



Artur Machura

Uniwersytet Ekonomiczny w Katowicach
Wydział Informatyki i Komunikacji
Katedra Informatyki
artur.machura@ue.katowice.pl

KONCEPCJA WYTWARZANIA ONTOLOGII DZIEDZINOWEJ INŻYNIERII WYMAGAŃ

Streszczenie: Artykuł, koncentrując się na warstwie praktycznej rozważań, dąży do odpowiedzi na pytanie – jak stworzyć model ontologiczny inżynierii wymagań, uwzględniając unikatowe zastosowanie współczesnych metod? Sformułowany proces tworzenia ontologii wymagań projektowych opiera się na przepływie sterowania pomiędzy głównymi aktywnościami, takimi jak: kreowana strategia przedsiębiorstwa, realizacja projektu informatycznego, eksploatacja rozwiązania. W artykule przedstawiony jest projekt bazujący na tzw. przypadkach użycia, który pozwala na powiązanie różnych dziedzin kompetencyjnych, tj. zapewnienia równowagi pomiędzy IT a biznesem (IT BSC Model by Van Grembergen), procesu wytwórczego oprogramowania (Open Unified Process), procesu tworzenia samej ontologii (Unified Process Ontology Development). Przedstawiony w artykule problem może mieć istotne implikacje badawcze, albowiem w rezultacie wykorzystania opisywanego podejścia zastosowania inżynierii wymagań określona grupa przedsięwzięć informatycznych może liczyć na wysoki poziom jakości, przy de facto stosunkowo niskim koszcie jej zapewnienia. Uporządkowany proces tworzenia ontologii zastosowania inżynierii wymagań wiąże w sposób unikatowy odrębne metody (IT BSC Model, Open UP, UPON) – na potrzeby realnego przedsiębiorstwa.

Słowa kluczowe: ODS, ODRE, UPON, OPENUP, IT BSC Model, UML.

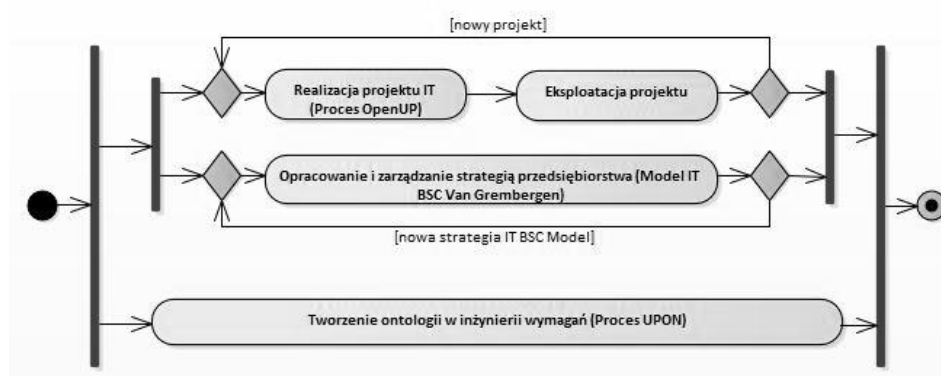
JEL Classification: A12, C8.

Wprowadzenie

Celem artykułu jest wstępna organizacja procesu tworzenia ontologii dziedzinowej inżynierii wymagań. Rozważania dążą do wsparcia pracy poświęconej wydobywaniu wymagań w projektach systemów informatycznych, stanowiąc odpowiedź na współczesne problemy projektowe systemów informatycznych, tj. brakujące, niekompletne i zmieniane wymagania.

W artykule przyjęto punkt widzenia T. Grubera, który poza zdefiniowaniem samej definicji ontologii („Ontologia jest formalną, jawną specyfikacją współdzielonej konceptualizacji”), określił również zasady oceny systemów ontologii (jasność, spójność, przejrzystość, minimalne zaangażowanie formalizacyjne oraz ontologiczne; www 1).

Zauważa się na wstępie znaczącą złożoność oraz istotne różnice pomiędzy inżynierią ontologii oraz inżynierią oprogramowania, wobec których proces tworzenia ontologii zastosowania inżynierii wymagań nie może pozostać obojętnym. Dokonuje się ich wstępnego uzgodnienia, względem przyszłego zastosowania efektu tworzonego procesu przez określoną grupę przedsiębiorstw. Spośród wielu możliwości wybrano rzeczywiste potrzeby przedsiębiorstw usługowych i realizowanych przez te przedsiębiorstwa takich przedsięwzięć informatycznych, które wspierają zasadniczo ich konkurencyjność. Projekt stanowi kontekst użycia prezentowanego procesu. Jednak ze względu na zakres artykułu, nie został w nim ten projekt opisany. Poniższy rys. 1 obrazuje ogólnie przepływ sterowania pomiędzy inżynierią ontologii, inżynierią oprogramowania i strategią przedsiębiorstwa. Na jego podstawie zostanie docelowo sformułowany proces tworzenia ontologii. Natomiast zawiera on wstępne uzgodnienie pomiędzy tymi obszarami wiedzy a formułowanym procesem.



Rys. 1. Przepływ sterowania pomiędzy czynnościami: tworzenie procesu ontologii, budowa ontologii, implementacja projektu, eksploatacja rozwiązania

Źródło: Opracowane własne.

