

Modelowanie współpracy klienta i dostawcy w przedsięwzięciach pozyskiwania oprogramowania.

Krzysztof Wyrzykowski, Janusz Górski
Katedra Inżynierii Oprogramowania
Politechnika Gdańska
jango@eti.pg.gda.pl, wyrzykowskik@poczta.onet.pl

Streszczenie. Efektywna współpraca dostawcy i klienta stanowi jeden z istotnych czynników sukcesu przedsięwzięcia pozyskania systemu informatycznego. Tworzy to silną motywację do poszukiwania efektywnych sposobów wspomagania takiej współpracy. Jednym z możliwych podejść do rozwiązania tego problemu jest modelowanie. Referat przedstawia koncepcję modelowania współpracy (interakcji) klienta i dostawcy oprogramowania. Uzasadniono w nim potrzebę metodycznego podejścia do problemu współpracy oraz wskazano cele jakie powinien realizować Model Współpracy Klient-Dostawca Oprogramowania (w skrócie MWKDO). Zaproponowano koncepcję modelowania wykorzystującą standard UMA, który krótko scharakteryzowano. Przedstawiono wybrane wyniki badań nad zawartością MWKDO jako rozszerzenie nowej generacji metodyki IBM RUP v.7. Porównano obszary wiedzy PMBok®PMI z MWKDO oraz przedstawiono wstępne wyniki związane z zastosowaniem MWKDO w studium przypadku dotyczącym pozyskiwania oprogramowania w układzie szpital (klient) – integrator oprogramowania medycznego (dostawca).

1. Wprowadzenie

Współpraca klienta i dostawcy oprogramowania w ramach przedsięwzięć informatycznych obarczona jest wieloma ryzykami. Jak wskazują źródła, np. [MCC1993], ryzyka obszaru współpracy dotyczą:

- a) procesu organizacji,
- b) użytkowników końcowych,
- c) klienta,
- d) wymagań,
- e) produktu,
- f) personelu i
- g) procesu wytwórczego.

Brak zapobiegania tym ryzykom spowodować może istotne zmniejszenie szansy osiągnięcia sukcesu przedsięwzięcia informatycznego. W konsekwencji (a) zakres przedsięwzięcia nie zostanie zrealizowany, (b) przedsięwzięcie nie zakończy się w zakontraktowanym terminie oraz (c) zostanie przekroczony jego budżet.

W artykule proponujemy podejście mające na celu obniżenie ryzyka dotyczącego współpracy klienta i dostawcy oprogramowania, polegające na wyodrębnieniu obszaru współpracy i opracowanie modelu referencyjnego opisującego interakcję obu stron. W wyniku powstaje *model współpracy klient-dostawca oprogramowania* (w skrócie MWKDO), który może zostać włączony w zakres kontraktu i wspierać obie strony podczas przedsięwzięcia pozyskania oprogramowania. W szczególności MWKDO wspomaga uzyskanie odpowiedzi na następujące pytania:

- a) Jak będzie przebiegała współpraca klienta z dostawcą oprogramowania?
- b) Jakie są niezbędne zasoby i procesy (bo obu stronach)?
- c) Co robić w sytuacjach konfliktowych?
- d) Jakie są czynniki ryzyka w przypadku odstępstw od rekomendowanego modelu współpracy?

2. Cele MWKDO oraz charakterystyka modelu

Istotą MWKDO jest wsparcie oraz przewodnictwo dotyczące współpracy klienta i dostawcy oprogramowania. Cele, które realizuje kompletny model są następujące:

- a) sformalizowanie oczekiwań klienta i dostawcy co do procesów związanych z pozyskaniem oprogramowania,
- b) analiza potencjału klienta do współpracy w trakcie pozyskiwania oprogramowania (mocne i słabe strony),
- c) przygotowanie klienta i dostawcy do efektywnej współpracy,
- d) poprawa zrozumienia przez obie strony ról, które wypełnią w procesie pozyskiwania oprogramowania,
- e) zrozumiały i jednoznaczny język komunikacji obszaru współpracy dla wszystkich jego udziałowców (różne grupy zawodowe),
- f) zharmonizowanie działań w zakresie współpracy z innymi procesami biznesowymi wynikającymi z działalności statutowej odbiorcy i klienta,
- g) uszczegółowienie najważniejszych zadań i czynności,
- h) wczesna identyfikacja ryzyk obszaru współpracy umożliwiająca skuteczne zarządzanie tymi ryzykami,
- i) dostarczenie mechanizmów, które umożliwiają analizę i zarządzanie ryzykiem pojawiającym się podczas współpracy (odstępstwa od MWKDO).

Proponowany MWKDO koncentruje się na cechach ogólnych. W konsekwencji model umożliwia wsparcie wielu 'podobnych' przedsięwzięć pozyskania oprogramowania, tzn. przedsięwzięć dotyczących oprogramowania tej samej klasy aplikacyjnej (na przykład, systemów w zastosowaniach medycznych). Takie podejście jest zgodne z zaleceniami w [SEI2004]. Tabela 1 przedstawia cechy MWKDO oraz korzyści biznesowe z ich implementacji.

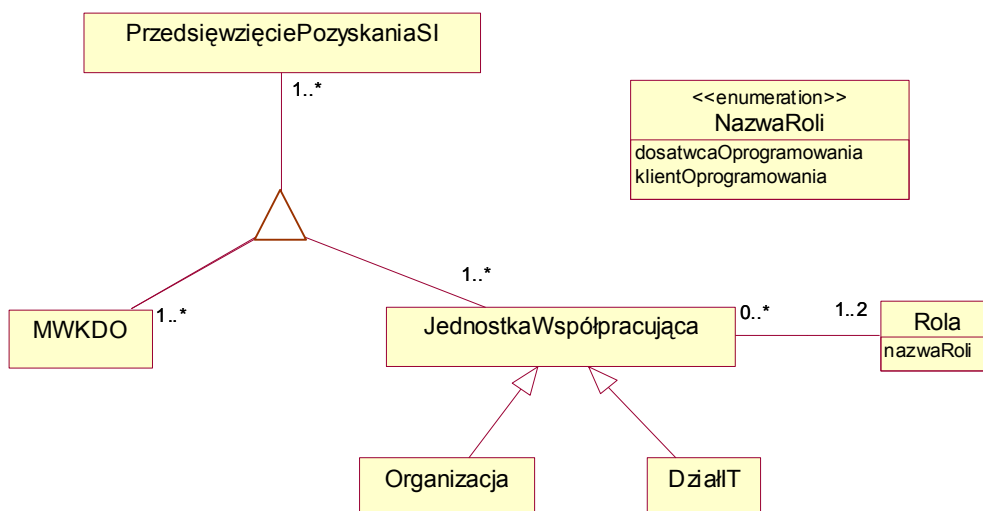
Tabela 1. Cechy MWKDO.

l.p.	Podstawowa cecha MKWDO	Korzyść biznesowa
1.	Szczegółowość	<ul style="list-style-type: none"> • Uszczegółowienie zakresu współpracy klienta i dostawcy oprogramowania. • Zmniejszenie niepewności, a tym samym zwiększenie zaufania dla przebiegu przedsięwzięcia. • Wiedza oraz kontrola współpracy; uniknięcie syndromu „my i oni” [SIW2003]; ustanowienie punktu odniesienia w przypadku konfliktu
2.	Podejście procesowe	<ul style="list-style-type: none"> • Wymusza myślenie systemowe o pozyskiwaniu oprogramowania, jako

