



Politechnika Gdańska
Wydział Zarządzania i Ekonomii
Katedra Zastosowań Informatyki w Zarządzaniu
Zakład Zarządzania Technologiami Informatycznymi



ROZPRAWA DOKTORSKA

**ADAPTACYJNY AGENTOWY MODEL
ZARZĄDZANIA PROJEKTAMI INFORMATYCZNYMI**

Artur ZIÓŁKOWSKI

PROMOTOR:

dr hab. inż. Cezary ORŁOWSKI, prof. PG

Gdańsk 2012

SPIS TREŚCI

Wstęp	5
Rozdział I Zarządzanie projektami informatycznymi — stan badań	8
1.1 Projekt informatyczny środowiskiem procesów zarządczych i wytwórczych.....	8
1.2 Metody i narzędzia wspomagające procesy zarządzania projektami informatycznymi.....	12
1.2.1 Przykład metody klasycznej — Rational Unified Process	16
1.2.2 Przykład metody klasycznej — PRINCE2.....	21
1.2.3 Przykład metody lekkiej — SCRUM	26
1.2.4 Przykład metody lekkiej — EXTREME PROGRAMMING.....	30
1.2.5 Przykład zbioru dobrych praktyk — PMBoK	33
1.3 Podsumowanie — Stan badań projektów informatycznych	36
Rozdział II Identyfikacja zmiennych decyzyjnych w projektach informatycznych	38
2.1 Zmienne decyzyjne w projekcie informatycznym	38
2.1.1 Dojrzałość klienta w projekcie informatycznym	40
2.1.2 Dojrzałość dostawcy w projekcie informatycznym	45
2.1.3 Entropia projektu informatycznego	52
2.2 Stosowanie dobrych praktyk zarządzania projektami informatycznymi w kontekście zmiennych decyzyjnych	54
2.3 Podsumowanie	57
Rozdział III Założenia do budowy adaptacyjnego agentowego modelu zarządzania projektami informatycznymi	58
3.1 Koncepcja budowy modelu	58
3.2 Analiza dziedzinowa	62
3.3 Dobór metody modelowania.....	67
3.4 Budowa prototypu funkcjonalnego modelu agentowego — submodel MAS_FUNK	71
3.5 Implementacja prototypu funkcjonalnego — submodel MAS_WEB	76
3.6 Zastosowanie procesów przetwarzania wstępnego na potrzeby submodelu MAS_WEB	79
3.7 Analiza możliwości adaptacji i weryfikacja modelu	80
3.8 Podsumowanie	83
Rozdział IV Adaptacyjny agentowy model zarządzania projektem informatycznym	84
4.1 Specyfikacja agentowego modelu zarządzania projektami	84
4.1.1 Specyfikacja agentów funkcjonalnych.....	85
4.1.2 Specyfikacja agentów zarządzających	86
4.1.3 Specyfikacja agentów wiedzy	88

4.2	Agentowy model ogólny zarządzania projektami informatycznymi.....	90
4.3	Agentowy model szczegółowy zarządzania projektami informatycznymi	94
4.4	Podsumowanie	104
Rozdział V Badania zasadności replikatywnej modelu.....		105
5.1	Procesy badania zasadności replikatywnej modelu.....	105
5.1.1	Badanie 1-R. Zasadności definiowania struktur projektowych.....	106
5.1.2	Badanie 2-R. Ocena możliwości identyfikacji elementów struktury projektowej jako agentów projektu	110
5.1.3	Badanie 3-R. Weryfikacja warstwy funkcjonalności modelu na poziomie definiowania 113	
5.1.4	Badanie 4-R. Weryfikacja warstwy zarządzania modelu (możliwości współpracy agentów).....	116
5.1.5	Badanie 5-R. Weryfikacja warstwy zarządzania modelu (definiowania relacji).....	120
5.1.6	Badanie 6-R. Weryfikacja warstwy zarządzania modelu (zarządzanie relacjami).....	123
5.1.7	Badanie 7-R. Weryfikacja warstwy adaptacji modelu (dobór dobrych praktyk)	125
5.2	Podsumowanie — konstrukcja modelu po procesach badania zasadności replikatywnej....	129
Rozdział VI Badania zasadności predykcyjnej modelu		133
6.1	Procesy badania zasadności predykcyjnej modelu.....	133
6.1.1	Środowisko badań i eksperymenty badawcze	134
6.1.2	Eksperyment 1-P. Wsparcie procesów definiowania agentów (poziom definiowania) 135	
6.1.3	Eksperyment 2-P. Wpływ modelu na proces realizacji rzeczywistego projektu	139
6.1.4	Eksperyment 3-P. Implementacja poziomu definiowania modelu szczegółowego	142
6.1.5	Eksperyment 4-P. Implementacja relacji pomiędzy agentami oraz zarządzania przepływem komunikatów modelu szczegółowego	148
6.1.6	Eksperyment 5-P. Implementacja dobrych praktyk modelu szczegółowego a2M	153
6.2	Podsumowanie	159
Rozdział VII Podsumowanie i wnioski		160
Spis literatury.....		164
Spis rysunków		169
Spis tabel		170
ZAŁĄCZNIKI		171

Wstęp

Zarządzanie projektami stanowi jedno z podstawowych wyzwań współczesnej gospodarki. Dynamicznie zmieniające swe struktury organizacje stanowiące trzon tej gospodarki implikują zmiany zarówno w procesach jak i strukturach zarządzania. Elementem tych zmian jest odmienne od dotychczasowego podejście do wytwarzania nowoczesnych, szybko zmieniających się (wraz z wymaganiami rynku) produktów. W ramach tych zmian klasyczne struktury organizacyjne zastępowane są przez dedykowane zespoły projektowe.

Wraz z rosnącym znaczeniem projektów dla funkcjonowania organizacji, nabiera znaczenia problematyka zarządzania nimi. W szczególności problematyka ta jest widoczna w branży IT, w której złożone (trudne do wstępnego zdefiniowania), niepowtarzalne projekty obarczone wysokim ryzykiem wykonawczym są realizowane przez rozproszone (trudne do zarządzania) zespoły projektowe.

Prezentowane przez firmy konsultingowe dane wykazują, że ponad połowa projektów informatycznych jest nieudana z racji przekroczenia harmonogramu czy budżetu¹. Wskazuje się także, że jedną z podstawowych przyczyn nieudanych projektów stanowią błędy kierowników projektów. Błędy te są konsekwencją niewłaściwie realizowanych procesów zarządzania projektami, co w przypadku projektów informatycznych sprowadza się do braku stosowania lub niedopasowania metod zarządzania projektami informatycznymi do specyfiki danych projektów.

Wychodząc z założenia, że problematyka zarządzania projektami informatycznymi jest ważna dla procesów współczesnej gospodarki, zarządzania organizacjami oraz rynku IT, w niniejszej rozprawie poddano analizie i ocenie procesy projektów informatycznych. Celem tej oceny była próba diagnozy przyczyn nieudanych projektów oraz wskazania na rozwiązanie, które może usprawnić procesy zarządzania projektami informatycznymi.

Na podstawie tak przeprowadzonej diagnozy stanu projektów informatycznych wyodrębniono problemy badawcze, następnie postawiono główną hipotezę rozprawy oraz zdefiniowano cel pracy. Za podstawowy problem badawczy uznano rolę struktur projektowych i metod zarządzania projektami w procesach zarządzania projektem. Drugim problemem badawczym jest rola i znaczenie współczesnych technologii informatycznych do implementacji struktur i procesów zarządczych (agregowanych w metody zarządzania projektami).

Wyodrębnienie powyższych obszarów badawczych pozwoliło na postawienie głównej hipotezy niniejszej rozprawy weryfikowanej w części III, IV, V i VI tej pracy:

„Istnieje możliwość implementacji procesów zarządzania projektami do postaci agentowego adaptacyjnego modelu zarządzania projektami informatycznymi. Model ten stanowi alternatywne (w stosunku do istniejących metod) podejście do zarządzania projektami informatycznymi.”

¹ Z badań przeprowadzonych w 365 firmach na 3682 projektach przez firmę konsultingową Standish Group w 2007 r. wynikało, że tylko 431, czyli 12% z tych projektów zostało zakończonych na czas i nie przekroczyło budżetu.

Postawiono także dwie hipotezy pomocnicze poddane procesom weryfikacji w części I, II oraz V i VI tej pracy:

- „Istniejące metody zarządzania projektami i zagregowane dobre praktyki zarządcze nie są dopasowane do istniejącego trójkąta ograniczeń projektowych (zakres, harmonogram, budżet).”
- „Uzupełnienie istniejącego trójkąta ograniczeń o nowy (dojrzałość klienta, dojrzałość zespołu dostawcy i entropia projektu) usprawnia proces doboru metod i dobrych praktyk do zarządzania projektami.”

Na podstawie postawionej hipotezy głównej oraz dwu hipotez pomocniczych określono główny cel pracy oraz cele szczegółowe. Za główny cel pracy postawiono:

„opracowanie adaptacyjnego agentowego modelu zarządzania projektami”

Do osiągnięcia celu głównego określono trzy cele szczegółowe:

- dokonanie dekompozycji metod zarządzania projektami informatycznymi do postaci dobrych praktyki, które mogą być adaptowane przez model agentowy,
- wyodrębnienie zmiennych decyzyjnych, względem których można dobierać dobre praktyki zarządzania oraz określenie sposobu mierzenia tych zmiennych w ramach modelu,
- wykazanie możliwości doboru dobrych praktyk oraz definiowania struktur projektowych dla potrzeb projektów informatycznych.

W związku z realizacją celów niniejszej rozprawy pracę podzielono na siedem części (rozdziałów). W pierwszej części przedstawiono problematykę zarządzania projektami informatycznymi w odniesieniu do metod zarządzania projektami. Omówiono cechy charakterystyczne projektów informatycznych pod kątem struktur projektowych. Zaprezentowano również metody zarządzania projektami, wyodrębniając z nich dobre praktyki. Taka dekompozycja metod do postaci dobrych praktyk była konieczna z punktu widzenia zastosowanej w pracy metody badawczej.

W części drugiej pracy wyodrębniono zagregowane zmienne decyzyjne, stanowiące podstawę doboru dobrych praktyk zarządzania projektami informatycznymi. Wskazano na dojrzałość zespołu projektowego dostawcy, dojrzałość klienta projektu informatycznego oraz entropię projektu. Zaproponowano również sposoby pomiaru tych zmiennych. Zdefiniowano pojęcie entropii projektu jako nowe, dotychczas niewystępujące w informatyce.

W trzeciej części pracy wskazano na potrzebę opracowania modelu, który pozwoli na usprawnienie procesów zarządzania projektami. W tym celu zaprezentowano wykorzystaną w pracy metodę badawczą. W rozdziale przedstawiono etapy budowy modelu (według przyjętej metody badawczej), prezentując odpowiednie submodele stanowiące etapy pośrednie w budowie modelu adaptacyjnego.

W czwartym rozdziale zaprezentowano agentowy adaptacyjny model zarządzania projektami w postaci ogólnej oraz w postaci szczegółowej. Podczas omawiania modelu szczegółowego scharakteryzowano poziomy i warstwy modelu oraz zwrócono uwagę na jego adaptacyjny informacyjno-implementacyjny charakter.

Ostatnie dwa rozdziały poświęcone zostały weryfikacji modelu szczegółowego. W rozdziale piątym model poddano weryfikacji replikatywnej (diagnostycznej) w celu

sprawdzenia, na ile struktury, procesy i dobre praktyki zarządzania zakończonymi projektami mogą być implementowane w adaptacyjnym agentowym modelu zarządzania projektami. Proces weryfikacji głównej hipotezy badawczej został przeprowadzony w ramach siedmiu badań (1-R do 7-R).

W rozdziale szóstym opracowany model poddano weryfikacji predykcyjnej (prognostycznej), której celem było wykazanie, że w oparciu o opracowany adaptacyjny agentowy model zarządzania projektami jest możliwe stworzenie wersji implementacyjnej modelu i jego wykorzystanie w procesach zarządzania projektami informatycznymi. W tym celu przedstawiono wyniki pięciu eksperymentów badawczych (1-P do 5-P).

Rozprawę kończy podsumowanie, w którym zawarto wnioski i wskazówki dotyczące dalszych badań.

Autor składa serdeczne podziękowania Panu prof. Cezaremu Orłowskiemu za wsparcie merytoryczne i metodologiczne przy powstawaniu niniejszej rozprawy. Dziękuje także swoim kolegom z Zespołu za pomoc uzyskaną przy realizacji tej rozprawy: mgrów inż. Adamowi Czarneckiemu, drowi inż. Tomaszowi Sitkowi oraz mgrów inż. Pawłowi Madejowi. Ta praca powstała także dzięki ich wsparciu, życzliwości i ciągłym dyskusjom.

Rozdział I

Zarządzanie projektami informatycznymi — stan badań

Zarządzanie projektami informatycznymi stanowi jedno z najbardziej aktualnych wyzwań dla menedżerów w branży IT. Liczba realizowanych projektów informatycznych wzrasta w kolejnych latach. Wzrasta również zakres tych projektów oraz ich złożoność. Wraz ze wzrostem liczby projektów informatycznych narasta także liczba problemów decyzyjnych, przed jakimi stają prowadzący je menedżerowie. Analizy wykonane przez organizację Standish Group badającą od wielu lat projekty informatyczne wskazują na duży odsetek nieudanych projektów, czyli takich, które kończą się przekroczeniem budżetu, harmonogramu bądź niedotrzymaniem zakresu. Zjawisko niepowodzeń projektów informatycznych nazywane bywa również syndromem LOOP [98] (*Late* — opóźnienie, *Over Schedule* — przekroczony harmonogram, *Overtime* — nadgodziny, *Poor Quality* — niska jakość), czyli dostarczaniem produktu projektu po czasie, z przekroczonym budżetem oraz niską jakością. Główne problemy zarządzania projektami informatycznymi koncentrują się na niewłaściwym zarządzaniu zespołem, procesami wytwarzania oraz wykorzystywanymi w procesach wytwarzania i zarządzania technologiami informatycznymi [27,28,69,71]. Sposób zarządzania zespołem projektowym, procesami wytwórczymi oraz stosowanymi technologiami przekłada się na dobór i stopień wykorzystania metod i narzędzi informatycznych przez kierującego przedsięwzięciem [23,33,64]. Źle dopasowana do projektu informatycznego metoda zarządzania lub narzędzie informatyczne wspomagające procesy zarządcze i wytwórcze może prowadzić do opóźnień w realizacji zadań projektowych czy też demotywacji członków zespołu realizującego projekt [38,12].

W tym rozdziale omówione zostaną podstawowe zagadnienia związane z projektami informatycznymi w celu zaprezentowania środowiska, w jakim menedżerowie projektów podejmują decyzje. Celem tego rozdziału jest też przedstawienie kompendium wiedzy o projektach i procesach zarządzania nimi. Omówione zostaną również najbardziej popularne metody zarządzania projektami informatycznymi dla identyfikacji i dekompozycji dobrych praktyk zarządczych i wytwórczych, które kierownicy projektów powinni uwzględniać i dobierać do swoich projektów. Wyodrębnione z metod zarządzania projektami dobre praktyki stanowiąc będą podstawę do identyfikacji (w dalszej części pracy) zmiennych decyzyjnych w budowie proponowanego w pracy modelu agentowego zarządzania przedsięwzięciami informatycznymi.