Tabela 1. Uzgodnienie stanowiska tworzenia ontologii w inżynierii wymagań

	Inżynieria oprogramowania	Inżynieria ontologii	Proces tworzenia ontologii w inżynierii wymagań
1	2	3	4
Okres	<ul style="list-style-type: none"> określony ograniczony do jednego projektu 	<ul style="list-style-type: none"> nieokreślony nieograniczony 	<ul style="list-style-type: none"> okres realizacji strategii konkurencyjnej
Struktura	<ul style="list-style-type: none"> fazy zgrupowanie zadań w iteracjach i działaniach 	<ul style="list-style-type: none"> ewolucja cykli zgrupowane w fazach i działaniach 	<ul style="list-style-type: none"> ewolucja cykli zgrupowanie zadań w iteracjach i działaniach
Podproces	<ul style="list-style-type: none"> dla komponentów dla przyrostów lub specjalnych zadań (np. Quality Assurance) 	<ul style="list-style-type: none"> w celu rozwinięcia lub zmiany poddomeny 	<ul style="list-style-type: none"> dla komponentów, przyrostów lub specjalnych zadań w celu rozwinięcia lub zmiany poddomeny
Paradygmaty	<ul style="list-style-type: none"> wodospad przyrostowy oparty na podzespołach prototyp spiralny 	<ul style="list-style-type: none"> przyrostowe oparte na komponentach ewolucyjność 	<ul style="list-style-type: none"> iteracyjno-przyrostowy
Aktywności i artefakty	<ul style="list-style-type: none"> znajdowanie i opisywanie przypadków użycia wydobywanie wymagań budowanie koncepcji struktury modelu/klaszary określenie komponentów i modułów kodowanie: użycie/generowanie konkretnego języka programowania testowanie i debugowanie integracja testowanie podsystemów wdrożenia systemów sprawdzanie poprawności dostosowywanie się do środowiska uzyskiwanie informacji zwrotnych użytkowników rozpoczęcie rewizji (jeśli to konieczne) 	<ul style="list-style-type: none"> zidentyfikowanie potencjalnych aplikacji analizowanie terminologii budowa taksonomii budowa i wypełnianie słownika definiowanie faktów i reguł porównywanie z innymi glosariuszami rozwiązywanie konfliktów terminologicznych tłumaczenia do formalnego języka: linkowanie i walidacja subontologii dostosowywanie do wspólnych/sąsiadujących ontologii publikowanie ontologii poprawnych otrzymywanie informacji zwrotnych o pozyskanych wymaganiach w trakcie rewizji 	<ul style="list-style-type: none"> opracowywanie strategii informatycznej (IT Balanced Scorecard) zarządzanie pomiarami i wydajnością (metryki, arkusze i raporty) wymogi dotyczące pracy proces analizy biznesowej i inżynierii wymagań (identyfikacja i priorytetyzowanie przypadków użycia) analizy (budowanie leksykonu domeny i więcej) projektowanie (modelowanie koncepcyjne, powiązania i więcej) implementacja (formalna reprezentacja ontologii w OWL) testowanie (sprawdzanie zawartości)
Języki/opisy narzędzia	<ul style="list-style-type: none"> diagramy przypadków użycia języki naturalne E/R diagram UML pseudokodowanie języki programowania 	<ul style="list-style-type: none"> schematy E/R w języku naturalnym słowniki UML sieci semantyczne DAML + OWL i inne 	<ul style="list-style-type: none"> zależne od etapu projektu (UML, OWL, Eclipse)

cd. tabeli 1

1	2	3	4
Grupa docelowa	<ul style="list-style-type: none"> • deweloperzy • użytkownicy • klienci 	<ul style="list-style-type: none"> • eksperci domenowi • deweloperzy • agenci innych systemów 	<ul style="list-style-type: none"> • klienci • użytkownicy • programiści • eksperci dziedzinowi i inne zainteresowane strony
Rezultaty i produkty	<ul style="list-style-type: none"> • specyficzne dla danego przedsięwzięcia • zorientowane przeważnie na krótkoterminowe zastosowanie i nienadające się do ponownego użytku 	<ul style="list-style-type: none"> • nakierowane na zastosowanie w wielu projektach • wielokrotnego użycia przez różne organizacje 	<ul style="list-style-type: none"> • zorientowane długoterminowo dla specyficznych grup przedsiębiorstw i typów projektów

Źródło: [www 2].

1. Przegląd literatury

Doświadczenia uczestników i interesariuszy projektów pozwoliły już w 1968 roku na konferencji NATO [www 3] stwierdzić potrzebę realizacji projektów z perspektywy określonej terminem *inżynieria oprogramowania*. U podstaw tego terminu znajduje się dbanie o jakość przy wzajemnej zależności: procesów wytwórczych, metod, narzędzi [Pressman, 2004]. Spostrzeżenia z tamtego czasu obrazują do dzisiaj praktyka i zarazem teoria poświadczona w raportach takich organizacji, jak np. Standish Group czy też PMResearch. Z badań tych organizacji wynika, że trzy najczęściej spotykane problemy projektów informatycznych to: brak informacji wejściowych od użytkownika, niekompletność tych informacji oraz zmiany wymagań i specyfikacji. Konsekwencją tego m.in. jest fakt, że 31% projektów zostało przerwanych przez planowanym zakończeniem, 53% kosztowało ponad 189% estymowanych wartości, a tylko 16% zakończyło się zgodnie z planem [Stanek, 2013].

