają ze specyfiki procesów
Powolna adaptacja do otoczenia	Wymagane zmiany w procesie ciągłego doskonalenia organizacji
Ustrukturalizowana kontrola przepływu informacji	Swobodny przepływ informacji
Przedsiębiorstwo nastawione na klientów zewnętrznych. Brak relacji klient-dostawca wewnątrz przedsiębiorstwa	Przedsiębiorstwo wchodzi w relacje zarówno z klientami wewnętrznymi, jak i zewnętrznymi
Zasoby rozdzielane pomiędzy konkurujące ze sobą funkcje i specjalności zawodowe	Zasoby przypisane poszczególnym procesom
Wynagradzanie na podstawie tradycyjnych miar wydajności działania, najczęściej jednak czasu pracy	Kryterium oceny wynagradzania są miary rezultatów działania
Przedsiębiorstwo o rozbudowanej strukturze hierarchicznej z dominacją komunikacji pionowej	Przedsiębiorstwo skoncentrowane wokół procesów z dominującą komunikacją poziomą

Źródło: [Bit09].

Organizacja jest traktowana jako zbiór procesów, które przenikają się wzajemnie, są powtarzalne i jasno zdefiniowane. Podejście takie ma na celu zwiększenie konkurencyjności organizacji w dynamicznie zmieniającym się otoczeniu, dzięki zmniejszeniu kosztów, podniesieniu jakości usług oraz usprawnieniu jej działania poprzez jasne określenie etapów realizacji procesu, możliwość jego monitorowania i eliminację potencjalnych „wąskich gardeł”. Podejście procesowe wspomaga efektywniejsze realizowanie celów przyjmowanych przez organizację.

Analizując rozwój podejścia procesowego w naukach o zarządzaniu wskazuje się ewolucję aktualnego podejścia procesowego. Pierwsze założenia dotyczące naukowej organizacji pracy i podziału procesu produkcyjnego na etapy, można odnaleźć w podejściu F. Taylora, skupiającym się na sferze produkcyjnej, co było związane z optymalizacją kosztów oraz czasu produkcji. Drugą fazą rozwoju podejścia procesowego było BPR (Business Process Reengineering) odnoszące się do radykalnej przebudowy struktury organizacji oraz wykorzystywanych metod zarządzania.

Ostatnia faza rozwoju tego podejścia jest związana z koncepcją BPM (Business Process Management) i ukierunkowana na prowadzenie ciągłych zmian funkcjonowania. Istotą tego podejścia jest całościowe spojrzenie na organizację, które ma przyczynić się do identyfikacji wszystkich możliwych procesów jakie w niej zachodzą. Podejście to zakłada także szerokie wsparcie procesów biznesowych przez narzędzia informatyczne w postaci oprogramowania wspomagającego modelowanie procesów oraz symulację ich realizacji.

2. Definicja, cechy oraz mierniki procesu biznesowego

Zgodnie z normą ISO 9000:2000 proces, to „(...) zbiór działań wzajemnie powiązanych lub wzajemnie oddziałujących, które przekształcają wejścia w wyjścia”, a podejście procesowe, to „(...) systematyczna identyfikacja procesów stosowanych w organizacji i zarządzanie nimi, a szczególnie oddziaływaniami między takimi procesami”. W związku z tym wyróżnia się pojęcie procesu biznesowego rozumianego jako zbiór powiązanych ze sobą ustrukturalizowanych działań lub czynności, które służą określonemu celowi, np. dostarczeniu konkretnej usługi lub produktu dla danego odbiorcy.

Proces biznesowy cechuje się:

- jasnym określeniem mierzalnego celu, dla którego jest definiowany,
- zdefiniowanym wejściem oraz wyjściem,
- powtarzalnością,
- mierzalnością wyników,

– możliwym wyjściem poza granice poszczególnych jednostek organizacyjnych.