1.1 Projekt informatyczny środowiskiem procesów zarządczych i wytwórczych

Na przestrzeni ostatnich lat obserwuje się wyraźny wzrost znaczenia prac projektowych w funkcjonowaniu organizacji. Projekty coraz częściej zastępują typowy, stanowiskowy sposób realizacji zadań. Pracownicy w organizacjach coraz częściej zatrudniani są „do projektu”, a nie „na stanowisku”. Samo zaś pojęcie „projektu” przeniknęło do wielu obszarów aktywności biznesowej. Zauważa się znaczący udział prac projektowych w działalności rynkowej [17,86]. Wydaje się zatem słuszne, aby dla uporządkowania rozważań dotyczących problematyki projektów i zarządzania nimi zaprezentować klasyfikacje projektów informatycznych, podstawowe definicje, następnie wskazać na specyfikę projektów informatycznych oraz procesów wytwórczych i zarządczych typowych dla projektów informatycznych.

W literaturze można spotkać szereg klasyfikacji projektów informatycznych. Zaprezentowana w tabeli 1.1 klasyfikacja projektów informatycznych i produktów wskazuje na złożoność zarówno projektów informatycznych, jak też i produktów będących efektem realizacji projektów informatycznych.

Tab. 1.1 Klasyfikacja projektów informatycznych wraz z możliwymi produktami

KRYTERIUM	RODZAJ PROJEKTU	PRZYKŁADY PRODUKTÓW
Cel	Software'owe	Prowadzą do wytworzenia określonego oprogramowania (np. systemu zarządzania zapasami w firmie, aplikacji bazodanowej do wspomaganie wystawiania faktur)
	Hardware'owe	Prowadzą do wytworzenia określonych elementów infrastruktury informatycznej organizacji (np. serwerowni, stanowisk roboczych w firmie).
Znaczenie praktyczne	Priorytetowe	Są niezbędne dla funkcjonowania organizacji, bez nich organizacja nie może funkcjonować (np. aplikacja do rozliczeń finansowo-księgowych).
	Prestiżowe	Pozwalają organizacji osiągać lepszą pozycję, lepszy status rynkowy (np. projekty prowadzące do stworzenia serwisów WWW).
Rozmiar	Duże	Projekty, w których liczebność zespołu przekracza 50 osób oraz budżet przekracza 10 mln EUR.
	Średnie	Projekty, w których skład zespołu mieści się w granicach od 6 do 50 osób, budżet całego projektu nie przekracza 10 mln EUR.
	Małe	Projekty w których skład zespołu nie przekracza 6 osób, budżet mieści się w pobliżu 0,1 mln EUR.
Stopień nowatorstwa	Innowacyjne	Projekty prowadzące do powstania nowych rozwiązań informatycznych, nie używanych dotychczas (np. oprogramowania telefonów komórkowych).
	Rozwojowe	Projekty prowadzące do rozszerzenia funkcjonalności dotychczas istniejącego produktu (np. rozbudowa serwisu WWW o system zarządzania treścią CMS).
	Odtworzeniowe lub Dostosowawcze	Projekty polegające właściwie na odtworzeniu lub dostosowaniu wcześniej wytworzonego produktu (np. aplikacja do obsługi sklepu X, gdy wytworzono uprzednio dla sklepu Y).
Pochodzenie zlecenia	Zewnętrzne	Oznacza projekt, gdzie klientem jest podmiot zewnętrzny zlecający i finansujący wykonanie określonego produktu informatycznego.
	Wewnętrzne	Coraz więcej projektów realizowanych przez firmy z branży IT dotyczy wytworzenia oprogramowania lub infrastruktury na własne potrzeby (np. system do zarządzania wersjami).

Źródło: opracowanie własne na podstawie [24] oraz [62]

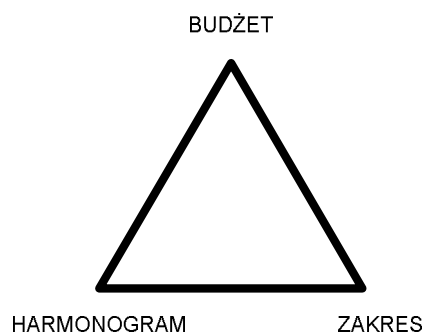
Definiując **projekt**, Griffin pisał, że „jest to sekwencja zadań mająca wyraźny początek i koniec określony w czasie, pod względem dostępnych zasobów i oczekiwanych rezultatów. Jest on wydzielony z toku codziennej, rutynowej pracy, jest realizowany równocześnie z nią lub z całkowitym oddelegowaniem członków zespołu do realizacji projektu. Po jego zakończeniu zawsze pozostaje coś nowego, co jest wprowadzone do dotychczasowej organizacji w celu jej usprawnienia”.[87] Kolejna definicja, nieco bardziej zwarta, mówi, że jest to „ograniczone w czasie działanie, podjęte w celu wytworzenia unikalnego produktu, usługi lub innego wytworu” [71]. O projekcie mówi się również, że „jest to specyficzne, nowe działanie, które w sposób metodyczny i progresywny tworzy strukturę przyszej rzeczywistości, dla której nie ma jeszcze dokładnego odpowiednika” [68].

Jak wynika z zaprezentowanych definicji, mimo pewnych składniowych różnic, łączy je wiele cech wspólnych. Przedstawione definicje traktują bowiem projekt bardzo podobnie, wskazując zawsze na zestaw działań prowadzących do określonego celu. Z powyższych definicji płyną też bardzo wyraźne konkluzje. Projekt bowiem odnosi się nie tylko do wymiaru technicznego, ale również mianem projektu coraz częściej przyjęło nazywać aktywność pracowników na rzecz zdefiniowanego celu. Projekt łączy bowiem ludzi z wielu działów, gromadzi ich przy prowadzeniu prac w ramach jednego wspólnego celu.

Drugą ważną konkluzją jest to, że z pojęciem projektu niemal nieodłączne kojarzony jest proces zarządzania nim. Gdyby pojęcie projektu odnieść wyłącznie do obszaru technicznego (np. projekt budynku), to trudno byłoby mówić o zarządzaniu. Zarządzanie bowiem związane jest zawsze z czynnikiem ludzkim. Mówiąc zatem o zarządzaniu projektami, należy mieć na myśli zarządzanie zespołem, relacjami z klientem i innymi aspektami związanymi z ludźmi. Takie rozumowanie pozwala zatem traktować projekt równorzędnie z innym słowem, jakiego używa się w języku polskim na określenie ukierunkowanych, celowych działań — „przedsięwzięcie”. Można niekiedy spotkać się ze sporem toczonym w celu określenia, czy przedsięwzięcie jest pojęciem szerszym niż projekt. Przyjmując jednak definicje projektu przedstawione wyżej, należy rozumieć projekt tożsamo z przedsięwzięciem. **Zarządzanie projektem** należy rozumieć natomiast jako sprawowanie podstawowych funkcji zarządzania (planowania, organizowania, motywowania, kontrolowania w ramach realizowanego projektu. [80].

Jedną z najbardziej ukierunkowanych branż, które od samego początku opierały swoją działalność na realizacji projektów, jest branża IT. Wszystkie bowiem produkty (sprzęt, oprogramowanie) powstałe na przestrzeni ostatnich kilkudziesięciu lat w tej branży są efektem określonych projektów[58]. Poniżej scharakteryzowano specyfikę projektów informatycznych, prezentując metody zarządzania projektami informatycznymi celem identyfikacji występujących tam problemów, jak też dobrych praktyk i zmiennych decyzyjnych wykorzystywanych przez kierujących projektami w procesach zarządzania nimi.

Po przedstawieniu definicji projektu oraz procesu zarządzania nim przystąpiono do określenia specyfiki projektów informatycznych, prezentując ich cechy. Stąd też dalsze rozważania skoncentrowane są na przedstawieniu zarówno cech wspólnych dla wszystkich projektów (w tym projektu informatycznego) oraz na wykazaniu immanentnych cech projektu informatycznego (czyli tylko dla niego właściwych). Takie cechy jak złożoność projektu, jego mała powtarzalność, a także wysokie ryzyko realizacji wynikające ze zmiennego zapotrzebowania na określone zasoby (ludzkie, finansowe), trudne do określenia ramy czasowe realizacji czy trudności w zdefiniowanego zakresu prac potrzebnych do wytworzenia systemu informatycznego, są typowymi cechami projektu informatycznego. Cechy te ujmuje się w klasyczny trójkąt ograniczeń



Rys. 1.1 Klasyczny trójkąt ograniczeń

Źródło: [59]

Z rysunku 1.1 wynika, że członkowie zespołów projektowych działają w ramach ograniczeń zasobów oraz w wyznaczonych ramach pracy (zakres), zaś osoby kierujące takim projektem (nazywani kierownikami projektu lub menedżerami projektu) podejmują szereg decyzji w ramach tych ograniczeń. Decyzje te są niezwykle istotne, ale też trudne z punktu widzenia realizacji całego projektu, gdyż zmiana któregokolwiek z klasycznych ograniczeń powoduje najczęściej zmianę w pozostałych [60]. W przypadku projektów informatycznych procesy zmian są jedynymi pewnymi procesami projektowymi, a umiejętność zarządzania procesami zmian minimalizuje ryzyko realizacji projektu [9,12]. Zwiększenie zakresu projektu informatycznego będzie bowiem pociągać za sobą zmiany w harmonogramie, co najczęściej pociąga za sobą potrzebę wzrostu kosztów projektu. W przypadku projektów informatycznych zmiany zakresu i harmonogramu często mają miejsce, zmiany kosztów są stosunkowo rzadkie.

Kolejną typową cechą projektu jest przypisanie do projektu osoby, która będzie nim zarządzać. Zatem z projektem nieodłączna jest rola lidera, nazywanego najczęściej kierownikiem projektu. W zależności od specyfiki projektu (w tym projektu informatycznego), takiemu kierownikowi powierza się najczęściej określony zakres odpowiedzialności związany z podstawowymi funkcjami zarządzania. Stąd też kierownik projektu ma za zadanie organizować zasoby (np. tworzyć zespół), planować działania (np. definiować zadania), kontrolować postępy projektu (np. odbierać raporty) oraz motywować pracowników do działania (np. opracowywać system motywacji dla uczestników projektu). To również kierownik projektu podejmuje główne decyzje odnośnie sposobu realizacji projektu. W zależności od rozmiarów projektu, procesy zarządzania takim projektem mogą być bardziej złożone (realizowane przez grupę osób zarządzających projektem) albo ograniczone (np. kierownik jest jednocześnie wykonawcą). W przypadku projektów informatycznych kompetencje kierownika w dużej mierze są konsekwencją procesów zarządzania ujętych w metodach zarządzania projektami informatycznymi. Metody te oraz procesy zarządcze grupowane w dobre praktyki zarządcze zostaną omówione w dalszej części niniejszego rozdziału.

Procesy wytwórcze projektu są podzielone na etapy stanowiące charakterystyczne punkty kontrolne (ang. *milestones*) z dużym nastawieniem na częstą weryfikację procesów wytwórczych dzięki zastosowaniu podejścia iteracyjnego do wytworzenia produktu [27]. Nie zakłada się w procesach realizacji projektu informatycznego następstw czasowych (gromadzenie wymagań, programowanie, akceptacja klienta). Takie założenia dotyczyły podstawowego sposobu wytwarzania, jakim był cykl kaskadowy (zaprezentowany w dalszej części). W przypadku np. projektu budowlanego zmiana wymagań jest właściwie niemożliwa, gdyż prace przebiegają według przygotowanego planu architektonicznego. W przypadku wytwarzania oprogramowania wydaje się to o wiele łatwiejsze, stąd też dopuszcza się możliwość zmiany wymagań przez klienta. Inną przyczyną dopuszczania zmienności wymagań w projektach informatycznych jest fakt, że często klient zamawiający określony produkt tworzony w projekcie informatycznym nie jest w stanie dokładnie przedstawić swoich oczekiwań i zmienia je w trakcie realizacji projektu.

Z podejściem ciągłych zmian wymagań przez klienta wiąże się ryzyko realizacji projektu. Jest ono szczególnie wysokie w przypadku ciągłych i trudnych do przewidzenia, a tym samym do zaakceptowania zmian. Proces analizy ryzyka ma szczególne znaczenie w projektach informatycznych z uwagi na ich złożoność [86,87]. Stąd też oszacowanie ryzyka dla projektu informatycznego jest niezwykle trudne i najczęściej wyrażane w wartościach lingwistycznych (np. małe, średnie, duże) [42,88]. Ryzyko to można sklasyfikować w czterech obszarach omówionych w tabeli Tab. 1.2:

Tab. 1.2 Klasyfikacja ryzyka w projektach informatycznych

RODZAJ RYZYKA	ZNACZENIE I ODPOWIEDZIALNOŚĆ
Organizacyjne	Oznacza możliwość popełnienia błędów podczas organizacji zasobów (np. za mało osób do realizacji zadań). Odpowiedzialność ponosi osoba zarządzająca projektem.
Technologiczne	Oznacza możliwość niedostosowania używanych technologii do oczekiwań klienta, niewłaściwe zaplanowanie niezbędnych środków technicznych (sprzętu i oprogramowania) do realizacji projektu. Odpowiedzialność ponosi analityk biznesowy.
Społeczne	Oznacza możliwość wystąpienia problemów personalnych, takich jak demotywacja zespołu, niechęć do zmian, złe uporządkowanie relacji między członkami zespołu. Odpowiedzialność za te działania ponosi kierownik projektu.
Biznesowe	Ryzyko związane ze zmiennością rynku (kursy walut, bankructwo klienta, pojawienie się konkurencji). Odpowiedzialność za kontrolę zagrożeń biznesowych ponosi osoba zarządzająca projektem.

Źródło opracowanie własne na [9,70,65]

Reasumując: projekt oraz jego procesy wytwórcze i zarządcze stanowią przykład wielowymiarowego systemu społeczno-technicznego, w ramach którego zespoły ludzkie wytwarzają złożone produkty. Procesy projektowe (wytwórcze i zarządcze) prowadzące do wytworzenia produktu obarczone są wysokim poziomem ryzyka z uwagi na zmieniające się uwarunkowania projektowe (zmiana wymagań czy też zmiana środowiska wytwarzania). Kierownik projektu, zarządzając projektem (wielowymiarowym systemem społeczno-technicznym), dysponuje specjalistycznym aparatem zarządczym, jakim są metody i technologie informatyczne wspierające procesy projektowe. Na ile ten specjalistyczny aparat jest adekwatny do złożonych procesów projektów i na ile jest on także adekwatny do wiedzy i doświadczenia kierowników projektów. Przedstawione we wstępie dane dotyczące realizacji projektów wskazują na znaczny odsetek tych nieudanych. Rodzi się więc pytanie, w czym tkwi problem realizacji projektów informatycznych? Jak zarządzać projektem, aby prezentowany trójkąt ograniczeń był trójkątem wiedzy, a nie trudnych do identyfikacji wierzchołków? Dlatego też w kolejnym podrozdziale przedstawiono metody i narzędzia zarządzania projektami, szukając w nich odpowiedzi na postawione powyżej pytania. Prezentacja metod zarządzania ma też na celu poszukiwanie kluczowych zmiennych decyzyjnych dla kierowników projektów, tak aby podejmowane przez nich decyzje odnosiły się do najważniejszych dla projektu problemów.