Warty uwagi jest również kontekst rynkowy i zmieniające się oczekiwania użytkowników, bowiem pierwsze rozwiązania informatyczne były jedynie synonimem nowoczesności bez większego przywiązania do kultury organizacyjnej przedsiębiorstw stosujących te rozwiązania. To doprowadziło do wniosków jeszcze w latach 90. ubiegłego stulecia o istocie analizy procesów biznesowych i dostosowywaniu rozwiązań informatycznych do potrzeb organizacji, a więc i wymagań. Ponadto oczekiwania rynkowe XXI wieku podnoszą nieustannie poprzeczkę odnośnie do jakości tych rozwiązań. Akceptacja informatycznego projektu inwestycyjnego może nastąpić tylko wtedy, gdy zostaną zidentyfikowa-

ne i zmierzone korzyści, wśród których decydującą rolę odgrywają te, które w sposób bezpośredni lub pośredni zapewniają wzrost wartości rynkowej organizacji [www 4]. Tendencja ta, wraz z rosnącym popytem na zastosowanie rozwiązań informatycznych w gospodarce, podkreśla wagę zapewnienia jakości w przedsięwzięciach. Jedną z kluczowych definicji jakości międzynarodowej organizacji – Institute of Electrical and Electronics Engineers (IEEE) – utożsamia jakość z satysfakcją użytkownika, na którą składają się: dobra jakość produktu, działanie produktu, zrealizowany budżet i harmonogram [Glass, 1998].

Wymienione lata 90. ubiegłego stulecia przyczyniły się do rozwoju metod i języków wspierających analizę potrzeb użytkowników systemów informatycznych, wśród których upowszechnił się standard organizacji o globalnym zasięgu Object Management Group (OMG) – zuniifikowany język modelowania UML (Unified Modeling Language). Jednak rzeczywiste zastosowanie języka w przedsięwzięciach informatycznych wiąże się z innymi perspektywami, np. dojrzałością procesu wytwórczego, o czym mowa pośrednio w standardzie CMMI (Capability Maturity Model Integration) opracowanym przez SEI (Software Engineering Institute). Na podstawie doświadczeń własnych autora uczestnictwa w wieloletnim projekcie, w którym zastosowano język UML oraz RUP (Rational Unified Process), formułuje się wniosek, że produkty analityczne powinny wiązać dwa równoległe i skomplikowane środowiska: 1) potrzeby organizacji – klienta, późniejszego użytkownika systemu, 2) zespołu wytwórczego wykonawcy i wynikających z tego właściwości [Machura, 2012].

Ten nurt badań i obserwacji jest również istotny w kontekście ewaluacji opłacalności realizacji przedsięwzięć informatycznych. Można wnioskować, że analiza biznesowa poparta warsztatem, tj. IIBA (International Institute of Business Analysis), mogłaby wspierać konkretne metody szacowania wspomnianej opłacalności [Machura, 2013]. Innymi słowy, w przypadku utrudnionego zastosowania tego warsztatu, poza problemami udanych projektów w ogóle, można zaobserwować trudności związane z poprawną inicjacją projektów i uzasadnieniem ich opłacalności. Poszukiwania swego rodzaju klucza do realizacji przedsięwzięć udanych doprowadziły do przeglądu metod na międzynarodowym rynku (Pańkowska, Machura, 2014; por. tab. 2). Poddano analizie wiele punktów widzenia, związanych z inicjacją projektu i zastosowaniem inżynierii wymagań. W dalszej części pracy dla realizacji celów badawczych dokonano wyboru użytecznych metod i modeli.

Tabela 2. Przegląd modeli i perspektyw postrzegania inżynierii wymagań

Model	Charakterystyka modelu
ISACA (Information System Audit and Control Association)	Model zwraca szczególną uwagę na aspekt kontroli i zarządzania strategicznego wszystkich procesów IT dla zapewnienia, że procesy zaspakajają wymagania biznesu
IIBA – BABOK (Business Analysis Body of Knowledge)	Model ujmuje konkretne etapy i zadania analityka biznesowego. Rozciąga tę pracę i jej wpływ na cały okres realizacji projektu
IBAQB – GASQ (Global Association for Software Quality)	Podobnie jak w IIBA, ujmuje konkretne etapy i zadania analityka biznesowego, przy czym dostrzega się tu pewne różnice, chociażby w kategoryzacji wymagań
BITA (Business IT Alignment)	Kluczową cechą jest scalenie strategii biznesowej przedsiębiorstwa z efektywnym zastosowaniem rozwiązań IT
Modele architektury przedsiębiorstwa	Model koncentruje się na prezentacji struktur organizacyjnych, technologii i systemów informatycznych
Product Line Requirements Management	Model przeznaczony dla środowisk wytwarzających produkty o wspólnych cechach. Praca oscyluje wokół rozwoju tzw. rodziny produktów
Formalne procesy produkcji oprogramowania	Podjęcie to umiejscawia inżynierię wymagań w określonym miejscu procesu produkcji oprogramowania, nadając konkretną odpowiedzialność i zależności względem pozostałych dyscyplin inżynierii oprogramowania
Zwinne procesy produkcji oprogramowania	Model cechuje się dużą dynamiką i elastycznością wynikającą z organizacji opartej o współpracę ludzi. Modele oparte o tzw. Manifest zwinnego wytwarzania oprogramowania
PMI – Business Analysis for Practitioners a Practice Guide	Model ujmuje konkretne etapy i zadania analityka biznesowego, zwraca uwagę na praktyczne aspekty zarządzania całymi przedsięwzięciami i wytwarzania oprogramowania

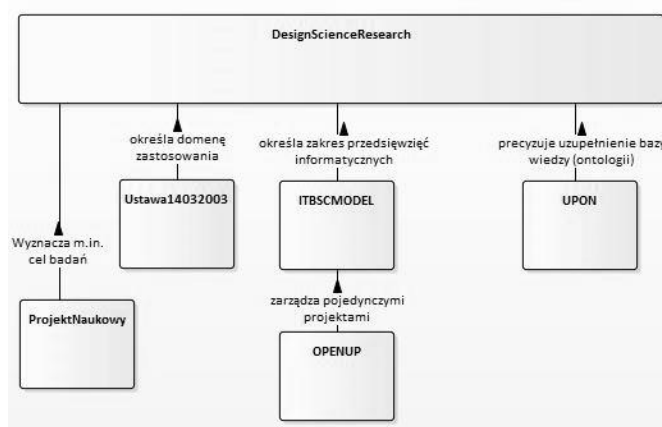
Źródło: Pańkowska, Machura [2014].