Wyróżnia się trzy główne rodzaje procesów biznesowych:

1. Procesy zarządcze (*management processes*) – procesy nadzorujące działanie organizacji, przykłady to: „nadzór korporacyjny” oraz „zarządzanie strategiczne”.
2. Procesy operacyjne (*operational processes*) – procesy kształtujące główną działalność gospodarczą organizacji i uczestniczące w tworzeniu łańcucha wartości, przykłady to: „zakupy”, „produkcja”, „marketing”, „sprzedaż”.
3. Procesy pomocnicze (*supporting processes*) – wspomagające procesy operacyjne, przykłady to: „księgowość”, „kadry i płace”, „pomoc techniczna”.

Z punktu widzenia działania organizacji konieczne jest zdefiniowanie właściciela procesu, którego zadaniem będzie nadzorowanie przebiegu i realizacji procesu. Z punktu widzenia organizacji osoba taka powinna rozumieć istotę całego procesu oraz posiadać kompetencje do określania zadań różnych komórek organizacji, które w tym procesie biorą udział. Jest to istotne dla zapewnienia sprawnego działania procesu oraz jego efektywności. Do mierników procesów biznesowych można zaliczyć:

- czas trwania,
- elastyczność,
- jakość,
- koszt,
- terminowość,
- znaczenie dla organizacji,
- znaczenie dla klienta [Bit09].

Prezentowane mierniki w różnym stopniu mogą określać efektywność realizacji danego procesu biznesowego oraz są konieczne do jego oceny i ewaluacji.

3. Modelowanie procesów biznesowych

Modelowanie procesów biznesowych jest związane z potrzebą graficznej reprezentacji procesów biznesowych dla celów ich optymalizacji i archiwizacji. W związku z tym, jak wskazano to wcześniej, konieczna jest znajomość struktury całej organizacji, określenie celu danego procesu, jego zasobów oraz przyjęcie określonej notacji, w której dany proces biznesowy zostanie zamodelowany. Jak wskazuje Bitkowska [Bit09] narzędzia wspomagające proces modelowania procesów biznesowych można podzielić na:

- narzędzia do tworzenia diagramów, służące głównie do wizualizacji i mapowania procesów, do których należą: Microsoft Visio, Flowchart (Micrografix),
- narzędzia CASE do modelowania procesów, szczególnie wtedy, gdy mają być one integrowane z rozwiązaniami informatycznymi, np. Designer/2000 (Oracle), Select Enterprise (Select Software),
- narzędzia projektowania i doskonalenia procesów, pozwalające na zaawansowane analizy i symulacje, takie jak: ARIS Toolset, Igrafix, Adonis oraz narzędzia modelowania procesów w ramach systemu ERP.

Z punktu widzenia modelowania procesu biznesowego i jego późniejszych zmian konieczne jest zastosowanie odpowiedniej notacji, w której proces biznesowy zostanie zaprezentowany. Zastosowanie języka UML (*Unified Modeling Language*) wspomagającego obiektowo zorientowane modelowanie aplikacji z punktu widzenia użytkownika biznesowego jest problematyczne, gdyż wymaga od niego wiedzy informatycznej dotyczącej specyfiki podejścia obiektowego.

Z tego względu w celu reprezentowania procesów biznesowych zaproponowano wiele formalizmów, wśród których znaczącą rolę odgrywają BPEL oraz BPMN.

4. Formalizm BPEL

Język BPEL (Business Process Execution Language) jest standardową notacją opisującą przepływ pracy zgodnie z koncepcjami SOA (Service Oriented Architecture). BPEL używa XML-a do opisu procesów i zależy od usług webowych WSDL i SOAP. Projektowanie procesu rozpoczyna się od szkicowania planowanych przepływów pracy, głównie w formie nietechnicznej za pomocą łańcuchów procesów sterowanych zdarzeniami. Kolejnym krokiem jest użycie jednego z narzędzi edycyjnych do utworzenia środowiska BPEL, tj. do sformułowania planu w przyjętej notacji. Przykładowo środowisko Eclipse oferuje programistom BPEL palety akcji, które pozwalają budować procesy za pomocą kilku kliknięć myszką. W języku BPEL procesy są modelowane jako sekwencje dyskretnych kroków zwanych „aktywnościami”. Aktywności te są zwykle dostarczane przez usługi sieciowe (Web Services) do bezpośredniego wywoływania jako elementy *invoke*, mogą też być dostępne w postaci zmiennych. Programiści odwzorowują poszczególne aktywności w postaci węzłów grafów. Każdy węzeł zawiera dokładnie jeden element kontrolny, np. warunki wyzwalające inne aktywności albo przepływ wykonujący równolegle inne aktywności. Możliwe jest użycie sekwencji oraz iteracji.