1.2 Metody i narzędzia wspomagające procesy zarządzania projektami informatycznymi

Poniżej zostaną przedstawione metody zarządzania projektami informatycznymi z podziałem na metody klasyczne i metody lekkie. Celem prezentacji tych metod jest zarówno próba odpowiedzi na postawione w uprzednim podrozdziale pytania, jak też potrzeba dekompozycji metod do poziomu dobrych praktyk tak istotnych dla identyfikacji zagregowanych zmiennych decyzyjnych leżących w gestii kierownika projektu. Identyfikacja zmiennych decyzyjnych stwarza możliwość zarówno oznaczenia problemów zarządzania projektem, jak i może stanowić szansę na identyfikację podstawowych zmiennych, których wartości decydują o stanie zagregowanych zmiennych decyzyjnych.

Metody zarządzania projektami informatycznymi stanowią zbiory doświadczeń zarządzania projektem informatycznym i organizowania prac z nim związanych [23,71]. Metody te składają się

z zestawu tzw. dobrych praktyk (ang. *best practices*). Dobre praktyki stanowią wskazówki i zalecenia dotyczące procesów zarządzania projektem informatycznym jak również działaniami operacyjnymi (procesami wytwarzania). Dobre praktyki zawarte w metodach zarządzania projektami odnoszą się również do tego, w jaki sposób organizować przepływy pracy w projekcie, jak konstruować zespół, jak również do tego, jak sporządzać dokumentację projektową.

Obecnie metody zarządzania projektami dzieli się zasadniczo na dwie grupy: metody klasyczne oraz metody lekkie [15]. Głównym kryterium tej klasyfikacji jest przede wszystkim udział procesów planowania w projekcie informatycznym. Metody ciężkie charakteryzują się bardzo wyraźnym podejściem procesowym, metody lekkie stawiają na minimalizację procesów planowania na rzecz aktywnej pracy całego zespołu [34].

Warto zauważyć, że obecnie funkcjonuje kilka wiodących metod zarządzania projektami informatycznymi oraz zbiorów dobrych praktyk zarządzania takimi projektami. Do wiodących metod zarządzania projektami informatycznymi należą takie metody jak Rational Unified Process (RUP), Project in Controlled Environment 2 (PRINCE2), SCRUM czy EXTREME PROGRAMMING (XP). Podstawowym zbiorem dobrych praktyk zarządzania jest natomiast Project Management Body of Knowledge (PMBok), stanowiący opracowanie zawierające generyczne (nie związane z konkretną dziedziną działalności) dobre praktyki zarządzania. Należy nadmienić również, że metody zarządzania projektami mają charakter dynamiczny, tzn. że koncepcje w nich zawarte ulegają z czasem rozwojowi. Na podstawie wspomnianych metod zarządzania projektami powstaje więc wiele ich odmian. Niekiedy korporacje tworzą w ten sposób własne metody zarządzania projektami, które stosują wyłącznie do swoich potrzeb [84].

Można zatem zauważyć, że zarządzanie projektami informatycznymi jest problemem złożonym i stanowi bardzo poważne wyzwanie dla kierowników. Stąd uznaje się za właściwie niemożliwe prowadzenie projektu informatycznego bez stosowania dobrych praktyk zarządzania projektami zawartymi w formalnych metodach zarządzania projektami. Jednakże duża liczba metod funkcjonujących obecnie w branży IT, a co za tym idzie, ogrom dobrych praktyk zarządzania projektami informatycznymi, powoduje, że kierownicy projektów stają zawsze przed dylematem, jakiej metody użyć do swoich projektów albo które dobre praktyki dobrać do danego projektu informatycznego. Dobór metody do projektu nie może być bowiem decyzją przypadkową, ale popartą wcześniejszą analizą realiów projektowych, czyli zdefiniowanej uprzednio struktury projektu.

Warto również nadmienić, że kierownicy projektów informatycznych wybierający metodę zarządzania projektami nie korzystają ze wszystkich dobrych praktyk w niej zawartych. Niektóre bowiem metody zawierają tyle zaleceń, że ich stosowanie dla jednego projektu jest właściwie niewykonalne (np. PMBoK). Kierownicy powinni zatem dobrać określone dobre praktyki wyłącznie pod kątem struktury projektowej danego projektu. Rodzi się jednak pytanie, czy powinni stosować dobre praktyki związane wyłącznie z jedną metodą zarządzania projektem. Czy nie lepszym rozwiązaniem jest jednak łączenie dobrych praktyk pochodzących z różnych metod zarządzania projektami informatycznymi?

Powyższe rozważanie wskazuje, że dobrym praktykom zawartym w metodach zarządzania projektami brakuje pewnego kontekstu (środowiska) ich zastosowania. Dobre praktyki są najczęściej bardzo szeroko opisane, jednak bardzo często nie mówi się, kiedy je stosować — do jakich projektów, do jakiego zespołu czy do jakiego klienta. Nie uwzględnia się zatem struktury projektowej w stosowaniu dobrych praktyk zarządzania projektami informatycznymi.

Należy zaznaczyć również, że dobre praktyki zarządzania projektami mogą być sprzeczne w zależności od tego, czy pochodzą z metod klasycznych czy lekkich. To również aspekt, który powinni uwzględniać kierownicy projektów. Stąd też poniżej zaprezentowano zbiór wspólnych dobrych praktyk dla metod klasycznych oraz wspólnych dobrych praktyk dla metod lekkich.

- **METODY KLASYCZNE ZARZĄDZANIA PROJEKTAMI**

Metody klasyczne, zwane również „ciężkimi”, charakteryzują się znacznym udziałem procesów planowania w zarządzaniu projektem informatycznym. Według zasad zawartych w metodach klasycznych [77] oczekuje się najczęściej od kierowników projektów tworzenia planów na każdym etapie projektu. Zaleca się także bardzo dokładne planowanie zasobów (zespół, środki finansowe), jak również oczekuje się bardzo precyzyjnej kontroli poprzez rozwinięcie narzędzi monitoringu i sprawozdawczości (np. częste raportowanie). W metodach klasycznych dominuje również duży poziom standaryzacji pracy. Określa się bowiem bardzo precyzyjnie role projektowe, zadania, a także przepływy zadań (nazywane także przepływami pracy) i oczekiwane produkty poszczególnych etapów (albo nawet produkty zadań). W metodach klasycznych znaleźć można szereg dobrych praktyk związanych zarówno z zadaniami dla kierownika projektu informatycznego, jak również dla pozostałych członków zespołu realizującego projekt (dotyczących wytwarzania oprogramowania, testowania czy wdrażania). Ponadto kładzie się duży nacisk na formalne przepływy informacji w postaci notatek służbowych czy sprawozdań.

Metody ciężkie kojarzone są najczęściej z dużymi projektami i dużymi zespołami projektowymi. Mogą być co prawda stosowane w projektach małych i średnich, ale ze względu na duży udział formalizmów mogą się okazać mało wydajne dla mniejszych projektów.

Podstawowe dobre praktyki, wspólne dla wszystkich klasycznych metod zarządzania projektami, zestawiono w tabeli poniżej.

Tab. 1.3 Podstawowe dobre praktyki zawarte w metodach klasycznych

KATEGORIE DOBRZYCH PRAKTYK	OBJAŚNIENIE
Ustalone plany	W metodach klasycznych zakłada się, że powodzenie projektu osiągnięte zostanie na skutek dokładnego planowania wszystkich niezbędnych elementów struktury projektu, począwszy od planowania zespołu, poprzez dokładne definiowanie zadań i przepływów tych zadań w ramach procesu.
Zadania i procesy	Metody klasyczne sugerują, aby każde zadanie projektowe zostało bardzo dokładnie zdefiniowane, miało przypisaną odpowiedzialność jak również parametry wejściowe i wyjściowe zadania i/lub procesu.
Planowanie i harmonogram	Metody klasyczne traktują, że jednym z najważniejszych procesów jest planowanie zadań i budowanie harmonogramów ich realizacji.
Niska niepewność i ryzyko	Zakłada się, że metody klasyczne sprzyjają projektom, w których ryzyko jest nieduże i niepewność (możliwość zmiany wymagań) właściwie niespotykana.
Średnie lub niewielkie zmiany	Dopuszcza się w metodach klasycznych wyłącznie drobne, niewielkie zmiany. Z uwagi na z góry ustalone plany, każda radykalna zmiana skutkowałaby tworzeniem wszystkich planów od nowa.
Lider podejmuje większość decyzji	W metodach klasycznych podejmowanie decyzji jest scentralizowane i najczęściej powierzone kierownikowi projektu lub odpowiedniemu naczelnemu organowi zarządzającemu.

Źródło: opracowanie własne na podstawie [9,70]

Dziś do najczęściej stosowanych metod klasycznych zarządzania projektami informatycznymi należą Rational Unified Process (RUP) oraz Project in Controlled Environment (PRINCE2)

- **METODY LEKKIE ZARZĄDZANIA PROJEKTAMI**

Metody lekkie powstały w pewnej opozycji do metod klasycznych, przeciwstawiając nadmiarowości procesów planowania nastawienie na działanie i możliwie dużą aktywizację członków

zespołu. Stąd też metody lekkie nazywa się także zwinnymi, żwawymi (od ang. *agile*). Metody lekkie charakteryzuje duży nacisk na pracę zespołową. Zmniejsza się nieco rolę kierownika projektu, traktując go najczęściej jako równorzędnego członka zespołu, który prowadzi prace kierownicze, ale także operacyjne. Odpowiedzialność zaś za prace prowadzone w projektach informatycznych realizowanych według metod lekkich scedowana jest na samoorganizujący się zespół, którego zadaniem jest właściwe określenie zadań i podzielenie pracy między poszczególnych członków. Należy zaznaczyć, że podstawą do rozwoju metod lekkich był tzw. Manifest Zwinności (ang. *Agile Programming Manifesto*) [24]. W dokumencie tym zawarto kilkanaście postulatów zachęcających do odejścia od koncepcji zarządzania procesowego w projektach informatycznych na rzecz podporządkowania projektów oczekiwaniom klienta. W manifestie tym stwierdzono, że zmienność wymagań jest tak typowa dla projektów informatycznych, że nie należy jej minimalizować, ale traktować jako normalną sytuację. Stąd też zadaniem zespołów realizujących projekt informatyczny powinno być umiejętne dostosowywanie się do częstych zmian. W Manifestie Zwinności mówi się także o konieczności włączenia klienta w prace zespołu oraz o budowaniu więzi (relacji) ze stroną biznesową projektu. Poprzez włączenie klienta do zespołu projektowego zakłada się także uzyskanie możliwości bieżącej weryfikacji tworzonego produktu informatycznego (głównie oprogramowania). W Manifestie Zwinności wspomniano, że zarządzając projektem informatycznym należy stosować możliwie krótkie cykle wytwarzania. Dzięki temu weryfikacja produktu wytwarzanego w projekcie informatycznym będzie częstsza, a tym samym ostateczny produkt będzie bardzo zbliżony do oczekiwań klienta. Wreszcie założono, że ważniejsze z punktu widzenia realizacji projektu informatycznego jest skoncentrowanie się na poprawnych przepływach informacji w zespole niż planowanie poszczególnych etapów realizacji projektu. Postawiono tym samym na zespoły samoorganizujące się, bez wyraźnych liderów, za to bardzo blisko ze sobą współpracujące i dzielące się wiedzą podczas częstych spotkań porządkowych.

Główne dobre praktyki zarządzania projektami według metod lekkich zgromadzono w tabeli poniżej:

Tab. 1.4 Podstawowe dobre praktyki w metodach lekkich

KATEGORIE DOBRZYCH PRAKTYK	OBJAŚNIENIE
Płynne plany	W metodach lekkich zakłada się, że planowanie powinno być krótkookresowe i dotyczyć głównie najbliższego etapu.
Wyniki	Zespoły realizujące projekt zgodnie z podejściem lekkim powinny nastawić się na osiąganie konkretnych wyników, które powinny być możliwie często prezentowane klientowi.
Interakcje i zarządzanie wiedzą	O wiele ważniejsze w projekcie od planowania powinno być budowanie relacji między uczestnikami, zwiększanie przepływów informacji i dzielenie się wiedzą.
Wysoka niepewność i ryzyko	Metody lekkie, ze względu na swój charakter rozwojowy (brak sztywnych planów), mogą dobrze adaptować się do projektów obciążonych dużym ryzykiem.
Duże zmiany	Zakłada się, że zmiany w projektach realizowanych metodami lekkimi mogą pojawiać się w dowolnym momencie,
Zespołowe podejmowanie decyzji	Podejmowanie decyzji jest raczej zdecentralizowane. Zakłada się, że wszystkie decyzje zapadają na forum zespołu, zespół odpowiada za podjęte decyzje.

Źródło: opracowanie własne na podstawie [96, 32, 74]

Powyższa analiza pozwoliła na wskazanie cech wspólnych dla metod klasycznych oraz cech wspólnych dla metod lekkich zarządzania projektami. Oprócz jednak tych wspólnych cech każda metoda zarządzania projektami informatycznymi cechuje się zestawem własnych dobrych praktyk.

Niektóre dobre praktyki zawarte w metodach są uszczegółowieniem tych nadrzędnych, zestawionych w tabelach 1.3 i 1.4. Wydaje się zatem słuszne, aby pogłębić analizę metod zarządzania projektami informatycznymi poprzez przyjrzenie się dobrym praktykom zawartym w najpopularniejszych metodach zarządzania w tej dziedzinie. Do analizy wybrano dwie najpopularniejsze metody klasyczne (RUP oraz PRINCE2) oraz dwie metody lekkie (SCRUM i XP). Analizie poddano również PMBoK jako źródło dobrych praktyk dla kierowników projektów. Celem analizy wymienionych metod będzie wyodrębnienie dobrych praktyk zawartych w tych metodach oraz pokazanie kontekstu ich zastosowania pozwalającego kierownikom projektów informatycznych dobierać te dobre praktyki do struktur projektowych swoich projektów. Takie wyodrębnienie jest niezbędne również z punktu widzenia wskazania zagregowanych zmiennych decyzyjnych i możliwości określania ich wartości.

1.2.1 Przykład metody klasycznej — Rational Unified Process

Rational Unified Process (RUP) to metoda zarządzania projektami informatycznymi klasyfikowana w grupie metod klasycznych (ciężkich), czyli takich, w których udział procesów planowania jest znaczny. Metoda powstała na bazie doświadczeń menedżerów projektów informatycznych zgromadzonych wokół środowiska wytwarzania Eclipse [9,73]. Nadrzędnym zadaniem metody RUP jest określenie sposobu realizowania projektów informatycznych oraz stworzenie ram pracy dla kierowników tych projektów. Stąd też głównym adresatem zaleceń zawartych w metodzie RUP są menedżerowie. Jednak metoda RUP może stanowić podstawę wiedzy o projektach informatycznych również dla pozostałych członków zespołów projektowych. Zawiera bowiem szereg wytycznych dla analityków biznesowych, programistów czy testerów.