Model Business IT Alignment (BITA) to szczególnie istotny dla podjętych rozważań warsztat metodyczno-narzędziowy, pozwala bowiem na realizację przedsięwzięć z perspektywy tzw. celów wyższego rzędu, np. celów biznesowych, jakościowych, technologicznych, konkurencyjnych [Frączkowski, 2003]. Zastosowanie tego modelu wymaga szerszego, strategicznego ujęcia przedsiębiorstwa. Autor, na podstawie wcześniejszych doświadczeń, podjął się realizacji projektu wykorzystującego jedno z podejść wspierających strategiczną równowagę pomiędzy IT a biznesem. Podjęta w wymienionej publikacji problematyka perspektywy czasu niezbędnego w realizacji strategicznych celów przedsiębiorstwa została dokładniej zbadana we współpracy z ARL S.A. i grupą zainteresowanych przedsiębiorstw. W rezultacie opublikowano artykuł, w którym zastosowano Model Van Gremebergen IT BSC (Information Technology Balanced Scorecard) oraz zauważono niebagatelne utrudnienia zastosowania tego warsztatu w praktyce przedsiębiorstw małych i średnich funkcjonujących co najmniej na terenie Polski [Machura, Szopa, 2016].

W rezultacie tych spostrzeżeń dalszy przegląd literatury międzynarodowej w dziedzinie ontologii inżynierii wymagań, wraz z doświadczeniami zdobywanymi w drodze udziału w rzeczywistych projektach, z biegiem lat doprowadził do koncentracji uwagi autora na podejściu naukowo-biznesowym, tj. ODSD (Ontology Driven Software Development) i ODRE (Ontology Driven Requirements Engineering) [Pan i in., 2011]. Okazuje się bowiem, że inżynieria ontologii wiąże się również z problematyką wytwarzania oprogramowania. W kolejnych punktach artykułu zostanie sformułowany proces tworzenia ontologii zastosowania inżynierii wymagań, z wykorzystaniem właśnie warsztatu metodycznego ontologii.

2. Metoda badań

Formułowany w niniejszym artykule proces tworzenia ontologii zorientowanej na ontologię inżynierii wymagań (Ontology Driven Requirements Engineering – ODRE) jest kompatybilny z podejściem do organizacji badań, zwanym paradygmatem DSR (Design Science Research) i opisanym przez Hevnera i in. [www 5]. Zastosowana w pracy [Machura, 2016b] metoda badawcza pozwoliła na realizację badań w zgodzie z paradygmatem badań DSR.



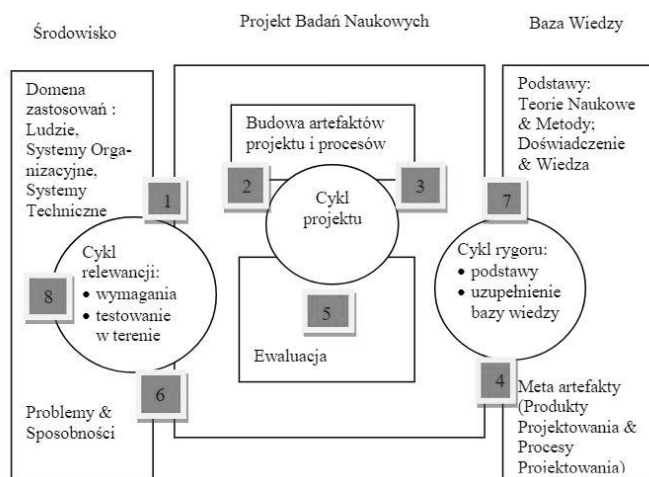
Rys. 2. Uzgodnienie wzajemnego oddziaływania obszarów na metodę badań

Źródło: Opracowane własne.

Na rys. 2 uzgodniono wzajemne oddziaływanie obszarów mających niebagatelny wpływ na końcowy scenariusz prowadzenia badań. Poniżej diagramu znajduje się opis poszczególnych klas, ich odpowiedzialności oraz podsumowanie całego prezentowanego modelu na przyjętą metodę badań.

2.1. Podejście Design Science Research

Paradygmat badań naukowych Hevnera i in. [www 5] jest zorientowany na rozwiązywanie problemów technicznych i konstrukcyjnych, co dla formułowanego procesu tworzenia ontologii w inżynierii wymagań ma niebagatelne znaczenie, bowiem wyznacza pewien model referencyjny, w ramach którego będzie się zarządzać kolejnymi zagadnieniami, tj. Ustawą 14032003, ITBSCModel, OpenUP, UPON. Model referencyjny Hevnera to przede wszystkim kolejne cykle: relewancji, projektu, rygoru.



Rys. 3. Cykle w paradygmacie DSR

Źródło: [www 5].