Definicja procesu w notacji BPEL przyjmuje postać dokumentu WSDL (Web Services Description Language – odmiana XML'a dla usług sieciowych) zawierającego struktury, metainformacje i opisy interfejsów jako tzw. elementy PartnerLinks. Elementy te tworzą dodatkową warstwę abstrakcji między wywoływanymi aktywnościami procesu i konkretnymi usługami sieciowymi. BPEL używa domyślnego schematu XML do przekazywania wartości, w efekcie zmienne mogą mieć postać struktur złożonych.

Procesy biznesowe zwykle trwają wiele dni lub tygodni, w zależności od przeznaczenia. Klasyczne sposoby reagowania na awarie, takie jak cofanie transakcji SQL, nie gwarantują w tych warunkach zachowania integralności danych. Przykładowo, trudno wymagać zapamiętywania wszystkich elementarnych czynności wykonywanych w całym okresie. Zamiast tego modelowany proces wykorzystuje kompensację, np. wysyłanie dokumentów korygujących. Przy długotrwałych procesach występuje jeszcze jeden problem: typowy czas zapytania i oczekiwania na odpowiedź HTTP nie wystarczą, jeżeli przygotowywanie odpowiedzi trwa zbyt długo. Zapytanie przesłane do usługi sieciowej synchronicznej wygasłoby zanim usługa ta mogłaby udzielić odpowiedzi. W związku z tym w notacji BPEL istnieją techniki komunikacji asynchronicznej dla realizacji usług sieciowych między klientem a serwerem.

Przykład zapisu w notacji BPEL

```
<process name="purchaseOrderProcess"
  targetNamespace="http://example.com/ws-bp/purchase"
  xmlns="http://docs.oasis-open.org/wsbpel/2.0/process/executable"
  xmlns:lns="http://manufacturing.org/wsd1/purchase">
  <partnerLinks>
    <partnerLink name="purchasing"
      partnerLinkType="lns:purchasingLT" myRole="purchaseService" />
    <partnerLink name="invoicing" partnerLinkType="lns:invoicingLT"
      myRole="invoiceRequester" partnerRole="invoiceService" />
  </partnerLinks>
  <variables>
    <variable name="PO" messageType="lns:POMessage" />
    <variable name="Invoice" messageType="lns:InvMessage" />
  </variables>
  <sequence>
    <receive partnerLink="purchasing" portType="lns:purchaseOrderPT"
```



```
        operation="sendPurchaseOrder" variable="PO"
        createInstance="yes">
        <documentation>Receive Purchase Order</documentation>
</receive>
<flow>
    <documentation>
        A parallel flow to handle shipping, invoicing and scheduling
    </documentation>
    <links>
        <link name="ship-to-invoice" />
        <link name="ship-to-scheduling" />
    </links>
    <sequence>
        <invoke partnerLink="shipping" portType="lns:shippingPT"
            operation="requestShipping"
            inputVariable="shippingRequest"
            outputVariable="shippingInfo">
            <documentation>Decide On Shipper</documentation>
        </invoke>
        <receive partnerLink="shipping"
            portType="lns:shippingCallbackPT"
            operation="sendSchedule" variable="shippingSchedule">
            <documentation>Arrange Logistics</documentation>
        </receive>
    </sequence>
</flow>
<reply partnerLink="purchasing" portType="lns:purchaseOrderPT"
    operation="sendPurchaseOrder" variable="Invoice">
    <documentation>Invoice Processing</documentation>
</reply>
</sequence>
</process>
```

5. Formalizm BPMN

Notacja BPMN posiada skończony i jednoznacznie zdefiniowany zestaw symboli wspomagających modelowanie procesów biznesowych.