Jak zostało zaznaczone na wstępie, RUP jest zbiorem dobrych praktyk, które powstały na bazie doświadczeń menedżerów projektów. Dobre praktyki zawarte w metodzie RUP powinny być wykorzystywane przez kierowników projektów w zależności od specyfiki projektu, którym kierują. Dlatego też wydaje się istotne, by wyodrębnić z dokumentacji tej metody tylko te dobre praktyki, które są niezbędne z punktu widzenia zarządzania projektem informatycznym i do których powinni stosować się kierownicy takich projektów.

RUP można zasadniczo podzielić na dwa wymiary zarządcze, z których każdy zawiera określone dobre praktyki prowadzenia projektu. Pierwszy wymiar dotyczy działań strategicznych oraz taktycznych i należy zaliczyć do niego zestaw dobrych praktyk w ramach „kluczowych wytycznych biznesowo zorientowanego wytwarzania” oraz w ramach planowania procesu realizacji projektu. Drugi wymiar jest odpowiednikiem działań planistycznych dla poziomu operacyjnego, czyli jak wykonywać poszczególne zadania projektowe, jak organizować zespół oraz jak zarządzać dokumentacją projektową.

Dobre praktyki – Wymiar 1

Główne wytyczne dla menedżerów związane ze strategią prowadzenia projektów zostały zawarte w zbiorze podstawowych wskazówek dla biznesowo zorientowanego wytwarzania (realizacji projektów informatycznych). Zasada ta (z ang. *Key principles for business-driven development*) składa się z sześciu porad dla menedżerów, z których każda zaczyna się kolejną literą alfabetu. W związku z tym przyjęło się nazywać tę zasadę zasadą ABCDEF.

“A” (Adapt the process) — Adaptuj procesy

Pierwsza wskazówka dla kierowników projektów zarządzających projektami zgodnie z metodą RUP dotyczy zastosowania podejścia procesowego w zarządzaniu. Podejście procesowe oznacza konieczność zdefiniowania przepływów zadań (pracy) w projekcie od jego początku do końca trwania.

Dodatkowo ta dobra praktyka zakłada, że menedżerowie planujący proces realizacji projektu informatycznego powinni go adaptować, czyli dostosowywać do specyfiki projektu (zdefiniowanej wcześniej jako struktura projektu). Adaptacja ta w rozumieniu metody RUP oznacza wybór właściwych zadań i przypisanie odpowiednich ról do realizacji projektu.

„B” (Balance competing stakeholder priorities) — Bilansuj (równoważ) oczekiwania uczestników

Druga z dobrych praktyk związanych z taktycznymi aspektami zarządczymi w projektach informatycznych dotyczy równoważenia oczekiwań wszystkich podmiotów zainteresowanych realizacją projektu. Zaleca się kierownikom projektów określenie i równoważenie oczekiwań wszystkich tzw. interesariuszy (czyli osób zainteresowanych efektami projektu). Do interesariuszy zaliczani są zatem członkowie zespołu, ale również kierownictwo firmy realizującej projekt (kierownictwo dostawcy), jak również kierownictwo odbiorcy projektu (czyli klienta). Według tej dobrej praktyki tylko wtedy możliwe jest właściwe organizowanie projektu, gdy kierownik zdaje sobie sprawę z oczekiwań, jakie stawiają sobie co do projektu wszystkie osoby zainteresowane rezultatem projektu (czyli interesariusze).

„C” (Colaborate across teams) — współpraca zespołów

Ta dobra praktyka dotyczy zapewnienia właściwej współpracy i koordynacji działań zespołów realizujących projekt informatyczny. Ponieważ metoda RUP zasadniczo powinna być stosowana do dużych projektów, rolą kierownika projektu jest zapewnienie właściwej współpracy między zespołami realizującymi zadania. Odnosząc to do jednego zespołu, zadaniem kierownika jest zapewnienie właściwego przepływu informacji i zadań pomiędzy członkami jednego zespołu. Głównym zatem przesłaniem tej dobrej praktyki jest zapewnienie stałego przepływu informacji wewnątrz projektu i koordynacja przepływów zadań (przepływów pracy) podczas realizacji projektu informatycznego.

„D” (Demonstrate value iteratively) — pokazuj wartość iteracyjnie

Ta dobra praktyka — wbrew niektórym opiniom, że podejście RUP stanowi implementację modelu kaskadowego — wskazuje wyraźnie, że nawet w dużych projektach należy stosować podejście iteracyjne do wytwarzania oprogramowania. Oznacza to, że końcowy produkt powstaje w oparciu o kilka lub kilkanaście cykli wytwarzania zwanych iteracjami. Każda iteracja prowadzi do pokazania działającej części wytwarzanego oprogramowania, która to część powinna zostać przetestowana i, po akceptacji przez klienta, przechodzi się do dalszego rozbudowywania tego oprogramowania. RUP zawiera również wskazówki co do tego, ile iteracji powinno zrealizować się w poszczególnych etapach projektu.

„E” (Elevate level of abstraction) — podnoś poziom abstrakcji

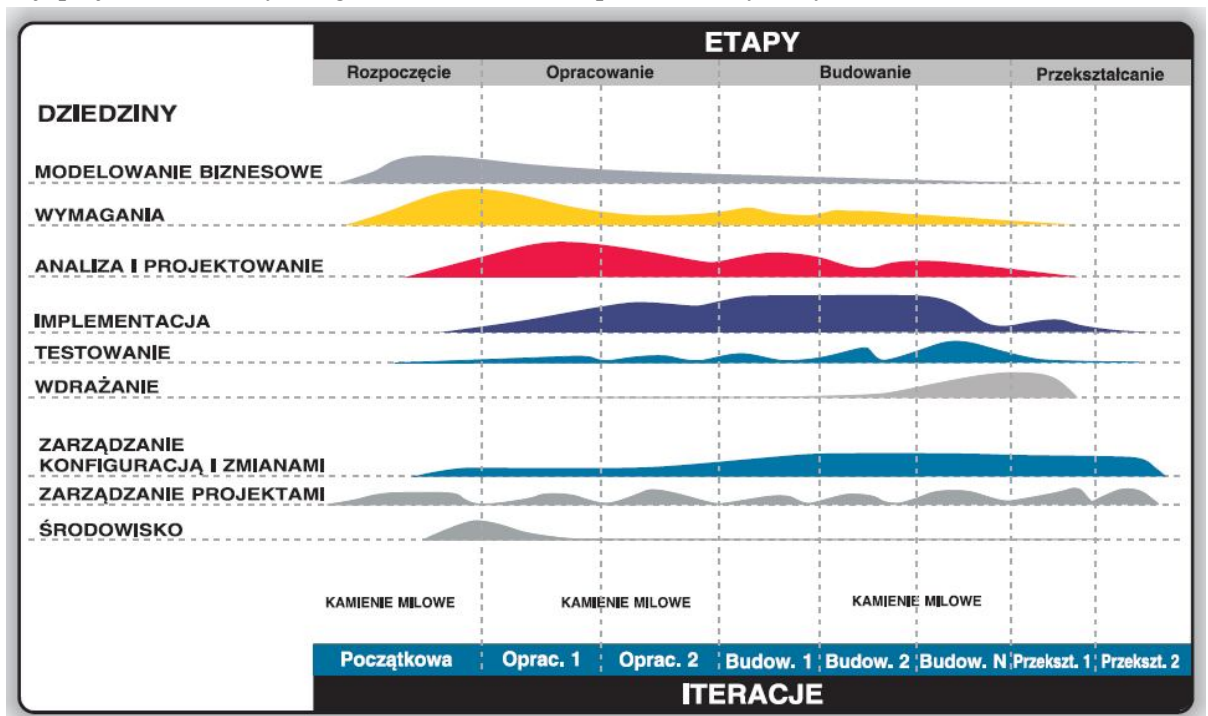
Ta dobra praktyka, odnosząc się do poziomu abstrakcji, wskazuje menedżerom, jak prowadzić prace analityczne w projektach informatycznych. W tej dobrej praktyce zaleca się bowiem stosowanie podejścia modelowego podczas rozmów z klientem i gromadzenia wymagań. Zaleca się zatem stosowanie powszechnie znanych języków modelowania (RUP posiada wbudowany przewodnik po notacji *Unified Modelling Language* — UML) oraz możliwie wysokopoziomowe prezentowanie klientowi całego systemu (np. w postaci architektury, modelu przypadków użycia itp.) Zasadniczo w tej dobrej praktyce zaleca się kierownikom projektów prezentowanie najważniejszych elementów projektu na jak najwyższym poziomie abstrakcji (diagramy UML, schematy blokowe, mapy myślowe). W takim przypadku klient nie musi znać kodu programowego, tym bardziej że prezentacja kodu będzie najprawdopodobniej dla klienta niezrozumiała. Dlatego też prezentacja w postaci diagramów, schematów czy tak popularnych map myślowych zdecydowanie ułatwi rozumienie projektu informatycznego i prac z tym projektem związanych.

„F” (Focus continuously on quality) — koncentruj się na jakości

Ostatnia z dobrych praktyk związanych z biznesowymi aspektami projektu dotyczy stałej koncentracji na jakości produktu i wpisuje się w dzisiejszy nurt zarządzania jakością. Zakłada się w niej, że wytwarzając określony produkt zawsze poszukuje się możliwości usprawniania i doskonalenia go.

Dobre praktyki — Wymiar 2: Etapy

Kolejnym zestawem dobrych praktyk RUP, które mogą zostać wykorzystane do identyfikacji zagregowanych zmiennych decyzyjnych, są te związane z komponowaniem (planowaniem) procesu realizacji projektu informatycznego. Ich układ został zaprezentowany na rys. 1.3



Rys. 1.2 Schemat RUP

Źródło: [9]

Według RUP zaleca się planowanie zadań w ramach wyodrębnionych etapów oraz w ramach poszczególnych dziedzin projektu (rys. 1.2).

Etapy projektu w RUP

Pierwsza dobra praktyka związana jest z wyodrębnieniem w projekcie podstawowych faz, w ramach których kierownicy projektu powinni planować zadania projektowe. Nasilenie określonych typów zadań (np. programowania albo testowania) jest różne w zależności od fazy projektu. Fazę projektu można również określić mianem etapu projektu, a zakończenie fazy kamieniem milowym (istotnym punktem) projektu. Metoda RUP wyróżnia 4 podstawowe fazy projektu w ramach, których realizowane są zadania projektowe.

- Etap wstępny (ang. *Inception*) powinien być poświęcony przede wszystkim na pełne zrozumienie wymagań klienta. Zaleca się kierownikom projektów zaplanowanie jak największej liczby zadań związanych z modelowaniem wymagań przekazanych przez klienta.
- Etap opracowania (ang. *Elaboration*) — według metody RUP zaleca się w tym etapie projektu wykonanie stabilnej architektury tworzonego produktu (oprogramowania). Stabilna

architektura według RUP to zestaw kluczowych funkcjonalności zapewniających poprawne działanie wytwarzanego oprogramowania. Stąd też w tej fazie kierownik projektu powinien dopilnować, aby wszelkie ryzyko techniczne projektu zostało wyeliminowane.

- Etap konstruowania (ang. *Construction*), w którym powinno występować największe nasilenie prac implementacyjnych (programowanie). Zaleca się kierownikom projektów położenie największego nacisku na zorganizowanie przepływu pracy. W tej fazie powstaje najwięcej funkcjonalności i według metody RUP zaleca się wykonanie takiej liczby iteracji, jaka jest niezbędna. Przy czym dobra praktyka zawarta w RUP wskazuje, że optymalna liczba iteracji mieści się w przedziale od 6 do 12 w trakcie trwania tej fazy projektu.
- Etap przekazania (ang. *Transition*), w ramach którego powinno nastąpić przekazanie wytworzonego produktu klientowi, wykonanie testów u klienta, przygotowanie (przeszkolenie) użytkowników i uruchomienie produktu. Zaleca się menedżerom dużą koncentrację na wdrożeniu i ewentualnym usunięciu błędów i wykonaniu poprawek.

Ważną dobrą praktyką związaną z procesem realizacji projektu (Rys. 1.2) jest określenie optymalnej liczby iteracji (kolejnych przybliżeń wytwarzanego produktu) dla poszczególnych etapów projektu, a tym samym dla całego procesu realizacji projektu. W etapie *Inception* zaleca się przeprowadzenie jednej iteracji, w etapie *Elaboration* dwie iteracje, w etapie *Construction*, jak zostało wspomniane, od sześciu do dwunastu, w etapie *Transition* dwie iteracje. W ten sposób kierownik projektu może oszacować, z ilu iteracji będzie składał się jego projekt, a następnie może planować zadania dla zespołu realizującego projekt w ramach poszczególnych faz.

Dobre praktyki — Wymiar 2: Dziedziny

Jednym z podstawowych zaleceń jest przygotowywanie zadań w oparciu o pewne uniwersalne kategorie działań, jakie można spotkać w projektach informatycznych. Te kategorie określono w metodzie RUP mianem „dziedzin”. Stąd też dziedziny w metodzie RUP to po prostu zbiory zadań o podobnym charakterze. RUP wyróżnia dziewięć podstawowych dziedzin takich jak:

- Zarządzanie wymaganiami (ang. *Requirements*) — dotyczy zadań zorientowanych na zrozumienie potrzeb klienta i przełożenie ich na język zrozumiały przez zespół twórczy.
- Modelowanie biznesowe (ang. *Business modelling*) — to dyscyplina gromadząca zadania związane z modelowaniem procesów biznesowych klienta oraz modelowania funkcjonalności wytwarzanego oprogramowania.
- Analiza i projektowanie (ang. *Analysis and design*) — oznacza działania ukierunkowane na analizę m.in. ryzyka biznesowego oraz dotyczy zadań związanych z określaniem architektury budowanego oprogramowania.
- Implementacja, programowanie (ang. *Implementation*) — stanowi zbiór zadań związanych z programowaniem (tworzeniem kodu).
- Testowanie (ang. *Testing*) — stanowi zbiór zadań realizowanych przez testerów w celu sprawdzenia poprawności funkcjonowania tworzonego w projekcie informatycznym oprogramowania.
- Wdrożenie (ang. *Deployment*) — oznacza działania związane z wdrożeniem wytworzonego oprogramowania, czyli zadania wynikające z konieczności instalacji oprogramowania u klienta, szkolenia użytkowników itp.
- Zarządzanie wersjami i konfiguracją (ang. *Configuration and change management*) — oznacza wszystkie zadania wykonywane w celu wersjonowania wytwarzanego oprogramowania.
- Zarządzanie projektem (ang. *Project management*) — dotyczy wszystkich prac zarządczych wykonywanych w projekcie.

- Środowisko (ang. *Environment*) — zarządzanie elementami sprzyjającymi realizacji projektu, takimi jak przewodnik, zestawy porad i wskazówek.

Warto zauważyć, że metoda RUP dostarcza kierownikom projektów informatycznych wskazówek dotyczących etapu projektu, w którym natężenie określonych zadań jest większe. Stąd też dobrą praktyką płynącą z tego założenia jest również to, aby menedżer uwzględnił natężenie określonych zadań w zależności od fazy projektu, a tym samym umiał planować zapotrzebowanie na określonych specjalistów (np. w fazie *Inception* zapotrzebowanie na pracę analityka biznesowego jest większe niż w fazie *Transition*).