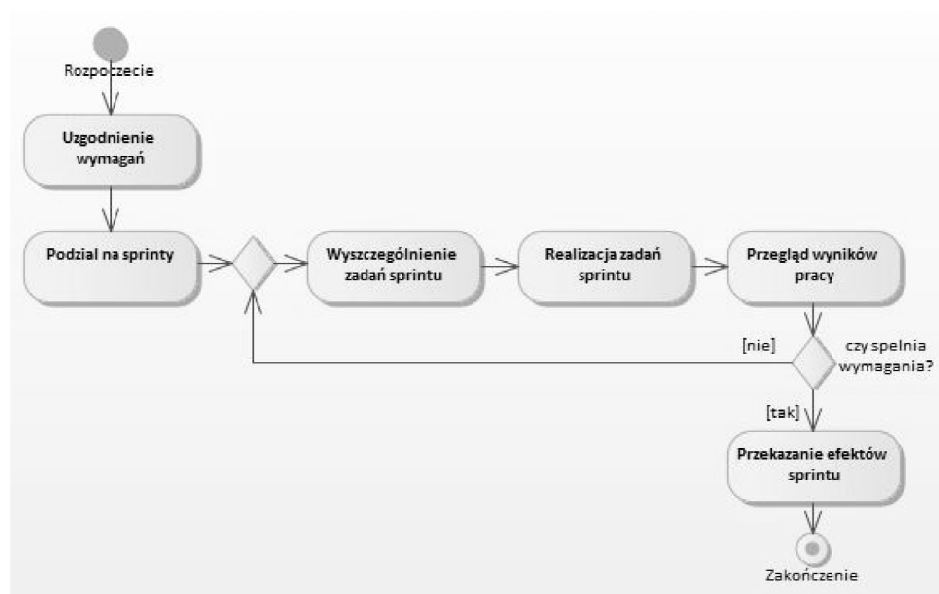
Cykl relewancji łączy środowisko kontekstowe projektu badań z działaniami naukowymi, co pozwala również odnosić wyniki badań do tego kontekstu. Cykl projekt pozwala badaczowi w sposób iteracyjny i inkrementalny na budowanie oraz ocenę artefaktów projektu i prac badawczych. Cykl rygoru natomiast łączy projektowanie i badania naukowe z bazą wiedzy. W rezultacie uzyskuje się tu odpowiedź na fundamentalne znaczenie – czy badanie wnosi wkład do nauki w postaci nowej wiedzy i doświadczeń? Tworzenie nowych artefaktów wzbogaca naukę.

Poniższa tab. 3 obrazuje kolejne etapy pracy DSR, propagowane przez Hevner i Chatterje. Natomiast kolejny rys. 4 podsumowuje postanowienia organizacyjne realizacji etapu ścieżki badawczej pojedynczego etapu/sprintu, gdzie dążono do zaadaptowania pewnych elementów zarządzania w sposób zwinny (AGILE), ponieważ praktyka naukowa i gospodarcza wymaga w tym przypadku takiego elastycznego i opartego na iteracjach podejścia.

Tabela 3. Etapy pracy badawczej Hevnera i Chatterje

Nazwa etapu pracy	Etapy/sprinty
<p>Identyfikacja problemu i motywacja Definiowanie specyficznych problemów badawczych i uzasadnienie znaczenia rozwiązania. Analiza zasobów wiedzy i analiza problemu dla pokazania złożoności rozwiązania. Przedstawienie znaczenia rozwiązania dla badaczy i pozostałych interesariuszy badania dla poszukiwania najlepszego rozwiązania i dla akceptowania wyników badania</p>	1
<p>Definicja celów badania Wnioskowanie odnośnie celu badania na podstawie definicji problemu i wiedzy dotyczącej tego, co jest możliwe i wykonalne. Rozwiązanie może być wyrażone opisowo lub mierzalne ilościowo i przedstawione w kategoriach rozwiązania lepszego niż aktualnie stosowane. Zasoby wymagane dla tego etapu obejmują wiedzę na temat stanu badań nad problemem, analizę aktualnych rozwiązań i ocenę ich skuteczności</p>	2
<p>Rozwój projektu Tworzenie artefaktów, czyli koncepcji, modeli, metod i przykładów wdrożeń, nowych własności technicznych, społecznych i informacyjnych zasobów. Artefaktem badania może być zaprojektowany przez badacza obiekt. Ten etap wymaga ustalenia pożądanej funkcjonalności obiektu i jego architektury oraz konstrukcji pozostałych artefaktów. Na tym etapie występuje przejście od celów projektu do jego realizacji i zastosowanie wiedzy teoretycznej w urzeczywistnianym rozwiązaniu</p>	3
<p>Demonstracja Przedstawienie użycia artefaktu na wybranych przez badacza przykładach. Ten etap wymaga zastosowania eksperymentu, symulacji, studium przypadku, analizy dowodów lub innej odpowiedniej metody jakościowej. Konieczne zasoby wiedzy dla realizacji tego etapu obejmują wiedzę, jak zastosować artefakty w praktyce</p>	4
<p>Ewaluacja Obserwacja i pomiar, na ile dobrze artefakt wspomaga praktykę. Analiza efektów zastosowania artefaktu wymaga wiedzy na temat odpowiednich metryk i technik analizy. Ewaluacja może obejmować porównanie funkcjonalności artefaktów, z odniesieniem do funkcjonalności najbardziej pożądanej, analizy budżetów i kosztów, badania satysfakcji użytkowników i klientów (zarówno ludzi biznesu, jak i nauki), symulacji funkcjonalnej. Brane są pod uwagę miary ilościowe funkcjonowania systemu, takie jak czas reakcji i dostępność. Ewaluacja może obejmować każdy odpowiedni dowód empiryczny lub logiczny. Pod koniec etapu badacz podejmuje decyzję o powrocie do etapu konstrukcji artefaktu dla udoskonalenia jego efektywności lub decyduje się na przejście do następnego kroku i pozostawia kwestie doskonalenia do rozstrzygnięcia w innych projektach. Na tym etapie badacz musi rozważyć, ile iteracji należy wykonać</p>	5
<p>Komunikacja dla dyfuzji wiedzy w społeczeństwie badaczy Komunikowane są problem badawczy, jego znaczenie i generowane artefakty, ich użyteczność, nowość, porządek projektu badawczego, użyte metody badawcze, efektywność działań badaczy, charakterystyka audytorium interesariuszy projektu, publikacje naukowe, dzięki którym w ogóle możliwa jest komunikacja naukowa</p>	6

Źródło: Machura [2016b].