		<p>przyczynowo-skutkowej zależności współpracy klienta i dostawcy.</p> <ul style="list-style-type: none"> • Umożliwia postrzeganie współpracy w szerszym kontekście.
3.	Otwartość	<ul style="list-style-type: none"> • Możliwość integracji z metodą identyfikacji ryzyka na podstawie modelu [RISK2007]. • Możliwość analizy i zarządzania ryzykiem obszaru współpracy.
4.	Wsparcie dokumentacyjne	<ul style="list-style-type: none"> • Stanowi naturalny środek dokumentowania przebiegu współpracy w trakcie procesu pozyskiwania oprogramowania. • Istnienie takiej dokumentacji tworzy warunki do analiz, które mogą wskazać drogi obniżenia kosztów dla przyszłych kontraktów zarówno dla klienta jak i dostawcy.
5.	Komunikatywność	<ul style="list-style-type: none"> • graficzna forma opracowania modelu – stereotypy graficzne, diagramy UML (ang. Unified Modeling Language) [UML2006], postać witryny internetowej - skutkują przyjazną formą prezentacji procesów współpracy w ramach pozyskiwania oprogramowania (forma czytelna dla wielu grup zawodowych).
6.	Precyzja	<ul style="list-style-type: none"> • formalna składnia modelu (diagramy UML oraz standard UMA [UMA2006] umożliwiają precyzyjną identyfikację ról, zadań, produktów pracy, szablonów dokumentów oraz procesu współpracy klienta i odbiorcy oprogramowania.
7.	Ukierunkowany na usługi	<ul style="list-style-type: none"> • interfejs zarówno dla klienta jak i dostawcy wspomaga dobór dostawcy potrafiącego dostosować się do MWKDO, co skutkuje efektywniejszą współpracą.

3. Obszary zastosowania MWKDO

Modele MWKDO umożliwiają zdefiniowanie współpracy następujących podmiotów: (a) klienta oprogramowania, (b) wewnętrznego dostawcy oprogramowania, (c) zewnętrznego dostawcy oprogramowania. Ilustruje to rysunek 1.



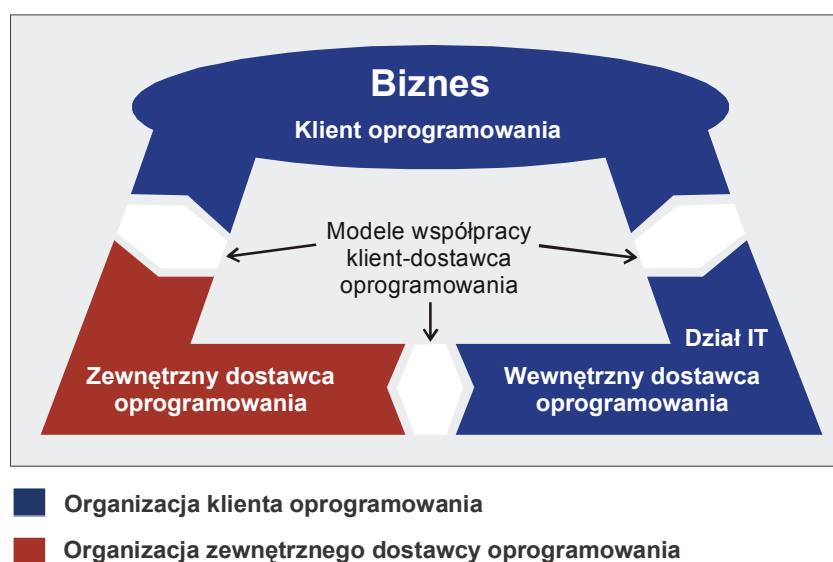
Rysunek 1. Model konceptualny modelowania interakcji klienta i dostawcy oprogramowania

W modelu wyróżniono trzy podstawowe obszary współpracy, które integrują współpracujące strony – rysunek 2. Pełnią one funkcję interfejsów poprzez które komunikują się ze sobą: Klient oprogramowania, wewnętrzny dostawca oprogramowania i zewnętrzny dostawca oprogramowania. Zgodnie z modelem współpracę należy rozpatrywać w układzie wewnętrznym i zewnętrznym, gdzie układ wewnętrzny to:

- Klient oprogramowania – współpracujący z własnym Działem IT (wewnętrzny dostawca oprogramowania),

natomiast układ zewnętrzny to:

- Klient oprogramowania współpracujący z zewnętrzną dostawcą oprogramowania oraz
- Dział IT (wewnętrzny dostawca oprogramowania) współpracujący z zewnętrzną dostawcą oprogramowania.