Elementy graficzne notacji BPMN można podzielić na cztery podstawowe grupy:

1. Obiekty związane z definiowaniem przepływu procesu, do którego należą zdarzenia, czynności oraz obiekty decyzyjne.
2. Elementy pozwalające na łączenie elementów modelu w postaci przepływów sterowania, komunikatów, asocjacji.
3. Obiekty grupujące inne elementy procesu w postaci zbiorników i torów.
4. Obiekty rozszerzające.

Przykłady graficznych reprezentacji wybranych elementów należących do poszczególnych grup prezentuje tab. 2.

Tabela 2

Przykłady notacji BPMN










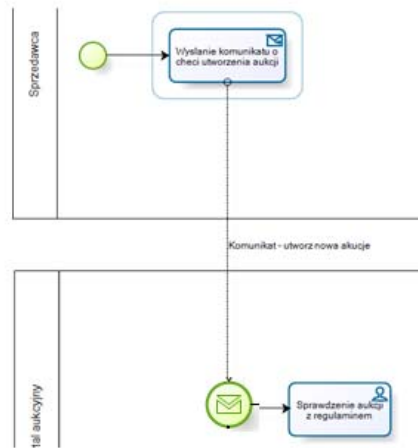
Zbiornik/Uczestnik	
Tor	
Przepływ czynności	
Przepływ komunikatu	
Zadanie	
Proste zdarzenie inicjujące	
Proste zdarzenie końcowe	
Bramka wykluczająca XOR	
Bramka równoległa AND	
Adnotacje	

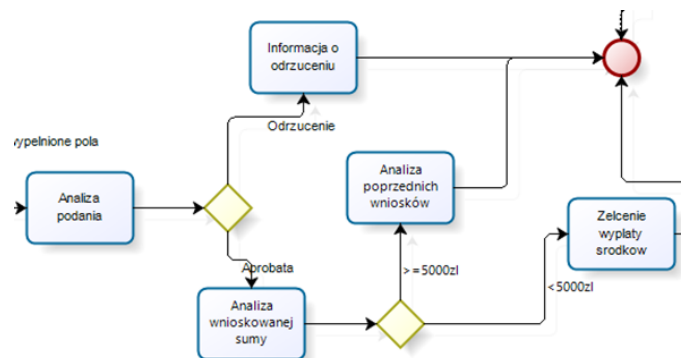
Diagram definiowany w postaci notacji BPMN powinien składać się z jednego (w przypadku jeżeli zdarzenie inicjujące jest anonimowe) lub wielu zdarzeń inicjujących, jednego lub wielu zdarzeń końcowych oraz zadań reprezentujących czynności, które są realizowane w trakcie trwania procesu. Zbiornik reprezentuje uczestnika danego procesu. Uczestnik najczęściej reprezentuje daną organizację, którą modelujemy lub inne podmioty zewnętrzne biorące udział w danym procesie. Jeżeli w ramach danego procesu istnieje tylko jeden uczestnik i nie jest on podzielony na tory symbolizujące jego funkcje, wówczas nie jest

wymagane graficzne nanoszenie zbiornika na diagram. Komunikacja między uczestnikami procesu następuje poprzez wymianę komunikatów (rys. 1). W przypadku gdy nie jest konieczne modelowanie procesu biznesowego w ramach danego uczestnika, komunikacja między nimi może mieć charakter anonimowy.