Rys. 4. Realizacja sprintu badawczego

Źródło: Machura [2016b].

Reasumując wyżej wymienione postanowienia metodyczne, niniejszy projekt badawczy dąży do zwinnej realizacji celów wyszczególnionych przez paradygmat badań DSR.

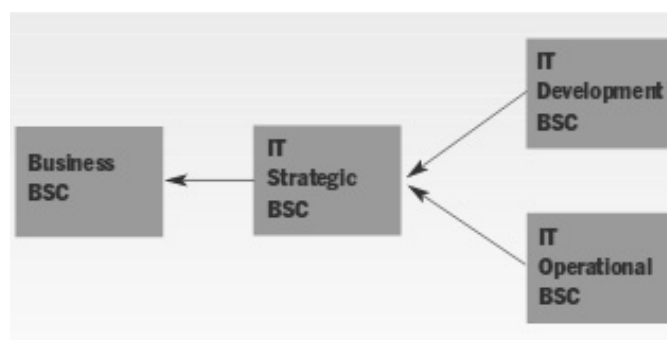
2.2. Projekt badawczy

Projekt badawczy określa zasadniczo cel i podporządkowuje merytoryczny aspekt opisywanej tu działalności naukowej. Ontologia w inżynierii wymagań ma stanowić rezultat wykorzystania opisywanego tu procesu. W drodze kolejnych iteracji i inkrementacji powstaje ontologia w inżynierii wymagań, co jest przedmiotem zainteresowania projektu badawczego.

2.3. Zastosowanie modelu Van Grembergena

Przedsięwzięcia informatyczne realizuje się z myślą osiągnięcia celów wyższego rzędu, gdzie jednym z nich jest zaspokojenie potrzeb konkurencyjności. Jest to możliwe w drodze realizacji efektywnej strategii wprowadzania technologii informatycznej (IT). W niniejszym artykule przyjmuje się, że zaproponowany przez Van Grembergena model IT BSC pozwala zarządzać tą strategią.

Tym samym wyznacza dla rozpatrywanego przedsiębiorstwa kluczowe i analizowane perspektywy. Wyniki przekładają się na konkretne działania, tym samym na wymagania względem m.in. systemów informatycznych. W niniejszym artykule przyjmuje się, że zaprezentowane obszary perspektyw IT BSC wyznaczają *de facto* dziedziny ontologiczne (rys. 5), w ramach których będą wytwarzane artefakty badawcze zgodnie z podejściem DSR.



Rys. 5. IT BSC Model by Van Grembergen

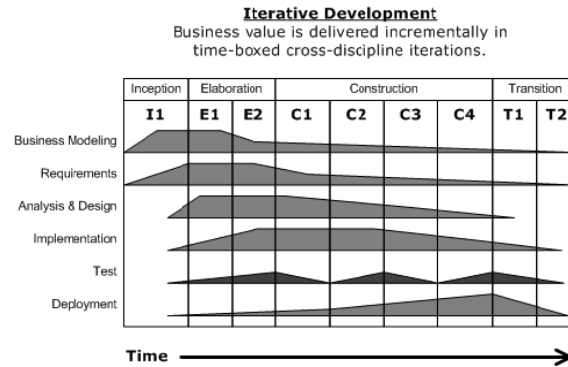
Źródło: Grembergen [2000].

Należy zwrócić uwagę na następujące założenia metodyczne:

- Business BSC Balanced Scorecard zawiera perspektywy: finansową, użytkownika, wewnętrzną, innowacji.
- IT Strategic Balanced Scorecard to perspektywy: wkładu korporacyjnego, użytkownika, wewnętrzną, innowacji.
- IT Development Balanced Scorecard dotyczy perspektyw: wkładu, użytkownika, doskonałości organizacyjnej, orientacji przyszłościowej.

2.4. Zastosowanie procesu wytwórczego OpenUP

Spośród wielu procesów wytwórczych wybrano na potrzeby realizacji projektu Open Unified Process (OpenUP), zważając z jednej strony na jego możliwy formalizm, natomiast z drugiej na elastyczność i zwinność podczas realizacji projektu. Proces OpenUP ma kluczowe cechy współczesnych procesów ewolucyjnych, tj. rozwój oprogramowania poprzez iteracje i przyrosty. Wobec tego z powodzeniem można wykorzystać pożądaną w projekcie perspektywę przypadków użycia. To w konsekwencji zapewni wspólną płaszczyznę komunikacji z wytwarzaną ontologią i wykorzystywanym procesem UPON (Unified Process Ontology Development).

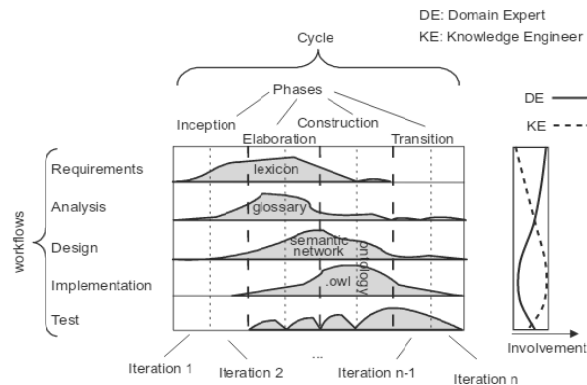


Rys. 6. Proces wytwarzania oprogramowania Open Unified Process

Źródło: [www 6].