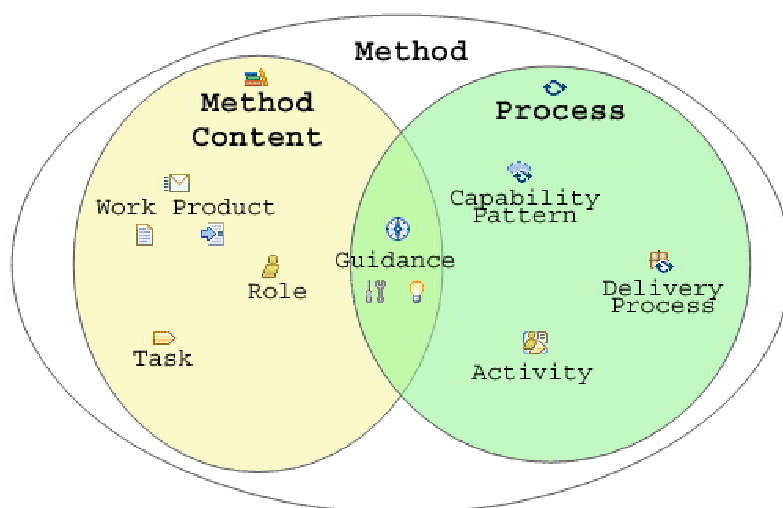


Rysunek 2. Podstawowe obszary współpracy klienta i dostawcy oprogramowania.

Warto podkreślić, że zgodnie z koncepcją MWKDO (rysunek 1), w różnych przedsiębiorstwach konkretna organizacja może występować zarówno w roli klienta jak i dostawcy oprogramowania.

4. Standard UMA

Proces współpracy klienta i dostawcy oprogramowania jest częścią szeroko rozumianego procesu inżynierii oprogramowania (ang. Software Engineering Process). Daje to możliwość zastosowania metamodeli procesu inżynierii oprogramowania do budowy MWKDO. Do niedawna dominujący w tym zakresie był standard SPEM (ang. Software Process Engineering Metamodel) - propozycja organizacji OMG (ang. Object Management Group) [SPEM2005]. Inny, zaproponowany w 2005 roku standard, którego celem jest unifikacja najpopularniejszych metamodeli procesu inżynierii oprogramowania to standard UMA (ang. Unified Method Architecture) [UMA2006]. Definiuje on terminologię i klasy, które umożliwiają zdefiniowanie zawartości i procesów danej metody (w ujęciu UMA, MWKDO jest metodą, którą definiujemy zgodnie z tym metamodelem). Rysunek 3 przedstawia koncepcję organizacji głównych abstrakcji standardu UMA.



Rysunek 3. Koncepcja organizacji głównych abstrakcji standardu UMA [RUP2005]

Zgodnie ze standardem metoda składa się z części statycznej tj. zawartości (ang. Method Content) oraz części dynamicznej tj. procesu (ang. Process). Elementami zawartości są:

- a) Produkt pracy (ang. Work Product),
- b) Rola (ang. Role),
- c) Zadanie (ang. Task).

Elementami procesu są:

- a) Czynność (ang. Activity),
- b) Wzorzec zdolności (ang. Capability Pattern),
- c) Proces dostarczania (ang. Delivery Process).