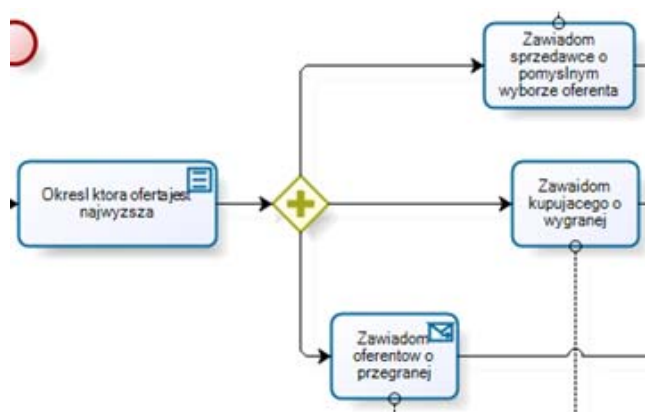
Rys. 1. Przykład komunikacji między użytkownikami procesu biznesowego

Kolejnym elementem, który definiuje się w ramach notacji BPMN są bramki decyzyjne, pozwalające na badanie warunków wpływających na przepływ tokenu w ramach procesu. Pierwszą z bramek jest bramka XOR ukazana na rys. 2. W przypadku tego typu bramki tylko jedno z przejść tokenu jest prawidłowe.



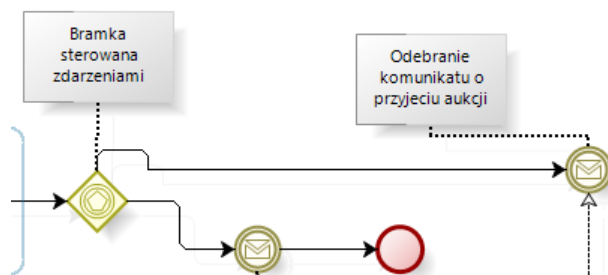
Rys. 2. Przykład zastosowania elementów notacji BPMN

W ukazanym przykładzie zdarzenie „Analiza podania” może doprowadzić do jego odrzucania lub zaakceptowania. Innym rodzajem bramki jest bramka typu AND. Jej zadaniem jest tworzenie zestawu subtokenów poruszających się równolegle lub łączenie ich w jeden token. Przykład zastosowania bramki AND przedstawia rys. 3.



Rys. 3. Przykład zastosowania bramki decyzyjnej AND

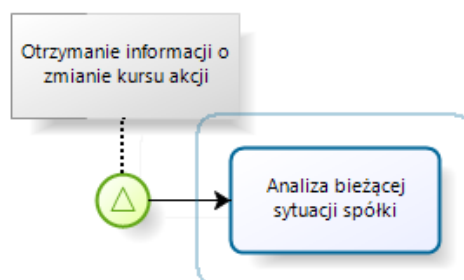
Jak wskazano na rys. 2 bramka typu XOR sterowana danymi pozwala określić zestaw warunków wykluczających, które określają przebieg procesu. Oprócz sterowania przebiegiem procesu za pomocą danych, bramka ta może być sterowana za pomocą zdarzeń, co zaprezentowano na rys. 4.



Rys. 4. Przykład zastosowania bramki decyzyjnej XOR sterowanej zdarzeniami

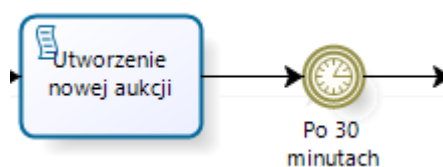
Stosując tę bramkę, określa się, iż dalszy przebieg procesu będzie zależny od wystąpienia danego zdarzenia, które wzbudza przepływ tokenu. Innymi stosowanymi bramkami są bramki typu OR, gdzie definiowane warunki mogą się pokrywać oraz bramki złożone.

Ukazane zdarzenia początkowe, końcowe oraz pośrednie także mogą być różnicowane. Przedstawione zdarzenia emisji oraz odbioru komunikatu pozwalają na komunikację między poszczególnymi uczestnikami procesu. W przypadku kiedy nie można zidentyfikować odbiorcy bądź nadawcy, można zastosować zdarzenia typu sygnał. Rysunek 5 prezentuje przykład zastosowania takiego zdarzenia.



Rys. 5. Przykład zastosowania zdarzenia typu sygnał

Oprócz tego, w diagramie mogą być stosowane np. zdarzenia czasowe, które w przypadku zdarzenia początkowego lub pośredniego oznaczają określony punkt czasu, kiedy dany proces zostanie wzbudzony. Rysunek 6 prezentuje przykład zastosowania takiego zdarzenia.