2.5. Unified Process Ontology Development

Spośród dostępnych procesów wytwarzania ontologii wybrano UPON (Unified Process Ontology Development), zwracając uwagę na formalną możliwość sterowania rozwojem ontologii poprzez przypadki użycia.



Rys. 7. Unified Process Ontology Development (UPON)

Źródło: Nicola, Missikoff, Navigli [2005].

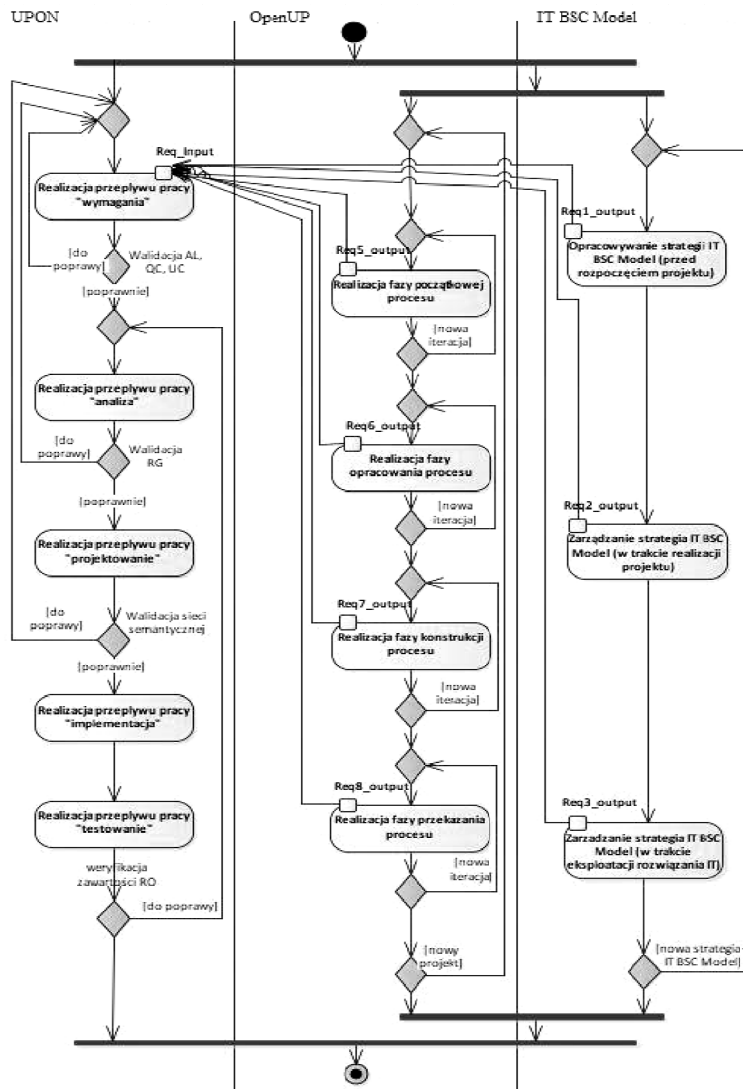
3. Rezultaty badań

Proces powiązania UPON, OpenUP, IT BSC

Poniższy proces (rys. 7), zaprezentowany z użyciem notacji UML (Unified Modeling Language) i diagramu aktywności, wiąże różne metody i służy wytwarzaniu ontologii w inżynierii wymagań w trakcie realizacji pierwotnie odrębnych:

1. Strategii przedsiębiorstwa z wykorzystaniem IT BSC Model by Van Grembergen.
2. Rozwoju oprogramowania z wykorzystaniem OpenUP.
3. Budowy ontologii z wykorzystaniem UPON.

Tor samego procesu UPON został uogólniony na podstawie artykułu źródłowego [Nicola, Missikoff, Navigli, 2009], skąd też pochodzą zalecenia samego autora procesu UPON względem miejsca na dokonywanie walidacji i weryfikacje.



Rys. 8. Proces tworzenia ontologii stosowanej inżynierii wymagań

Źródło: Opracowanie własne.

4. Dyskusja

Rezultatem ścieżki badawczej jest sformułowanie procesu tworzenia ontologii w inżynierii wymagań. Rozwijany w taki sposób model ontologiczny, w drodze ewolucji opartej na iteracjach i przyrostach – może stanowić alternatywne podejście dla zespołów wytwarzających oprogramowanie.

Podsumowanie literatury zawarte w tym artykule wskazuje na trwającą nieprzerwalnie pracę ukierunkowaną na rozwój warsztatu metodyczno-narzędziowego i identyfikowaną m.in. z profesją analizy biznesowej. Na uwagę w artykule zasługuje podejście równowagi pomiędzy IT a biznesem, niestety czasochłonne i z reguły zbyt kosztowne do uwzględnienia przez małe i średnie przedsiębiorstwa [Machura, Szopa, 2016]. Wobec specyfiki tej pracy i odrębności małych i średnich przedsiębiorstw, nasuwa się pytanie, czy inżynieria wymagań jest zarezerwowana wyłącznie dla dużych podmiotów dysponujących dużym budżetem projektowym. Rozważania na ten temat podjęto już we wcześniejszej publikacji [Machura, 2016a]. Przedstawione w tej pracy podejście pozwala na wnioskowanie o pewnym rozwiązaniu biznesowym, które znajduje swoje oparcie na gruncie naukowym. Takim warsztacie metodyczno-narzędziowym w inżynierii wymagań, który wychodzi naprzeciw problemom współczesnych przedsiębiorstw realizujących projekty informatyczne, gdzie mowa o:

- 1) braku informacji wejściowych od użytkownika,
- 2) niekompletności wymagań,
- 3) zmianie wymagań i specyfikacji.