Zestawienie dobrych praktyk dla definiowania zagregowanych zmiennych decyzyjnych

Na podstawie zaprezentowanych dobrych praktyk RUP przystąpiono do wybrania i zestawienia tych dobrych praktyk, które powinny stanowić podstawę do podejmowania decyzji, a tym samym wskazywać na potencjalne zagregowane zmienne decyzyjne. Wydaje się wręcz niemożliwe uwzględnienie wszystkich zaleceń metody RUP w jednym projekcie, skoro sama metoda powstawała na bazie doświadczeń płynących z wielu projektów. Stąd też uznaje się za słuszne wyodrębnienie tylko tych dobrych praktyk zawartych w metodzie RUP, które są związane z bezpośrednią realizacją projektu informatycznego. Takie podejście jest zgodne z dzisiejszymi trendami w informatyce, gdzie dąży się do rozwiązań dopasowanych (ang. *customised/tailored*). Podejście dopasowywania jest również zgodne z teoriami zarządzania, takimi jak np. LEAN Management, gdzie mówi się o odchudzaniu procesów poprzez eliminowanie działań zbędnych. Uznano więc, że w celach adaptacji określonych dobrych praktyk te, które zostały wyodrębnione podczas omawiania metody RUP należy zestawić w zbiorczej tabeli wraz ze wskazaniem, kiedy dana dobra praktyka powinna zostać wykorzystana (kolumna UZASADNIENIE). Zestawienie dobrych praktyk wyodrębnionych na podstawie analizy metody RUP zaprezentowano w tabeli 1.5. Zdecydowano się na wybór stałej liczby dobrych praktyk z każdej omawianej metody zarządzania projektami. Takie zawężenie (do 6 dobrych praktyk) było konieczne z punktu widzenia późniejszego algorytmu pozwalającego dopasowywać dobre praktyki do zagregowanych zmiennych decyzyjnych.

Tab. 1.5 Wybrane dobre praktyki zarządzania projektami informatycznymi na podstawie RUP

DOBRA PRAKTYKA	UZASADNIENIE
Realizacja projektu w czterech fazach	Pozwala kierownikom projektu na lepsze planowanie działań i uporządkowanie procesu realizacji projektu.
Definiowanie produktów pracy jako wejścia/wyjścia do/z zadań	Pozwala kierownikowi na przekazanie zespołowi oczekiwań co do zrealizowania określonego zadania.
Definiowanie precyzyjnych ról projektowych	Pozwala kierownikowi zorganizować zespół w zależności od określonych działań.
Planowanie struktury przepływu pracy	Pozwala kierownikowi uporządkować przepływ zadań i informacji w projekcie, pokazując logiczne następstwa zadań.
Stosowanie modeli wysokopoziomowych	Pozwala kierownikowi zaprezentować produkt projektu informatycznego w sposób bardziej zrozumiały przez klienta.
Gromadzenie zadań w ramach dziedzin	Pozwala kierownikowi planować zadania w zależności od fazy i przewidywać natężenie określonych prac w ramach faz projektu.

Źródło: opracowanie własne na podstawie [9]

Ocena metody RUP

Przeprowadzona analiza metody RUP miała za zadanie wyłonienie dobrych praktyk, które kierownicy powinni wykorzystywać w swoich projektach, jak też tych, które będą niezbędne dla wyodrębnienia zagregowanych zmiennych decyzyjnych. Wydaje się jednak, że dla pełnej oceny RUP konieczne staje się także spojrzenie na RUP z punktu widzenia przydatności do zarządzania projektem informatycznym. O ile bowiem metoda RUP precyzuje, jak należy planować projekt informatyczny

i zawiera szereg dobrych praktyk planistycznych, o tyle nie zawiera wskazówek, w jakich warunkach należy tej metody używać. Wskazuje się, że może być stosowana do małych lub dużych projektów (ang. *RUP for Small Projects*, *RUP for Large Projects*) jednakże nie podaje się, które dobre praktyki przydają się w przypadku małych, a które w przypadku dużych projektów.

Drugim bardzo istotnym wskaźnikiem oceny jest brak warunków wstępnych, które należy spełnić, aby RUP można było stosować. Przykładem mogą być dobre praktyki ABCDEF omówione powyżej. Wytyczne dla ich stosowania są zorientowane biznesowo i dotyczą zarządzania na poziomie strategicznym i taktycznym. Nie wskazuje się jednak, dla jakiego klienta, dla jakiego zespołu albo chociaż przy jakim rodzaju projektów powinny być stosowane. Wydaje się więc, że brakuje tym dobrym praktykom wskazania kontekstu (zespołu, procesów, technologii) ich odniesienia.

Kolejna cecha uwidoczniła podczas analizy metody RUP, która jest istotna dla kierowników projektów, to złożoność tej metody. Zaletą takiego ujęcia jest szczegółowe omówienie wszystkich istotnych procesów projektów informatycznych. Takie podejście stwarza warunki młodym, niedoświadczonym kierownikom na dotarcie do wiedzy o najważniejszych aspektach zarządzania projektami informatycznymi. Z drugiej jednak strony taka mnogość opisów procesów powoduje, że wyodrębnienie głównych składników metody RUP osobom dopiero zapoznającym się z problematyką zarządzania przedsięwzięciami może sprawiać trudności.

Liczba procedur, zasad postępowania i szczegółowych opisów powoduje, że RUP prezentuje się jako metoda mało elastyczna i mało odporna na zmiany w projekcie informatycznym. Stosowanie tej metody w przypadku projektów o dużej dynamice zmian niesie oczywiście zagrożenie. Przyjmuje się bowiem, zgodnie z podejściem RUP, że cel projektu powinien być znany od początku, a wszystkie działania zaplanowane długofalowo. Jednak planowanie długookresowe w projektach informatycznych jest właściwie niemożliwe (zwłaszcza w projektach o dużej dynamice zmian). Dlatego też metodę RUP należy stosować wtedy, kiedy oczekiwania klienta są bardzo dobrze znane i cel projektu jest wyraźnie określony. Określony precyzyjnie powinien być również produkt projektu informatycznego. W sytuacji, kiedy klient nie do końca wie, jakie funkcjonalności systemu informatycznego są dla niego satysfakcjonujące (czyli w sytuacji dużej zmienności wymagań w projekcie), metoda RUP nie jest odpowiednia.

Ta niska elastyczność RUP wiąże się przede wszystkim z omówionym wcześniej podejściem procesowym. Udział procesów planowania w metodzie RUP jest bowiem bardzo znaczny. Dobre praktyki zalecają zaplanowanie każdego etapu projektu, każdego zadania, a jeszcze dodatkowo produktu pracy. Ta znaczna liczba działań planistycznych powoduje jednocześnie, że od menedżera oczekuje się pełnego zdefiniowania nie tylko całego procesu realizacji, ale również wszystkich innych elementów struktury projektowej.

Podsumowując przeprowadzoną ocenę metody RUP, należy podkreślić, że jest to metoda dobrze opisana, bardzo szczegółowa i przez swoją dużą liczbę procedur/procesów/dobrych praktyk logicznie ułożona. Niestety, nie jest wolna od wad, z których największą wydaje się brak wskazówek dotyczących kontekstu stosowania tej metody. Nie jest powiedziane ani dla jakich projektów, ani dla jakich zespołów należy stosować zasady zawarte w RUP. Wydaje się zatem, że bardzo istotne jest wyodrębnienie z metody RUP tylko tych dobrych praktyk, które są użyteczne z punktu widzenia kierowania własnym projektem po określeniu kontekstu własnego projektu.

1.2.2 Przykład metody klasycznej — PRINCE2

Kolejną prezentowaną w pracy metodą klasyczną zarządzania jest PRINCE2 (ang. *Projects in a Controlled Environment*). Jest ona istotna dla wyodrębnienia dobrych praktyk stanowiących podstawę dla zagregowanych zmiennych decyzyjnych. Prezentacja tej metody jest konieczna. Jest ona istotna także dla oceny jej przydatności dla potrzeb zarządzania projektami informatycznymi. PRINCE2 zakłada, że sukces w realizacji projektu informatycznego zapewniony jest przez szczegółowe

zaplanowanie nie tylko poszczególnych etapów projektu, ale wszystkich elementów struktury projektu. Pod pojęciem struktury projektu należy rozumieć zestaw powiązanych relacjami elementów występujących w projekcie informatycznym, względem których kierownik podejmuje określone decyzje. Zaliczyć do niej należy zarówno zasoby ludzkie (członkowie zespołu, klient, eksperci), jak również zasoby wiedzy (metody zarządzania projektami, dobre praktyki zarządzania projektami, procedury postępowania).

Metoda PRINCE2 uznawana jest obecnie za metodę zarządzania dowolnym projektem (nie tylko informatycznym), ale swoimi korzeniami osadzona jest w projektach informatycznych. Należy w tym momencie zauważyć pewne podobieństwo do metody RUP, gdzie dobre praktyki również opracowano na bazie wieloletnich doświadczeń projektowych i stworzono z nich formalne (proceduralne) podejście do zarządzania projektami informatycznymi. Stąd też analizując podejście PRINCE2 do zarządzania projektami należy przyrzeć się zawartym w tej metodzie dobrym praktykom, które kierownicy projektów mogą adaptować do swoich projektów.

Metodę PRINCE2 odróżnia od innych metod bardzo wyraźna standaryzacja. Metoda zawiera bowiem szereg opisów odnośnie struktury zespołu realizującego projekt, jak również pełną standaryzację procesów realizacji projektu z podziałem na główne etapy i kroki w ramach poszczególnych etapów. Ten bardzo proceduralny charakter metody PRINCE2 zostanie omówiony w dalszej części. Oprócz dość sztywnych ram związanych z organizacją zasobów w ramach projektu realizowanego według metody PRINCE2 oraz zaleceń odnośnie dokładnego planowania poszczególnych etapów, metoda PRINCE2 koncentruje się w dużej mierze na roli klienta. Klient według metody PRINCE jest elementem złożonej hierarchii organizacyjnej realizowanego projektu i umiejscowiony jest na szczycie tej hierarchii.

Te główne wyróżniki metody PRINCE2 zostaną poddane analizie pod kątem wyodrębnienia z nich dobrych praktyk zarządzania procesami, zespołem, zadaniami i zakresem oraz dokumentacją, które są istotne dla kierownika projektu. Jakkolwiek w metodzie PRINCE2 jest to szczególnie trudne, gdyż tutaj kierownik projektu określany jest jako jedna z ról niższego szczebla, a nadzór nad całym projektem prowadzi specjalnie powołany Komitet Sterujący. Stąd też należy podkreślić, że dobre praktyki zawarte w PRINCE2 odnoszą się nie tylko do kierownika projektów, ale do całej organizacji realizującej projekt, z naczelnym kierownictwem włącznie.

Dobre praktyki PRINCE2: Procesy

Jak zaznaczono na wstępie, PRINCE2 charakteryzuje bardzo formalne podejście. Mówi się wręcz o pełnej standaryzacji każdego elementu projektu. Dobre praktyki metody PRINCE2 zalecają na przykład pełną definicję etapu procesu realizacji projektu celem sprawowania nad nim właściwej kontroli. W ramach metody przyjęto, że tylko stały nadzór i monitorowanie poszczególnych etapów umożliwiają zakończenie projektu.

Pierwszą dobrą praktyką związaną z planowaniem poszczególnych elementów projektu jest położenie nacisku na planowanie oparte na produktach. Przyjmuje się, że jeżeli produkt jest zdefiniowany, członkowie zespołu projektowego powinni realizować takie procesy, które doprowadzą do powstania określonego produktu. Aby to jednak było możliwe, produkt należy bardzo precyzyjnie zdefiniować: podać miary (czasowe, finansowe), a także należy określić trzy podstawowe parametry produktu: strukturę produktu (ang. *Product Breakdown Structure*), jego opis (ang. *Product Description*), a także diagram następstwa produktu (ang. *Product Flow Diagram*). Jak łatwo zauważyć, w miejsce przepływów pracy (ang. *Workbreakdown Structure*) występujących w innych metodach zarządzania wprowadzono przepływy w projekcie uzależnione od produktów, nie od procesów, traktując procesy jako element niezbędny, ale pochodny od produktów.

Drugą dobrą praktyką metody PRINCE2 jest wyodrębnienie w projekcie dwóch zasadniczych etapów projektu informatycznego. Stosując się do niej komitet sterujący (ang. *Steering Committee*) powinien podzielić projekt na etap zarządczy związany z inicjowaniem projektu oraz etap realizacji.

Przyjmuje się, że etap realizacyjny powinien składać się z wielu pomniejszych etapów (iteracji w przypadku projektów informatycznych).

Trzecia dobra praktyka stosowana w podejściu PRINCE2 wskazuje na potrzebę tworzenia precyzyjnego harmonogramu projektu, który powstaje na etapie inicjowania projektu. Etap realizacyjny (kolejny po etapie inicjacji) jest więc od początku kontrolowany i sterowany. Kolejna dobra praktyka odnosi się do przebiegu prac projektowych. Sugeruje się podtrzymanie pełnej sterowalności (czyli dokładnego definiowania i kontrolowania) wszystkich zasobów wykorzystywanych do realizacji zadań. Dobre praktyki metody PRINCE2 odnoszą się również do etapu zakończenia projektu, czyli kontrolowania stanu projektu już po zakończeniu wszystkich prac. Na tym etapie sugeruje się również, aby po przekazaniu produktu do klienta zaplanować procesy utrzymania produktu.

Ostatnią dobrą praktyką jest zdefiniowanie poszczególnych podprocesów prowadzących do zrealizowania celu projektu i wytworzenia właściwego produktu.

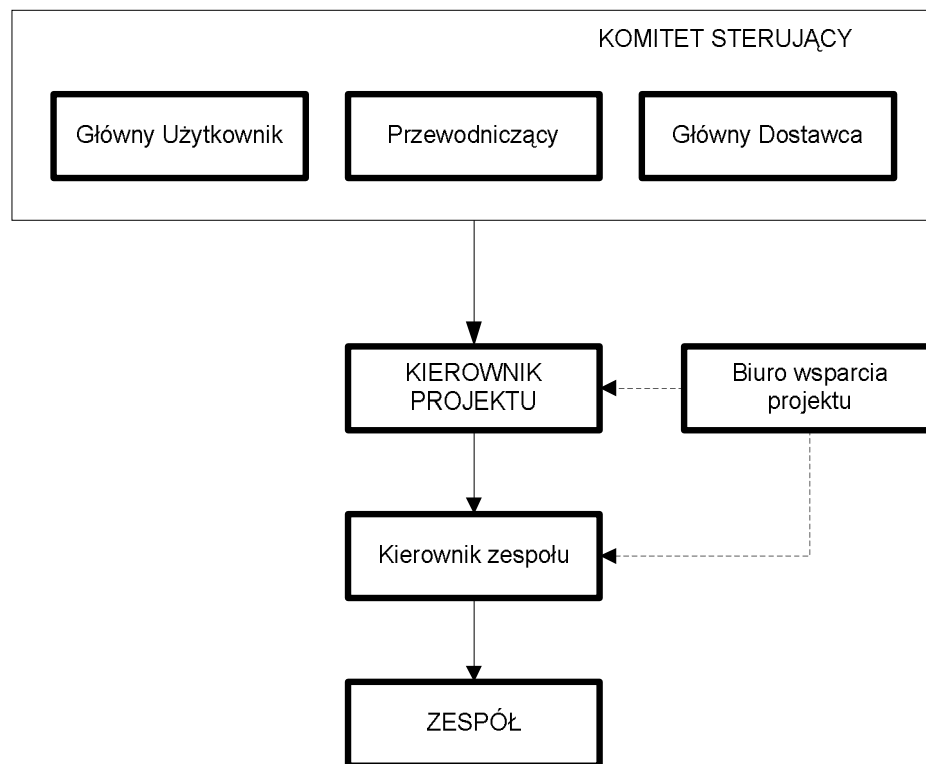
Dobre praktyki PRINCE2: Zespoły

Trudno mówić o dobrych praktykach dla zespołu projektowego, gdyż metoda traktuje zespół jako organizację, przyjmując zależności hierarchiczne w ramach projektu. Zgodnie z dobrymi praktykami PRINCE2 w ramach projektu należy powołać Komitet Sterujący. Z kolei w ramach Komitetu Sterującego PRINCE2 wyodrębnia się role i obowiązki z nimi związane. W skład Komitetu Sterującego wchodzi Przewodniczący (najczęściej reprezentant klienta), Główny Użytkownik (reprezentant strony użytkownika) oraz Główny Dostawca (reprezentujący stronę wytwórczą realizowanego projektu informatycznego). Przewodniczący to przedstawiciel podmiotu finansującego realizację projektu. Główny Użytkownik reprezentuje osoby zamierzające wykorzystać produkt powstały w wyniku realizacji projektu informatycznego. Główny Dostawca to osoba zapewniająca realizację projektu informatycznego przy wykorzystaniu odpowiednich technologii i umiejętności.