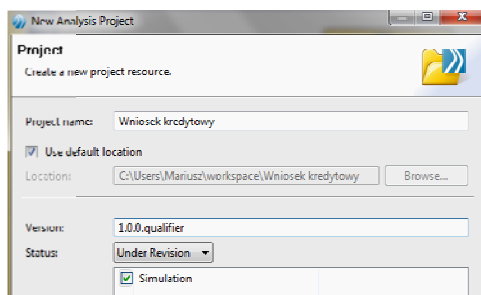
Rys. 6. Przykład zastosowania zdarzenia czasowego

W przypadku stosowania zdarzeń określających kryterium czasu, należy pamiętać o konieczności niepodawania dokładnych dat ze względu na potrzebę powtarzalności procesu. Więcej na temat notacji BPMN można przeczytać w pracach Piotrowskiego oraz Laska i Otmianowskiego [Pi07; LaOt07].

6. Przykład prostego procesu biznesowego

TIBCO Business Studio [WWW2] pozwala na generowanie procesów biznesowych oraz symulowanie ich przebiegu w czasie. Przykładowo dla danego procesu biznesowego możliwe jest określenie uczestników, kosztu ich pracy oraz czasu realizacji zadań. Na podstawie poznanych elementów diagramu BPMN, można spróbować stworzyć prosty model biznesowy.

W pierwszej kolejności ze strony http://developer.tibco.com/business_studio/ należy pobrać oprogramowanie TIBCO. Po jego instalacji uruchomić edytor procesów biznesowych i stworzyć nowy projekt (File → New → Quality Process Project) o nazwie „Wniosek kredytowy” (rys. 7). Następnie wybrać opcję File → New → Analysis Project.



Rys. 7. Definiowanie nazwy projektu

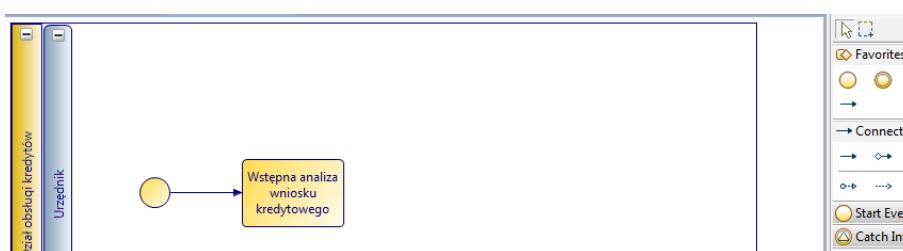
Kolejne działania polegają na tworzeniu procesu biznesowego (w oknie Asset Type Selection należy wybrać Business Processes). Po zatwierdzeniu nazw elementów pakietu na ekranie pojawi się okno procesu. Po lewej stronie, jeżeli nie zostało ustawione to w czasie tworzenia pakietu, można dokonać zmiany nazwy procesu za pomocą opcji Rename. Rysunek 8 przedstawia główne elementy interfejsu użytkownika programu.



Rys. 8. Elementy interfejsu użytkownika

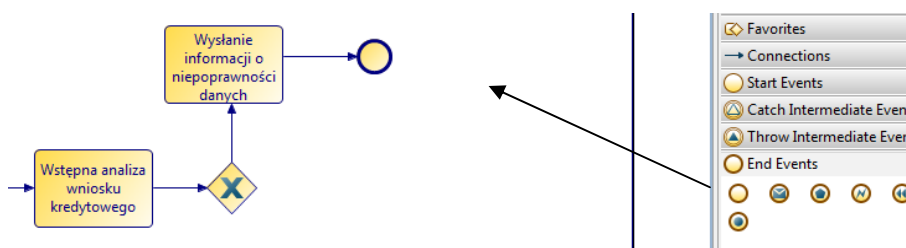
Na rysunku za pomocą jedynki zaznaczono okno przedstawiające aktualne pakiety oraz definicję tworzonego procesu. Liczba 2 ukazuje miejsce, w którym definiuje się proces biznesowy. Liczba 3 oznacza miejsce składowania palety z obiektami możliwymi do umieszczenia w projekcie.