Elementami wspólnymi dla części dynamicznej i statycznej jest przewodnik (ang. guidance), którego składnikami są:

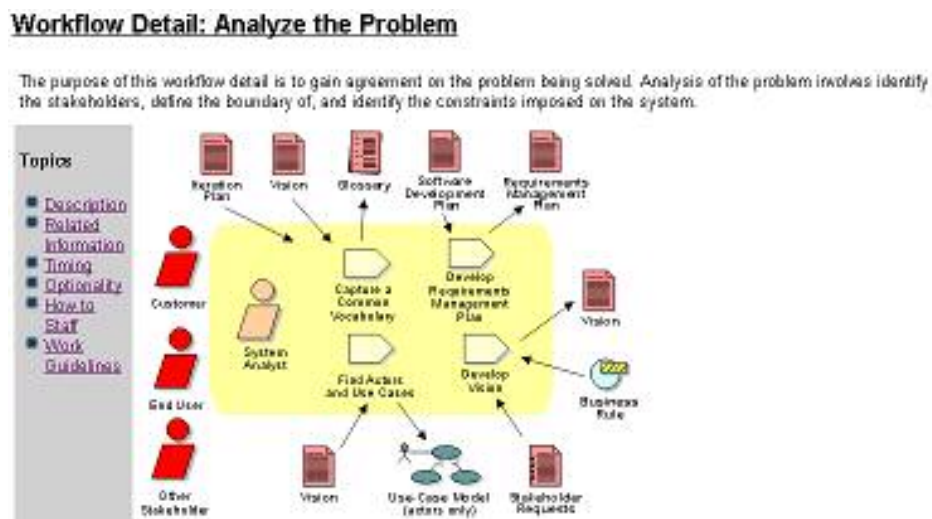
- a) Lista kontrolna (ang. Checklist),

- b) Koncepcja (ang. Concept),
- c) Przykład (ang. Example),
- d) Wskazówka (ang. Guideline),
- e) Praktyka (ang. Practice),
- f) Raport (ang. Report),
- g) Zasób ponownego użycia (ang. Reusable Asset),
- h) Mapa przewodnia (ang. Roadmap),
- i) Materiał pomocniczy (ang. Supporting Material),
- j) Szablon (ang. Template),
- k) Definicja pojęcia (ang. Term Definition),
- l) Doradca narzędziowy (ang. Tool Mentor).

5. MWKDO jako rozszerzenie IBM RUP

Model IBM RUP (ang. Rational Unified Process) dostarcza metodycznej wiedzy, która umożliwi efektywne realizowanie cyklu życia systemu przez zespół wytwórczy, koncentrując się na części związanej z produkcją oprogramowania. MWKDO można traktować jako rozszerzenie modelu IBM RUP o zakres dotyczący współpracy klienta i dostawcy oprogramowania (dotyczy to również modelu OpenUP [OUP2006] - lekkiej metodyki opartej na RUP). Głównym elementem rozszerzenia są role dziedzinowe, specyficzne dla środowiska pozyskującego oprogramowanie danej klasy.

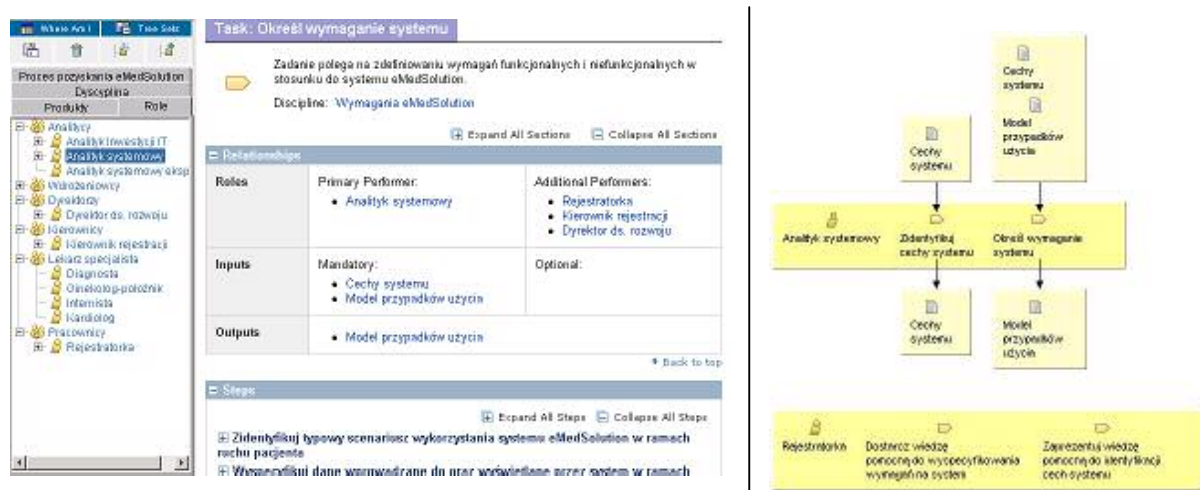
Przykładowo, wersja IBM RUP 2003 [RUP2003] definiuje następujące role zespołu wytwórczego: 4 analityków, 7 deweloperów, 7 menedżerów, 5 pracowników pomocniczych, 3 testerów oraz 5 ról dodatkowych. Dla udziałowców spoza zespołu wytwórczego zdefiniowano wyłącznie trzy role: Klient, Użytkownik końcowy oraz Inny Udziałowiec – rysunek 4. Są one uogólnieniem ról dziedzinowych charakterystycznych dla danej dziedziny przedmiotu. Przyjmując takie uogólnienie, nie wskazuje się jednoznacznie ról współpracujących w tym przypadku z analitykiem systemowym.