W skład Komitetu Sterującego nie wchodzi Kierownik Projektu, ponieważ przyjmuje się, że w myśl dobrych praktyk podlega Komitetowi Sterującemu. Komitet Sterujący podejmuje decyzje na poziomie strategicznym (decyzje biznesowe), zaś Kierownik Projektu ma za zadanie organizować prace operacyjne wykonywane w ramach projektu, a następnie raportować Komitetowi Sterującemu stan projektu.

W ramach metody PRINCE2 zaleca się również, aby Kierownik Projektu odpowiadał za organizowanie i sterowanie projektem. Ma być więc nie tylko osobą odpowiedzialną przed Komitetem Sterującym za postępy prac, ale również za właściwe zorganizowanie wszystkich procesów zgodnie z metodą PRINCE2. Kierownik projektu jest także odpowiedzialny za dobór członków zespołu, jak również powinien być gwarantem terminowej realizacji projektu.

Oprócz Komitetu Sterującego oraz Kierownika Projektu w metodzie PRINCE2 sugeruje się powołanie osób do pełnienia funkcji kontrolnych. Takie role określa się mianem Nadzoru Projektu (nadzorem mogą być objęte działania zarówno Klienta, jak i Głównego Użytkownika oraz Głównego Dostawcy). Funkcje administracyjne (związane z dokumentowaniem prac, rozliczaniem finansów) powierza się Biurze Wsparcia Projektów. Powołuje się także Właściciela Ryzyka, osobę odpowiadającą za zachowania w sytuacjach kryzysowych i działaniach w przypadku niezgodności z wcześniej ustalonym harmonogramem. Na rys. 1.3 zobrazowano konstrukcję zespołu projektowego zgodnie z metodą PRINCE2.



Rys. 1.3 Konstrukcja organizacji projektowej wg PRINCE2

Źródło: opracowanie własne na podstawie[93,94,95]

Dobre praktyki PRINCE2: Zarządzanie zadaniami i dokumentacją

Metoda PRINCE2 klasycznym przepływowi pracy przeciwstawia koncepcję przepływu produktów (czyli efektów) pracy w ramach poszczególnych etapów. Zamknięcie każdego etapu stanowi kamień milowy i po jego akceptacji przez Komitet Sterujący rozpoczyna się realizacja zadań w następnym etapie. Każdy etap realizacyjny powinien zostać poddany kontroli ze strony Komitetu Sterującego, dzięki czemu nastąpi weryfikacja, czy dostarczono wszystkie wymagane produkty i czy produkty te spełniają zaplanowane uprzednio wymagania biznesowe. Po weryfikacji i akceptacji produktów danego etapu projektu następuje tworzenie szczegółowego planu kolejnego etapu.

Obowiązek prowadzenia i przygotowywania dokumentacji jest cechą charakterystyczną wszystkich klasycznych metod zarządzania projektami. Dobre praktyki metody PRINCE2 związane z dokumentacją sugerują kategoryzację dokumentacji w czterech podstawowych obszarach: dokumentacja inicjująca projekt, rejestry, raporty i plany. Dokumentacja inicjująca projekt obejmuje te dokumenty, które związane są z podstawowymi założeniami do realizacji danego projektu informatycznego. Stąd też wymaga się w tym obszarze takich dokumentów jak: dokument zawierający kontekst projektu, dokument definiujący instrumenty sterowania, dokument zawierający definicję projektu (a w nim cele, założenia, metody osiągania i mierzenia celów, spodziewane produkty poszczególnych etapów, ograniczenia projektowe i wiele innych). Dodatkowo sugeruje się kierownikom przygotowanie dokumentacji zawierającej uzasadnienie ekonomiczne projektu, strukturę organizacyjną stworzoną w projekcie, plan projektu (obejmujący produkty, działania i zasoby) oraz definicję przepływów informacyjnych (tzw. plan komunikacji). Dokumentacja rejestrowa według metody PRINCE2 powinna zawierać rejestr ryzyka, rejestr doświadczeń czy rejestr zarządzania konfiguracją projektu informatycznego. Natomiast w skład dokumentacji raportowej, mającej na celu zapewnienie odpowiedniej sprawozdawczości w projekcie, powinny wchodzić m.in. takie dokumenty jak raporty okresowe, raport końcowy, raport z punktu kontrolnego itp. Dobre praktyki odnoszą się

również do uporządkowania dokumentów planistycznych sugerując tworzenie takich planów jak plany poszczególnych etapów, ale również plan awaryjny, plan wyjątkowy (naprawczy), plan pracy itp.

Zestawienie dobrych praktyk dla definiowania zagregowanych zmiennych decyzyjnych

Na podstawie zaprezentowanych dobrych praktyk PRINCE2 przystąpiono do wybrania i zestawienia tych dobrych praktyk, które powinny stanowić podstawę do podejmowania decyzji, a tym samym wskazywać na potencjalne zagregowane zmienne decyzyjne.

Tab. 1.6 Wybrane dobre praktyki zarządzania projektami informatycznymi wg PRINCE2

DOBRA PRAKTYKA	UZASADNIENIE
Powołanie organu zarządzającego w postaci Komitetu Sterującego	Pozwala na utworzenie zależności hierarchicznych pomiędzy wszystkimi osobami zaangażowanymi w projekt.
Definiowanie celów projektu	Pozwala na precyzyjne określenie kierunku rozwoju projektu i przebiegu prac projektu.
Powołanie biura wsparcia projektu	Pozwala na wyodrębnienie grupy osób, które zajmą się obsługą formalno-prawną.
Przygotowanie produktów pośrednich projektu	Pozwala na precyzyjne definiowanie zadań realizowanych w trakcie projektu.
Przygotowanie dokumentacji na każdym etapie projektu	Pozwala na posiadanie odpowiedniej dokumentacji zawierającej zapis realizacji całego projektu od początku do końca.
Definiowanie wszystkich procesów potrzebnych do realizacji projektu	Pozwala na uporządkowanie przepływu zadań w projekcie, przez co członkowie zespołu wiedzą, jakie zadania wykonać.

Źródło: opracowanie własne na podstawie [93,94]

Ocena metody

PRINCE2 to niewątpliwie ciężka metoda zarządzania projektem informatycznym. Oceniając tę metodę z punktu widzenia dobrych praktyk dla menedżerów projektów, należy zacząć od oceny dobrych praktyk odnoszących się do procesów. Niewątpliwie im bardziej precyzyjne jest planowanie poszczególnych etapów projektu, tym łatwiej powinno się je realizować. Jednak nasuwa się w tym miejscu pytanie, jak dokładne może być planowanie etapu w projekcie informatycznym. A tym samym, czy definiując w ten sposób etapy oraz zasoby i procesy, a nawet wszystkie podprocesy, kierownik projektu uchroni się przed tak typową dla projektów zmiennością? Niemal automatycznie nasuwa się odpowiedź przecząca. Planowanie długookresowe w burzliwym (dynamicznym) otoczeniu nie znajduje zatem większego zastosowania. Wydaje się zatem, że dobra praktyka płynąca z oceny procesów w PRINCE2, to potrzeba definiowania etapów projektu przyrostowo, rzadziej dla całego projektu.

Kolejnym zestawem dobrych praktyk jest sposób organizacji zespołu projektowego, a ściślej mówiąc hierarchiczna struktura ról w projekcie. Oczywiście z punktu widzenia przepływu informacji oraz określenia stopnia ważności, taka struktura wydaje się jak najbardziej słuszna. Jednakże rola kierownika projektu wydaje się nieco umniejszona. Kierownik projektu ma co prawda swobodę w kwestii organizowania prac projektowych, ale z punktu widzenia relacji pomiędzy odbiorcą projektu (Klientem) a osobą odpowiedzialną za realizację projektu (Kierownik Projektu) zalecenie umiejscowienia kierownika jako roli podległej względem klienta może być zjawiskiem utrudniającym budowanie poprawnych relacji.

Kolejną cechą metody PRINCE2 jest zastosowanie procesów kontroli, sterowania i monitorowania procesów projektu. Takie podejście poddaje pod wątpliwość zgodność tych procesów z nowoczesnymi koncepcjami zarządzania wskazującymi na eliminację procesów kontroli na rzecz precyzyjniejszego planowania. Jeżeli zatem w ramach metody PRINCE2 sugeruje się położenie nacisku na planowanie wszystkich etapów (o czym pisano uprzednio), to czy konieczna jest także dokładna weryfikacja i kontrola procesów projektowych? Wydaje się bowiem, że tak częste

stosowanie procesów kontrolnych może zniechęcić członków zespołów do stałego raportowania wykonywanych zadań.

Zastosowanie dobrych praktyk dla gromadzenia dokumentacji jest niewątpliwie zaletą tej metody, ale też wymaga wykonania znacznych prac. Rodzi się wątpliwość, czy w metodzie PRINCE2 nie występuje problem nadmiarowości dokumentacji, przez co kierownik projektu zamiast dbać o postępy prac, koncentruje się na kontrolowaniu/przygotowaniu dokumentacji.

Ważnym składnikiem oceny metody PRINCE2 jest wysoki poziom standaryzacji odniesiony do organizacji, realizowanych przez nią zadań oraz produktów. Zaletą takiego podejścia jest ład w projekcie i zapewnienie odpowiedniego podziału kompetencji oraz podziału prac.

1.2.3 Przykład metody lekkiej — SCRUM

Po prezentacji metod klasycznych, zawartych w nich dobrych praktyk oraz wyborze tych, które mogą być wykorzystane do definiowania zagregowanych zmiennych decyzyjnych zostaną w podobny sposób zaprezentowane metody lekkie. Celem tych prezentacji będzie z jednej strony dekompozycja metod do poziomu dobrych praktyk, z drugiej — wykazanie, na ile odmienne podejście do zarządzania projektami może stanowić źródło takich dobrych praktyk i potencjalnie zawęzić zakres zagregowanych zmiennych decyzyjnych.

Jako pierwsza zostanie przedstawiona metoda SCRUM, która w odróżnieniu od wcześniej przedstawionych metod ciężkich należy do kategorii metod zwinnych (lekkich) (ang. *Agile*). Metody te charakteryzują się większą elastycznością i odpornością na zmiany niż to ma miejsce w przypadku metod ciężkich [80]. Metody lekkie minimalizują także procesy planowania, kładąc większy nacisk na działanie (realizację zadań) w możliwie krótkich cyklach, po których weryfikuje się poczynione kroki i wyznacza kolejne. Podobnie jak to miało miejsce w przypadku metod ciężkich, metody lekkie zostaną omówione z punktu widzenia najważniejszych elementów struktury projektu w postaci dobrych praktyk odniesionych do realizowanych zadań oraz zarządzania zespołem.

W metodzie SCRUM należy wskazać na trzy obszary dobrych praktyk związanych z zarządzaniem projektem. W pierwszym dobre praktyki są nastawione na zarządzanie zespołem. Drugi obszar to praktyki nastawione na zarządzanie zadaniami/zakresem projektu. Obszar trzeci to dobre praktyki skoncentrowane na przygotowaniu dokumentacji. Poniżej krótko scharakteryzowano te obszary dla wyboru — podobnie jak to miało miejsce uprzednio — sześciu dobrych praktyk, które stanowić będą podstawę do doboru zagregowanych zmiennych decyzyjnych.

Dobre praktyki — zarządzanie zespołem

Spotkania (ang. Daily Scrum)

Zaleca się, aby zespół każdego dnia spotkał się na krótkiej naradzie „operacyjnej” w celu zwiększenia przepływu informacji między członkami zespołu. W trakcie tych krótkich, kilkunastominutowych spotkań zaleca się, aby członkowie zespołu odpowiedzieli na trzy zasadnicze pytania:

- Co zrobiłeś wczoraj?
- Co zrobisz dzisiaj?
- Co Ci przeszkadza w realizacji zadań?

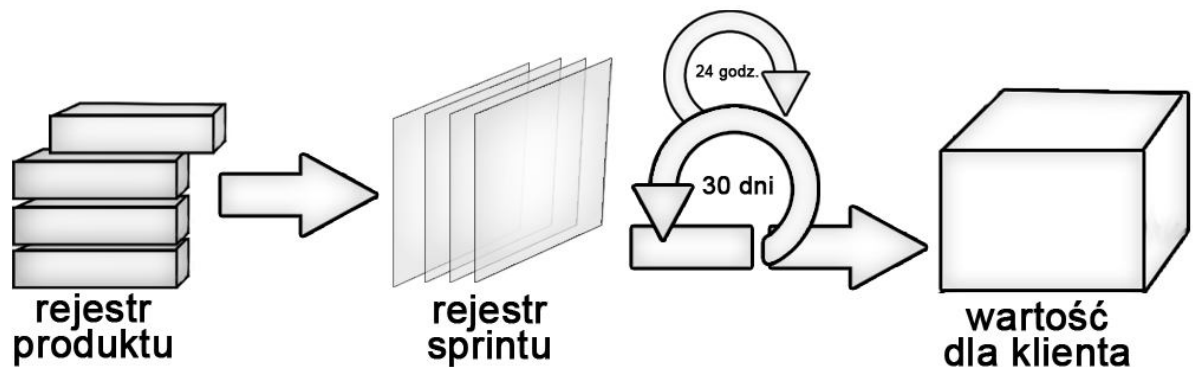
Udział klienta w projekcie

Drugą dobrą praktyką jest zapewnienie udziału klienta w realizacji prac projektowych. Mówi się o tym, że klient powinien uczestniczyć w spotkaniach codziennych, bez prawa ingerencji w planowanie zadań, ale za to z głosem opiniotwórczym w stosunku do wytwarzanego produktu. Dzięki takiemu podejściu możliwe jest zminimalizowanie zmienności wymagań klienta. Z klientem

wiąże się jeszcze jedna dobra praktyka, która mówi, że każdy etap projektu musi dostarczyć klientowi kolejną działającą wersję produktu, którą mógłby ocenić.