Należy zmienić nazwę „Poll” na „Dział obsługi kredytów” oraz nazwę „Line” na „Urządник”. W pierwszym należy dodać anonimowe zdarzenie inicjujące. Następnie dodać do diagramu zdarzenie (Task) dotyczące analizy wniosku. Zdarzenie to nazywamy „Wstępna analiza wniosku kredytowego” oraz łączymy je za pomocą narzędzia „Sequence Flow” (rys. 9).



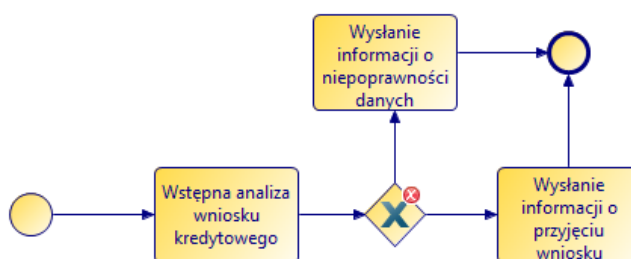
Rys. 9. Połączenie elementów diagramu

Jeżeli dane są niepoprawne, powinniśmy poinformować wnioskodawcę o niepoprawności wniosku i przerwać proces. Aby zamodelować takie działanie, należy dodać nowy obiekt w postaci bramki decyzyjnej (Gateway). W naszym przypadku będzie to bramka wykluczająca typu XOR wskazująca na nowe zdarzenie, którym będzie anonimowy obiekt kończący proces (End Event).



Rys. 10. Dodanie zdarzenia końcowego

W przypadku gdy dane są poprawne, wysyłamy informację o przyjęciu wniosku kredytowego i kończymy proces (rys. 11).



Rys. 11. Definiowanie bramki logicznej

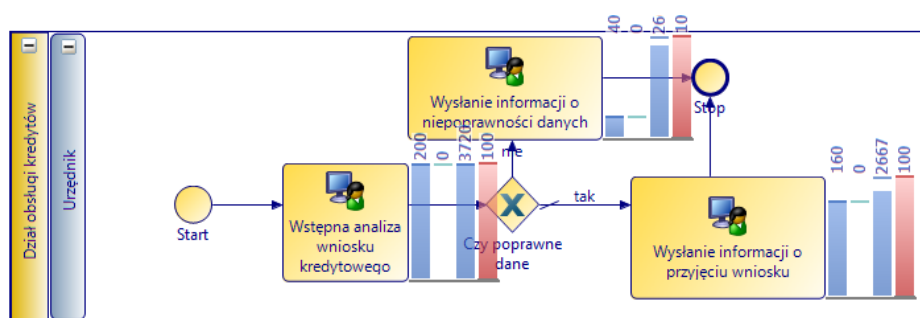
Po dodaniu kolejnego zdarzenia na bramce decyzyjnej pojawi się czerwony krzyżyk. Oznacza to brak określenia, który przepływ jest domyślny. Dodatkowo także powinniśmy dodać nazwę bramki. Klikamy na bramkę decyzyjną i dodajemy nazwę „Czy poprawne dane?”. Następnie klikamy na linię łączącą bramkę decyzyjną oraz wysłanie informacji o przyjęciu wniosku. Przepływ ten nazwiemy *tak* oraz ustawiamy go jako domyślny. Drugi z przepływów nazywamy *nie* i ustawiamy jako warunkowy (Conditional) w oknie jego właściwości.

Teraz należy przeprowadzić symulację. W tym celu należy dodać uczestników procesu. W oknie ukazującym elementy pakietu, należy wybrać węzeł Participants → New → Participant. Mimo iż nie jest wymagane, aby nazwy uczestników były identyczne jak nazwy torów i zbiorników, nazwijmy go „Urzędnik”. Następnie należy wybrać zakładkę Destinations i zaznaczyć Simulation.