Rysunek 4. Role zaangażowane w analizę problemu w metodyce IBM RUP [RUP2003].

MWKDO umożliwia poszerzenie zestawu ról o role specyficzne dla dziedziny odbiorczej, dla której jest przeznaczone wytwarzane oprogramowanie. Na przykład, w studium przypadku dotyczącym zastosowania MWKDO dla dostawcy systemów medycznych (szpitale i

przychodnie) zidentyfikowano szereg ról dziedzinowych. Rysunek 5 prezentuje fragment MWKDO, który pokazuje współpracujące role dziedzinowe wraz z jedną rolą zespołu projektowego - Analitykiem Systemowym. Model wykonano przy użyciu narzędzia IBM Rational Method Composer [RMC2007].



Rysunek 5. Role zespołu projektowego oraz dziedzinowe role MWKDO

W tym przypadku Analityk Systemowy współpracuje z Rejestratorką, Kierownikiem rejestracji oraz Dyrektorem ds. rozwoju. Zawartość ta dotyczy definiowania wymagań względem modułu związanego z ruchem pacjenta. Dla tzw. części białej (medycznej) wymagane są już inne role i będą to lekarze specjaliści: Internista, Kardiolog, Pielęgniarka itd.

6. Porównanie obszarów wiedzy PMBok®PMI z MWKDO

MWKDO koncentruje się wyłącznie na współpracy klienta i dostawcy oprogramowania. Jego zawartość to procesy, w których następuje interakcja między tymi dwoma stronami. Metodyka PMBok®PMI [PMI2004] koncentruje się na obszarze zarządzania przedsięwzięciem, proponując najlepsze praktyki dla procesów tego obszaru. Tabela 2 przedstawia porównanie zasięgu MWKDO w stosunku do standardu PMBok®PMI. Jako kryterium porównawcze przyjęto 9 obszarów wiedzy metodyki PMBok.

Tabela 2. Porównanie metodyki PMBok®PMI z MWKDO

	Obszary wiedzy PMBok®PMI	MWKDO
Zarządzanie	integralnością projektu	tak
	zakresem	tak
	czasem	tak
	kosztami	-
	jakością	tak
	zasobami ludzkimi	tak
	komunikacją	tak
	ryzykiem	tak
	zaopatrzeniem	tak

7. Główne korzyści z zastosowania MWKDO

Główne korzyści z zastosowania modelowania do opisu współpracy klienta i dostawcy oprogramowania można rozpatrywać w trzech aspektach:

- (a) wspólne,
- (b) dla klienta oraz
- (c) dla odbiorcy oprogramowania.

Korzyści wspólne to:

- a) opracowanie koncepcji współpracy w ramach przedsięwzięcia informatycznego,
- b) możliwość ciągłego usprawnianie obszaru współpracy (interakcji) klienta i dostawca, a w konsekwencji poprawianie jakości współpracy.