Procesy wytwarzania

Podstawowym czasem realizacji zadań projektowych stają się sprinty. Pod pojęciem sprintu należy rozumieć etap realizacji projektu, po którym powstaje produkt pośredni projektu w postaci fragmentu działającego oprogramowania, który można zaprezentować klientowi. Natomiast dobra praktyka zawarta w metodzie SCRUM wskazuje na długość sprintu — nie powinien być dłuższy niż 6 tygodni. Przyjmuje się, że każdy taki sprint powinien być uprzednio zaplanowany wspólnie przez zespół. Należy przy tym zaznaczyć, że o ile dobre praktyki metody SCRUM wskazują jak długi powinien być sprint i wspominają o spotkaniach codziennych, o tyle nie definiują konkretnej liczby sprintów. Proces realizacji projektu według metody SCRUM został zaprezentowany na rysunku 1.4.



Rys. 1.4 Proces realizacji projektu wg metody SCRUM

Źródło: opracowanie własne na podstawie[96]

Jak wynika z rysunku 1.4, zebrane przez klienta wymagania w postaci wykazu prac do realizacji gromadzone są w rejestrze produktu (ang. *Product Backlog*) i zostają podzielone na mniejsze jednostki nazywane dziennikiem sprintu lub rejestrem sprintu (ang. *Sprint Backlog*). Rejestr sprintu stanowi podstawę do planowania sprintu. Na rysunku 1.6 zaprezentowano także dwie pętle — mniejsza obrazuje codzienne spotkania, większa — przebieg sprintu. Każdy zaś sprint powinien zakończyć się prezentacją dokonań (ang. *Sprint Review*). Dobrą praktyką sprintu jest przegląd sprintu w postaci 2-godzinnych prezentacji. Celem takiego przeglądu jest nie tylko przedstawienie efektów pracy zespołu podczas sprintu, ale również podjęcie decyzji co do ewentualnych korekt oraz przyjęcie ram do planowania kolejnego sprintu.

Kolejny zestaw dobrych praktyk związany jest z organizacją samego zespołu oraz organizacją pracy tego zespołu. Sugeruje się tylko trzy podstawowe typy ról, w które mogą wcielić się uczestnicy projektów:

- Właściciel Produktu (ang. *Product Owner*) określa wizję i cel projektu oraz zarządza rejestrem produktu [32,36].
- Mistrz Młyna (ang. *Scrum Master*) to osoba, której zadaniem jest zapewnienie poprawnej organizacji pracy zespołu zgodnej z zasadami metody SCRUM.
- Zespół projektowy (ang. *Development Team*) — nie jest pojedynczą rolą, jak Mistrz Młyna czy Właściciel Produktu. Zespół projektowy rozumiany jest jako grupa składająca się z kilku osób (najlepiej od 4 do 6). Metoda SCRUM zaleca, aby zespół był interdyscyplinarny, przy

czym nie profiluje pojedynczych ról ze względu na rodzaj zadań (jak np. w RUP — tester, programista itp.), ale każdego członka określa mianem „deweloper”.

Kolejny zestaw dobrych praktyk odnoszących się do zespołu dotyczy zarządzania zespołem i sposobu organizowania jego pracy. W SCRUM sugeruje się kierownikom projektów, aby zespół był samoorganizujący się i wielofunkcyjny. Stąd też w skład zespołu powinni wchodzić zarówno specjaliści od modelowania, programowania czy testowania. Zaleca się także, aby członkowie zespołu samodzielnie decydowali, jakie zadania mają wykonać w danym sprincie projektu.

Dobre praktyki — Zarządzanie zadaniami/zakresem projektu

Podstawowa dobra praktyka zarządzania zadaniami wskazuje, że wszystkie wymagania pozyskane od klienta dotyczące funkcjonalności wytwarzanego oprogramowania w ramach projektu należy zgromadzić w postaci zadań. Zadania te powinny stanowić logiczny zbiór dostępny dla wszystkich uczestników w rejestrze produktów.

Kolejna dobra praktyka wskazuje, że w trakcie planowania sprintu zespół podejmuje decyzje, co zamierza wykonać i tylko zespół ma prawo zmodyfikować rejestr sprintu. Przyjmuje się także, że zadania są wykonywane na podstawie przyjętych przez zespół priorytetów. Zakłada się również, że jeżeli (z różnych przyczyn) któreś zadanie nie zostało wykonane w ramach danego sprintu, powinno zostać zrealizowane w następnym, jako zadanie o przyjętym przez zespół priorytecie.

Oprócz tworzenia rejestru sprintu i rejestru produktu, dobrą praktyką zarządzania projektami według metody SCRUM jest stałe monitorowanie stanu realizacji projektu. Podstawowym narzędziem wykorzystywanym w tej metodzie jest tzw. wykres wypalania (ang. *Burndown Chart*) [96]. Dobrą praktyką związaną z monitorowaniem stanu projektu jest również zalecenie dodawania do wykresu projektu uwag i komentarzy. Tworzy się w ten sposób repozytorium projektu z bazą wiedzy o bieżących zadaniach i pojawiających się problemach. Przykładowy wykres wypalania wraz z komentarzami zespołu projektowego przedstawiony został na rys. 1.5.



Rys. 1.5 Tablica informacyjna projektu realizowanego wg metody SCRUM (*Scrumboard*)

Źródło: [101]

Dobre praktyki — tworzenie dokumentacji

Dokumentacja jest jednym z kluczowych elementów projektu pokazującym przebieg realizowanych działań i mogącym stanowić punkt wyjścia do przyszłych usprawnień. Dobre praktyki w SCRUM wskazują, że ważniejsze dla powodzenia projektu jest działające oprogramowanie niż

dokumentacja. Stąd też ogranicza się dokumentację najczęściej do wymagań technicznych czy opisu w formie modelu (np. diagramy UML).

Zestawienie dobrych praktyk w SCRUM

Zgodnie z założeniami tego rozdziału jego celem jest prezentacja dobrych praktyk zawartych w wybranych do oceny metodach zarządzania projektami informatycznymi. Te wybrane i zamieszone w tabeli 1.5 odnoszą się zarówno do dobrych praktyk zarządzania zespołem, jak i zadaniami. W metodzie SCRUM procesy zarządzania zespołem odnoszą się do Mistrza Młyna i Właściciela Produktu, z kolei zarządzanie zadaniami zawartymi w rejestrze produktu i rejestrze sprintu powierzone jest całemu zespołowi. Stąd też wykorzystanie dobrych praktyk z zakresu zarządzania zespołem jak i zadaniami zależne są od stanu dojrzałości organizacji dostawcy (przyjęcie ról i ich realizacja), jak też od stanu dojrzałości klienta (reprezentant klienta w zespole SCRUM). Z kolei zakres zadań zawartych w obu dokumentach — dzienniku produktu i sprintu — zależy od możliwości opisu zakresu projektu (jego zadań przewidzianych do realizacji).

Tab. 1.7 Dobre praktyki wg metody SCRUM

DOBRA PRAKTYKA	UZASADNIENIE
Realizacja prac w krótkich sprintach (przebiegach)	Pozwala kierownikowi na planowanie zadań tylko w ramach krótkiego cyklu wytwarzania oraz pozwala na częste prezentowanie wartości dla klienta.
Codziennie spotkania zespołu oraz przedstawiciela klienta	Pozwala na zwiększenie przepływu informacji w ramach projektu oraz na aktywne uczestnictwo klienta.
Wyznaczenie osoby (np. Mistrza Młyna) do nadzoru procesu realizacji	Pozwala na zapewnienie właściwego nadzoru realizacji projektu zgodnie z przyjętą metodą (założeniami).
Sporządzenie rejestru produktu	Pozwala na pogrupowanie wymagań zebranych przez klienta oraz na spojrzenie na projekt jako całość (ilość pracy do wykonania).
Sporządzenie rejestru sprintu (przebiegu) i przyjęcie priorytetów realizacji zadań	Pozwala na ustalenie, jakie zadania powinny być zrealizowane najpierw w ramach danego przebiegu (porządkuje przepływy zadań).
Monitorowanie projektu za pomocą wykresu wypalania	Pozwala w wygodny sposób śledzić postępy w realizacji projektu i uwidocznić je wszystkim uczestnikom.

Źródło: opracowanie własne [32,36]

Ocena metody SCRUM

Przystępując do oceny metody SCRUM należy zwrócić uwagę na specyficzne cechy, które odróżniają tę metodę od innych i dobrych praktyk wynikających z dekompozycji tej metody. Pierwszą dobrą praktyką tej metody jest stosowanie specyficznych ról związanych z wewnętrzną kontrolą wykorzystywania SCRUM. Obecność Mistrza Młyna, który czuwa nad poprawnym stosowaniem metody SCRUM, jest zjawiskiem nowym w metodach zarządzania projektami. Jego obecność po pierwsze powoduje, że prace realizowane w projekcie są stale nadzorowane, po drugie wprowadza poczucie pewności zespołu. Przydatne wydają się również wszelkiego rodzaju formy dokumentacji zamieszczonej w repozytorium projektu, ale odniesionej do stanu prac zespołu (wykresy wypalania wraz z uzupełnieniami) oraz tworzenie tablic informacyjnych (ang. *scrum boards*) z informacjami również dla klienta. Dzięki takiemu podejściu włącza się klienta w planowanie sprintów i zapewnia przejrzystość realizacji projektu.

Podsumowując, należy jednak stwierdzić, że o ile SCRUM zawiera szereg dobrych praktyk, o tyle brakuje w opisie metody zasad mówiących o warunkach ich stosowania. Nie należy zgodzić się z założeniem, że metoda SCRUM może być stosowana do każdego projektu [75]. W przypadku

wieloletnich infrastrukturalnych projektów informatycznych, w których zmienność wymagań jest niewielka, a liczba osób zaangażowanych przekracza kilkadziesiąt, stosowanie metody SCRUM wydaje się trudne. W przypadku zespołów małych spotkania codzienne będą miały zastosowanie. W przypadku jednak dużych projektów, o wymiarze transnarodowym, jak również wymagających zaangażowania bardzo wielu zasobów, zasady zawarte w metodzie SCRUM będą trudne do stosowania. Stąd też dobre praktyki oraz stopień ich wykorzystania będą miały sens w przypadku grup projektów oraz zespołów dostawcy i klienta o określonych poziomach dojrzałości. Wydaje się, że brak takiej analizy (stanu projektu oraz stanu zespołów) może powodować, że zastosowanie wybranej metody zarządzania bądź jej dobrych praktyk może być ograniczone.

1.2.4 Przykład metody lekkiej — EXTREME PROGRAMMING

Drugą z metod lekkich, która została podobnie jak SCRUM poddana analizie z punktu widzenia przydatności do zarządzania projektami jak też dekompozycji do poziomu dobrych praktyk, była metoda programowania ekstremalnego XP (ang. *Extreme programming*) [67]. Jej dobór wynikał ze znacznej przydatności tej metody do zarządzania projektami, jak też jej reprezentatywności grupy metod lekkich. Metoda ta była odpowiedzią na dość proceduralne i formalne prowadzenie projektów z wykorzystaniem metod ciężkich [74]. Zamysłem głównego twórcy metody, Kenta Becka, było opracowanie metody zarządzania projektami informatycznymi bazującej na perspektywie programisty, a nie perspektywie klasycznego zarządzania. Tym samym metoda ta stanowi zestaw dobrych praktyk dotyczących sposobu realizacji projektów informatycznych. Poniżej zebrano dobre praktyki XP zagregowane w trzech grupach: realizacji procesów, zarządzania zespołem oraz wytwarzania dokumentacji.

Dobre praktyki — zarządzanie procesami

Dobre praktyki zarządzania procesami opierają się na prostych regułach odniesionych do czterech procesów projektowych: planowania, projektowania, kodowania i testowania. Koncentrują się na procesach: maksymalnego uproszczenia realizowanych zadań, bezpośredniej komunikacji członków zespołu i zapewnienia mechanizmów sprzężenia zwrotnego w ocenie powstających produktów. Koncentrują się także na odwadze w formułowaniu wizji systemu oraz wspólnej odpowiedzialności za realizowane prace.

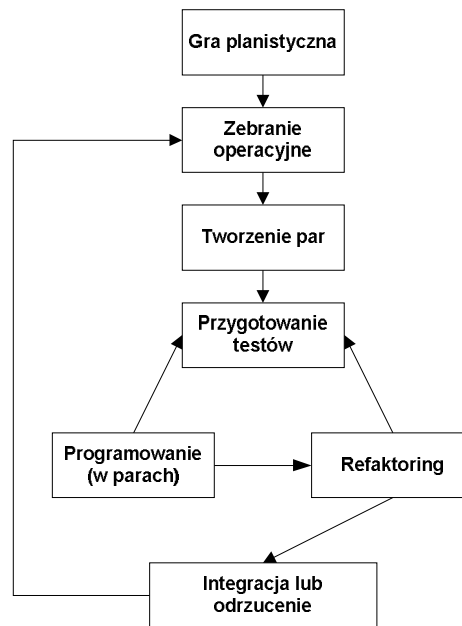
Maksymalne uproszczenia realizowanych zadań oznacza eliminację wszelkich zbędnych czynności podczas realizacji projektu oraz możliwie duże uproszczenie kodu źródłowego. Cechą charakterystyczną związaną z tą dobrą praktyką jest podejście refaktoryzacji, czyli poprawiania oprogramowania metodą małych przyrostów.

Drugą dobrą praktyką sugeruje położenie największego nacisku podczas realizacji zadań na bezpośrednią (werbalną) komunikację wszystkich członków zespołu. Zaleca się w tym celu krótkie spotkania operacyjne (podobnie jak w metodzie SCRUM), których celem jest porządkowanie bieżących spraw i zespołowe rozwiązywanie problemów.

Trzecia dobra praktyka dotyczy zapewnienia mechanizmu sprzężeń zwrotnych w procesach oceny powstających produktów. Dzięki takiemu podejściu możliwe jest szybkie poprawienie ewentualnych błędów z udziałem klienta. Podstawą dla takich procesów powinny być testy akceptacyjne, które klient przygotowuje wraz z testerami.

Kolejną dobrą praktyką jest wartościowanie odwagi w formułowaniu wizji projektu oraz podejmowaniu działań przyrostowych. Praktyka ta dotyczy także potrzeby odrzucania części wykonanej pracy. W przypadku, jeżeli system nie spełnia oczekiwań klienta albo klient te oczekiwania zmienił, należy podjąć decyzję o odrzuceniu części wytworzonego kodu.

Ostatni z głównych fundamentów metody XP opiera się na zasadzie wspólnej odpowiedzialności za realizowane prace. Zaleca się zatem, aby tworzony kod programistyczny był na tyle przejrzysty, aby inne osoby były w stanie wykorzystać ten kod w dalszych pracach. Przykład procedury postępowania z zastosowaniem dobrych praktyk przedstawiono na rys. 1.6.



Rys. 1.6 Realizacja projektu wg metody XP

Źródło: opracowanie własne na podstawie [102]

Dobre praktyki — zarządzanie zespołem

Metoda XP zawiera także szereg dobrych praktyk związanych z zarządzaniem zespołem. O tym, że udział zespołu w podejmowaniu decyzji jest znaczny, zostało już powiedziane w poprzedniej części poświęconej dobrym praktykom odniesionym do procesów realizacji projektu. Dobre praktyki dotyczące organizacji zespołu sugerują uwzględnienie dwóch nadrzędnych ról: programisty oraz klienta. Nie mówi się zatem o żadnej złożonej strukturze organizacyjnej (jak to było w metodzie PRINCE2), ale o równorzędnych członkach zespołu. Każdy z nich ma prawo głosu przy podejmowaniu decyzji odnośnie realizacji określonych prac projektowych.