W drodze weryfikacji empirycznej opisywanego procesu należy się spodziewać dalszego doskonalenia zaproponowanego podejścia, przez co należy rozumieć zarówno samą pracę, jak i wykorzystywane tu metody i narzędzia.

Podsumowanie

W artykule sformułowano proces tworzenia ontologii w inżynierii wymagań. Powiązано ze sobą w tym celu DSR, UPON (Unified Process Ontology Development), OpenUP (Open Unified Process) oraz Agile Method, IT BSC Van Grembergen.

Zaprezentowane w rezultacie podejście zapewnia w przedsięwzięciach informatycznych skuteczność specyfikacji wymagań odnoszących się do przyszłego rozwiązania, stanowi nowatorską alternatywę dla zespołów wytwórczych. W drodze realizacji analogicznych przedsięwzięć oczekuje się ewolucyjnego rozwoju ontologii inżynierii wymagań – rozwiązań informatycznych wspierających np. konkurencyjność przedsiębiorstw usługowych.

Literatura

- Frączkowski K. (2003), *Zarządzanie projektem informatycznym*, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław.
- Glass R.L. (1998), *Defining Quality Intuitively*, IEEE Software.
- Gołuchowski J., Smolarek M. (2014), *Semantyczne Modelowanie Organizacji*, Difin, Warszawa.
- Grembergen W. (2000), *The Balanced Scorecard and IT Governance*, Information Resources Management Association International Conference, Alaska.
- Machura A. (2012): *Praktyczne zastosowanie metod i narzędzi inżynierii oprogramowania* [w:] *Systemy Wspomagania Organizacji*, Wydawnictwo Uniwersytetu Ekonomicznego, Katowice, s. 301-319.
- Machura A. (2013), *Szacowanie opłacalności przedsięwzięć IT a warsztat metodyczno-narzędziowy analityka biznesowego*, Wiedza i Technologie Informacyjne, Częstochowa.
- Machura A. (2016a), *Inżynieria wymagań czy inżynieria możliwości?* PTZP, Zakopane.
- Machura A. (2016b), *Propozycja procesu pozyskiwania danych rynkowych na potrzeby realizacji badań poświęconych zastosowaniu informatyki w gospodarce*, WNiB, Wrocław.
- Machura A., Szopa A. (2016), *Model of an IT-Business Alignment Implementation in the Economic Practice*, aisel.aisnet.org/isd2014/proceedings2016/ (dostęp: 17.03.2017).
- Nicola A., Missikoff M., Navigli R. (2005), *A Proposal for a Unified Process for Ontology Building: UPON*, Dexa, Copenhagen.
- Nicola A., Missikoff M., Navigli R. (2009), *Software Engineering Approach to Ontology Building*, "Information Systems", No. 34, s. 260.
- Pan J.Z., Staab S., Assmann U., Ebert J., Zhao Y. (2011), *Ontology-Driven Software Development*, Springer, Heidelberg i in.
- Pańkowska M., Machura A. (2014), *Przegląd metod inżynierii systemów i wymagań w świetle rzeczywistych praktyk gospodarczych*, EMAG, Katowice
- Pressman R. (2004), *Praktyczne podejście do inżynierii oprogramowania*, WNT, Warszawa.
- Stanek S. (2013), *Analiza wybranych koncepcji w obszarze projektowania wymagań*, „Studia Ekonomiczne. Zeszyty Naukowe Uniwersytetu Ekonomicznego w Katowicach”, nr 128, s. 137-162.
- [www 1] <http://www.inzynieriawiedzy.pl/ontologie/definicje> (dostęp: 04.03.2018).
- [www 2] <https://pdfs.semanticscholar.org/1c2f/786b0a8930835fe014f6aaa861df1ea4f4e4.pdf> (dostęp: 16.08.2016).
- [www 3] <http://homepages.cs.ncl.ac.uk/brian.randell/NATO/nato1968.PDF> (dostęp: 16.08.2016).

[www 4] <http://www.cxo.pl/felieton/Nowe-impulsy-w-ocenie-oplaczalnosci-projektow-z-obszaru-IT,321473,2.html> (dostęp: 16.08.2016).

[www 5] <http://docplayer.pl/1535388-Seminarium-dla-doktorantow-paradygmat-badan-naukowych-hevnera-i-in.html> (dostęp: 30.08.2017).

[www 6] <http://www.eclipse.org/epf/general/OpenUP.pdf> (dostęp: 16.08.2017).

A PROPOSAL OF THE PROCESS OF CREATING THE ONTOLOGY FOR REQUIREMENTS ENGINEERING

Summary: The success of business operations of an enterprise is more and more dependent on an effective use of IT solutions. Project management from the perspective of the Requirements Engineering method is a response of the software engineering. However, very often it is an unattainable challenge in the economic practice, particularly for small and medium enterprises. As the main limitation for SME are time and costs which are necessary on account of performing the work. The article suggests an innovative approach to solve this problem. The formulated process of creating the ontology use of requirements engineering, is to lead to easier, and most of all more effective, use of requirements engineering. For this purpose, it uses the achievements of ontology engineering, software engineering, as well as the IT and business balance management. In the article, a comprehensive methodological and tool method is proposed, based on free ontology and software engineering standards.

Keywords: ODSD, ODRE, UPON, OPENUP, IT BSC Model, UML.