Korzyści klienta oprogramowania to:

- a) możliwość uzyskania przewagi konkurencyjnej w obszarze pozyskiwania oprogramowania,
- b) zwiększenie tempa pozyskiwania oprogramowania,
- c) możliwość pozyskania dostawcy oprogramowania o największym potencjale do efektywnej współpracy.

Korzyści dostawcy oprogramowania to:

- a) posiadanie ogólnego, generycznego (a więc nadającego się do powtórznego wykorzystania) modelu współpracy dla danej klasy systemów,
- b) możliwość uzyskania przewagi konkurencyjnej w obszarze dostarczania oprogramowania,
- c) zwiększenie tempa dostarczania oprogramowania.

8. Podsumowanie

W referacie zaprezentowano modelowanie jako podejście do zdefiniowania współpracy klienta i dostawcy oprogramowania. Istotą tego podejścia jest identyfikacja wszystkich interakcji klienta i dostawcy w ramach współpracy, która ma na celu zaopatrzenie klienta w oprogramowanie. MWKDO wspiera udziałowców obu stron współpracy oraz, że może być traktowany jako rozszerzeniem metodyki IBM RUP. Proponowane podejście ułatwia modelowanie zróżnicowanych obszarów współpracy, a dana współpracująca jednostka może w nich występować w dwóch rolach klienta i odbiorcy – zależnie od jej udziału w danym przedsięwzięciu informatycznym.

Choć MWKDO nie obejmuje wszystkich procesów dotyczących tworzenia i przekazywania oprogramowania, to w opinii autorów jest on istotnym rozszerzeniem dotychczasowych propozycji o obszar współpracy klienta i dostawcy. Zastosowanie tego modelu może usprawnić taką współpracę, tworząc środowisko efektywnego przekazywania i pozyskiwania oprogramowania.

Model MWKDO jest obecnie w trakcie rozwoju. Jego pierwsze zastosowania dotyczą klasy systemów medycznych – obecnie opracowywane jest nietrywialnie studium przypadku we współpracy z partnerami, którymi są: szpital (klient) oraz dostawca oprogramowania wspomagającego funkcjonowanie szpitali. Badania obejmują również opracowanie ogólnego modelu, który będzie przewodnikiem dla tworzenia MWKDO w ramach współpracy dotyczącej pozyskania systemów różnych klas. Planowane jest, że również ten model będzie zgodny ze standardem UMA.

BIBLIOGRAFIA

- [EPF2007] Eclipse Process Framework Project, <http://www.eclipse.org/epf/>, 2007.
- [MCC1993] McConnell S., *Code Complete*, Microsoft Press, 1993.
- [OUP2006] *OpenUP_Basic_published-0.9-20061002*,
http://www.eclipse.org/epf/downloads/openup/openup_downloads.php, 2007.
- [PMI2004] Project Management Institute, *A Guide to the Project Management Body of Knowledge – Third Edition*, 2004.
- [RISK2007] Strona Projektu RiskGuide, <http://iag.pg.gda.pl/RiskGuide/>, 2007.
- [RMC2007] IBM Rational Method Composer, <http://www-306.ibm.com/software/awdtools/rmc/index.html>, 2007.
- [RUP2003] *Rational Unified Process 2003.06.00*, 2003.
- [RUP2005] *Rational Unified Process v7.0*, 2005.
- [SEI2004] CMMI® Acquisition Module (CMMI-AM), Version 1.0, Technical Report CMU/SEI-2004-TR-001
<http://www.sei.cmu.edu/publications/documents/04.reports/04tr001.html>, 2004.
- [SIW2003] Kubiak B.F., *Strategia informatyzacji współczesnej organizacji. Teoria i praktyka*, Akwila 2003.
- [SPEM2005] Object Management Group, *Software Process Engineering Metamodel Specification, Version 1.1*, 2005.
- [UMA2006] Unified Method Architecture: *Meta-Model Specification version 1.0.2*, 2006.
- [UML2006] Object Management Group, *Unified Modeling Language Specification, Version 2.1 Superstructure*, 2006.