W zakładce Problems pojawią się opisy problemów na jakie natrafił edytor (konieczny jest wybór rodzaju: zadania użytkownika lub zadania realizowane przez system). Należy wybrać poszczególne zadania PPM, a następnie opcję Activity type → User Task. Po tej zmianie w zakładce General każdego z zadań należy wybrać urzędnika w polu Participants, a następnie zaznaczyć zakładkę Simulation i przypisać pracownikowi koszt 7 zł za godzinę pracy.

Ostatnim krokiem jest sparametryzowanie zadań urzędnika. Dla wstępnej analizy należy wybrać stały czas 5 min, dla pozostałych zadań będzie obowiązywał przedział 2-5 min. Następnie należy zdefiniować warunek początkowy. Zakłada się, że analizie będzie podlegać 200 wniosków. Ich występowanie będzie generowane rozkładem normalnym średnio co 3 min z odchyleniem standardowym 2 min.

Ostatnim etapem będzie określenie parametru wskazującego, czy dane w dokumencie są poprawne. Parametr ten jest stosowany przez bramkę XOR naszego diagramu. W poddrzewie procesu znajduje się węzeł parametrów (Parameters). Należy kliknąć na węzeł oznaczający proces i wybrać opcję przygotowania symulacji (Prepare Simulation). Następnie należy kliknąć na bramkę decyzyjną i w zakładce Rule Parameter zaznaczyć odpowiednie wagi. Zakłada się, iż 20% dokumentów jest niepoprawnych. W celu rozpoczęcia symulacji, należy wybrać prawym przyciskiem myszy węzeł wskazujący na proces i opcję Run AS → Run Simulation.



Rys. 12. Wynik symulacji

Pierwszy ze słupków diagramu (rys. 12) pokazuje ile dokumentów zostało przetworzonych w danym zadaniu. Drugi słupek wskazuje ile dokumentów jest w kolejce. Trzeci słupek wskazuje na sumę opóźnień analizy dokumentów wynikającą z obciążenia pracownika. Nowe dokumenty trafiają do kolejki, ale pracownik jest zajęty przetwarzaniem poprzednich i dlatego pojawiają się opóźnienia. Ostatni słupek to procentowe obciążenie pracownika w ramach tego procesu. Jak widać, założenie iż przydziela się tylko jednego pracownika powoduje ogromne opóźnienia w realizacji tego procesu.

Kolejne działania dotyczące rozwoju procesu biznesowego mogą polegać na dodaniu kolejnego toru wskazującego na działania kierownika działu, dokonującego kolejnej analizy wniosku w przypadku, kiedy wskazane zabezpieczenia kredytu są niewystarczające (rys. 13).

Literatura

- [Bit09] Bitkowska A.: *Zarządzanie procesami biznesowymi w przedsiębiorstwie*. VIZJA PRESS&IT, Warszawa 2009.
- [LaOt07] Lasek M., Otmianowski B.: *BPMN-standard opisywania procesów biznesowych Budowa modeli BPMN*. WIT, Warszawa 2007.
- [Pi07] Piotrowski M.: *Notacja modelowania procesów biznesowych*. BTC, Legionowo 2007.
- [WWW1] BPEL for Programmers and Architects – <http://www.bptrends.com/publicationfiles/BPEL4ProgArchies.pdf>.
- [WWW2] <http://www.tibco.com/products/bpm/process-modeling/business-studio/default.jsp> [dostęp: 20.04.2012].

BUSINESS PROCESS MODELING USING BPMN NOTATION

Summary

Business process modeling in software engineering is the activity of representing processes of an enterprise, so that the current process may be analyzed and improved in the future. One of the most significant formal notations used commonly to create descriptions of business process models is BPMN which stands for Business Process Modeling Notation. This notation provides set of formalisms that are readily understandable by all business stakeholders yet able to represent complex process semantics. These business stakeholders include the business analysts who create and refine the processes, the technical developers responsible for implementing the processes, and the business managers who monitor and manage the processes. The BPMN specification provides a mapping between the graphics of the notation to the underlying constructs of execution languages, particularly (BPEL) Business Process Execution Language.