Należy zaznaczyć, że niezwykle ważne jest traktowanie klienta jako członka zespołu. Według dobrych praktyk klient powinien stale współpracować z zespołem. Rolą klienta jest specyfikowanie wymagań w stosunku do systemu poprzez przygotowanie odpowiedniej jego metafory dotyczącej wytwarzanego oprogramowania.

Oprócz tych dwóch głównych ról, dobrą praktyką jest powołanie ról pomocniczych, takich jak tester (którego zadaniem jest przygotowanie skryptów testowych podczas rozmów z klientem), osoby czuwającej nad poprawnym przebiegiem prac i pomagającej rozwiązywać problemy (ang. *coach*) (odpowiednik Mistrza Młyna w metodzie SCRUM) oraz osoby odpowiedzialnej za analizę postępów prac i monitorowanie przebiegu projektu (ang. *tracker*) [85].

Dobłą praktyką charakterystyczną dla metody XP PROGRAMMING jest programowanie parami. Utworzenie par wpływa na jakość wytwarzanego oprogramowania, gdy podczas programowania jedna osoba (zwana autorem) tworzy kod, a druga (recenzent) go weryfikuje. Praca parami ma również wymiar integracyjny. Warto nadmienić również, że metoda XP zakłada wymiennosc osób w ramach par.

Dobre praktyki — Wytwarzanie dokumentacji

Dobre praktyki związane z dokumentacją zawarte w metodzie XP sugerują brak wytwarzania dokumentacji. Stosowanie procesów refaktoringu, upraszczania kodu i programowanie parami powinny zapewnić czytelny i przejrzysty kod programu (uzupełniony dokumentacją w formie komentarzy). Jedyne dokumenty, jakie zaleca się w metodzie XP, to małe, zazwyczaj pojedyncze kartki służące spisaniu oczekiwań klienta podczas gry planistycznej. Nie stanowią one jednak scalonej formy dokumentacji.

Wybór dobrych praktyk

Po dokonaniu analizy metody XP można przystąpić do zestawienia tych dobrych praktyk, które mogą być istotne z punktu widzenia podejmowanych decyzji przez kierownika projektu informatycznego i które mogą przyczynić się wyodrębnienia zagregowanych zmiennych decyzyjnych.

Wybrane dobre praktyki zestawiono w tabeli 1.8, prezentując także kontekst ich zastosowania.

Tab. 1.8 Dobre praktyki wg metody XP

DOBRA PRAKTYKA	UZASADNIENIE
Rozpoczęcie pracy od wymagań klienta	Pozwala na włączenie klienta do pracy zespołu oraz na szybką weryfikację przez klienta wykonanych działań.
Zastosowanie programowania parami	Pozwala na dzielenie się wiedzą oraz wzajemną kontrolę tworzonego kodu programowego.
Realizowanie zadań w oparciu o testy	Pozwala na realizację tylko tych funkcjonalności, które są niezbędne.
Planowanie krótkich przebiegów, zaczynając od najprostszycz działań	Pozwala na częstą weryfikację wytwarzanego oprogramowania oraz na rozwijanie produktu projektu informatycznego oddolnie.
Zawieranie dokumentacji w kodzie w formie komentarzy	Pozwala na tworzenie przejrzystego kodu, czytelnego dla pozostałych członków zespołu.
Tworzenie metafory systemu	Pozwala na przełożenie oczekiwań klienta w sposób zrozumiały dla wszystkich.

Źródło: opracowanie własne na podstawie [85,102]

Ocena metody

Dobre praktyki zawarte w metodzie XP podkreślają znaczenie programisty w procesach wytwarzania i zarządzania projektem. Oznacza to, że koncentrują się one na procesach twórczych celem ich maksymalnego uproszczenia. Jeżeli założymy, że ograniczone jest także wytwarzanie dokumentacji, pojawia się pytanie, w jaki sposób zarządzać projektem w tak ograniczonych warunkach. Odpowiedzią na tak postawione pytania jest wskazanie, że możliwe jest to pod warunkiem odpowiedniego poziomu dojrzałości zespołów realizujących projekty z wykorzystaniem dobrych praktyk metody XP oraz w sytuacji niewielkich projektów realizowanych z zastosowaniem tych praktyk.

Ważne staje się podkreślenie znaczenia roli klienta. Klient i jego rola w procesach wytwarzania oprogramowania była oznaczana wielokrotnie w uprzednio omawianych dobrych praktykach. W tym przypadku jednak klient jest przeciwstawiony programiście i wskazane jest środowisko współpracy obu partnerów. Środowiskiem tym są: metafora systemu oraz małe wydania produktów wytwarzanego systemu informatycznego. Podkreśla się także znaczenie dobrych praktyk dotyczących komunikacji partnerów projektowych. Oznacza to, że porządkując role w zespole i traktując je z perspektywy programisty lub klienta można tworzyć metafory systemu i małe wydania produktów. Takie podejście oznacza także, że można tworzyć odpowiednie warunki dla procesów zarządzania projektami. Ten widziany z perspektywy XP ogranicza liczbę ról, ale jednocześnie dostarcza odpowiednie mechanizmy komunikacji.

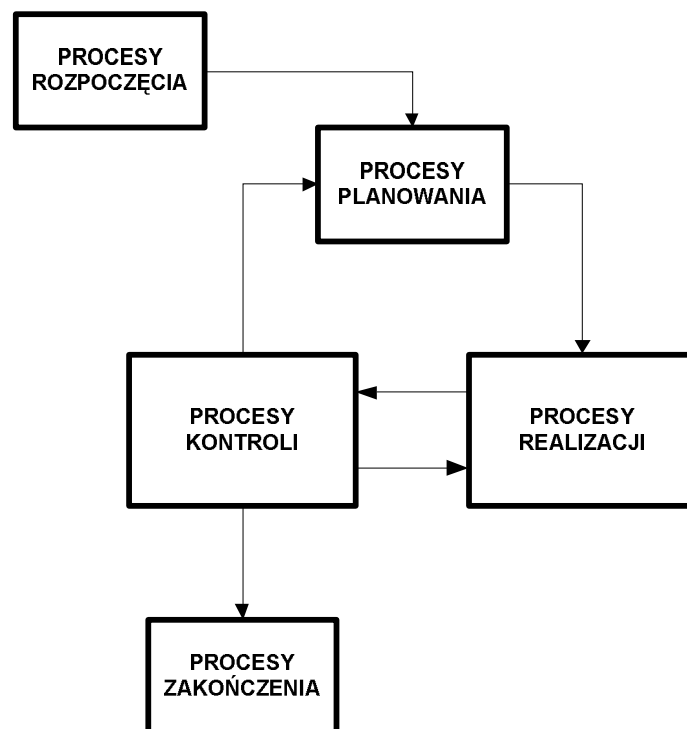
Analiza dobrych praktyk XP wskazuje również na możliwości analizy stanu projektu poprzez jego ocenę w perspektywie małych wydań systemu. Jeżeli duży system może być wytwarzany w oparciu o znaczną liczbę małych iteracji, to proces wytwarzania może być w całości kontrolowany na poziomie klienta i wytwórcy. Stąd też procesy kontroli projektu, stanu dostawcy i klienta wskazują na stan wykorzystania dobrych praktyk zarządzania projektem.

1.2.5 Przykład zbioru dobrych praktyk — PMBoK

PMBoK (ang. *Project Management Body of Knowledge*) stanowi kompendium wiedzy o zarządzaniu projektami. Kompendium to zawiera zbiór dobrych praktyk o zarządzaniu dowolnymi projektami. Stanowi więc ważne źródło wiedzy o projektach i dobrych praktykach stosowanych w zarządzaniu projektami i dlatego jest przedmiotem analizy podjętej w tej pracy. Stanowi także uzupełnienie wiedzy o metodach zarządzania projektami prezentowanych uprzednio. Dlatego też, podobnie jak to miało miejsce w przypadku omawiania metod, dobre praktyki zostaną zgrupowane dla wyodrębnienia tych, które stanowiąc będą podstawę do klasyfikacji zagregowanych zmiennych decyzyjnych.

Dobre praktyki — zarządzanie procesami

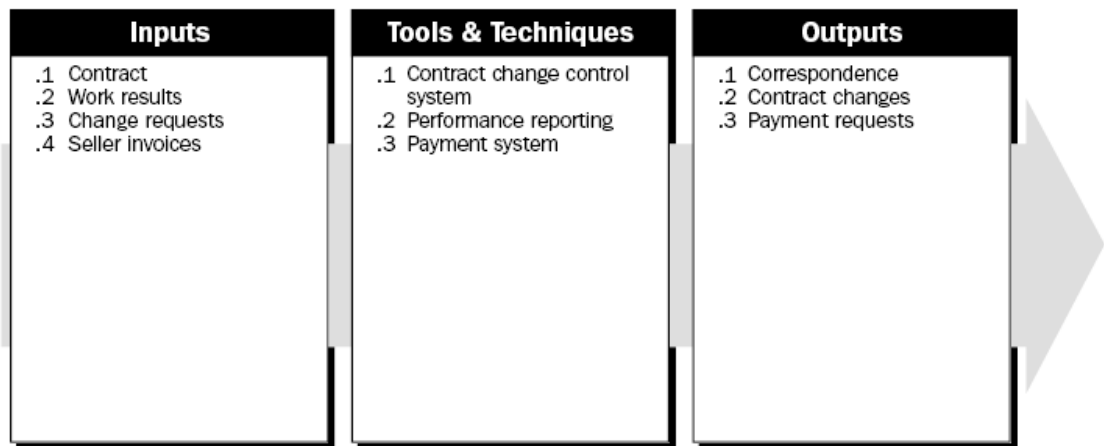
W ramach dobrych praktyk PMBoK przyjmuje się, że projekt jest dekomponowany na grupy procesów, pomiędzy którymi dodatkowo zachodzą interakcje. Na rys. 1.7 przedstawiono zależności pomiędzy poszczególnymi procesami w realizowanych projektach.



Rys. 1.7 Realizacja projektu wg PMBoK

Źródło: opracowanie własne na podstawie [71]

Każda z przedstawionych na rysunku grup procesów składa się z pojedynczych procesów także opisanych w PMBoK. Przyjmuje się także, że każdy proces jest rozpatrywany na trzech płaszczyznach: wejścia (które stanowiąc powinny odpowiednie dokumenty, plany), narzędzia i techniki (odpowiednie mechanizmy przetwarzania wejść) oraz wyjścia (dokumenty oraz produkty).



Rys. 1.8 Przykładowa wizualizacja procesu w PMBoK

Źródło: [71]

Zaprezentowane podejście do opisu procesów prezentowane na rys. 1.8 jest zbliżone do omawianego uprzednio podejścia RUP, tyle że w przypadku RUP mówiono o zadaniach i produktach pracy na płaszczyźnie wejścia i wyjścia. W przypadku PMBoK każdą aktywność, każdy obszar realizacji projektu traktuje się jako proces. W PMBoK zawarto 42 podstawowe procesy zestawione w pięciu grupach procesów: inicjowania, planowania, realizacji, kontrolowania i zamknięcia.

Tab. 1.9 Zalecenia do konkretnych grup procesów w PMBoK

GRUPA PROCESÓW	WYBRANE ZALECENIA
Procesy rozpoczęcia	PMBok zaleca kierownikom projektów rozpatrywanie procesów rozpoczęcia w odniesieniu do opracowywania dokumentacji otwarcia (dokument, w oparciu o który definiuje się wszystkie zasady, jakimi kierować się będzie kierownik w projekcie). Zaleca się również opracowanie wstępnego zakresu projektu. Efektem procesów rozpoczęcia powinno być zatwierdzenie projektu do realizacji.
Procesy planowania	Kierownicy projektów powinni wykonać szereg działań planistycznych pozwalających określić, w jaki sposób będzie realizowany projekt, oraz określić niezbędne do realizacji zasoby. Dlatego do podstawowych procesów planowania PMBoK zalicza m.in. opracowanie planu zarządzania projektem, definiowanie zakresu, określenie przepływów pracy, szacowanie czasów trwania zadań, określenie harmonogramu, budżetowanie czy identyfikacja ryzyka. Efekty procesów planowania powinny przybliżać projekt do realizacji.
Procesy realizacji	PMBok zaleca kierownikom projektów koncentrację na właściwym wykorzystaniu uprzednio zaplanowanych zasobów (ludzkich, rzeczowych, finansowych) oraz stałą kontrolę jakości wytwarzanego produktu (np. oprogramowania w projektach informatycznych). Stąd też w ramach procesów realizacji zaleca się kierownikom, aby koncentrowali się na takich procesach jak zapewnianie jakości, rozwój zespołu, zapewnienie obiegu informacji, zarządzanie zadaniami.
Procesy kontroli	Zaleca się również kierownikom, aby wszelkie prowadzone prace w ramach projektu poddawane były kontroli. W PMBoK mówi się również o odpowiednim reagowaniu na odchylenia od założeń. Do procesów kontroli zalicza się m.in. monitorowanie i nadzór prac, weryfikowanie zakresu, zarządzanie zmianami, kontrola jakości, nadzór budżetu, raportowanie postępów prac itp.
Procesy zakończenia	Zaleca się kierownikom dopilnowanie po zrealizowaniu projektu przygotowania odpowiednich dokumentów zamykających projekt. Wskazuje się również na wykonanie działań zamykających umowę z klientem. Procesy zakończenia mogą zatem obejmować przygotowanie końcowego rozliczenia, protokołów przekazania itp.

Źródło: opracowanie własne na podstawie [71]

Dobre praktyki — zarządzanie zespołem

W odróżnieniu od prezentowanych uprzednio metod zarządzania projektami, PMBoK nie definiuje precyzyjnie składu zespołu projektowego. Zawarte są dobre praktyki dotyczące doboru składu zespołu, jednak pozostawia się pełną swobodę w konstruowaniu zespołów kierownikowi projektów. Stosunkowo dużo dobrych praktyk odniesionych jest do zarządzania zespołem przez jego kierownika. Kierownik projektu odpowiada za większość procesów zachodzących w trakcie prac projektowych. Ponadto w PMBoK dobre praktyki odniesione są do dobru członków zespołu do pełnienia określonych ról w projektach (np. kierowników zespołów, interesariuszy). Z kolei brak jest dobrych praktyk dla definiowania ról w zespołach projektowych.

Dokumentacja

W odróżnieniu od XP PROGRAMMING dobre praktyki PMBoK wskazują na dokumentację jako nieodłączny składnik każdego projektu. Brak jest jednak dobrych praktyk dotyczących jej przygotowywania. Zawarte są natomiast zalecenia, jakie dokumenty mogą być potrzebne na poszczególnych etapach projektu traktowanych jako dokumentacja wejściowa i wyjściowa w określonych procesach zachodzących podczas trwania projektu.

Wybrane dobre praktyki dla kierowników

Na podstawie przeprowadzonej analizy zestawu dobrych praktyk PMBoK wybrano te, które są istotne z punktu widzenia decyzji podejmowanych przez kierownika projektu informatycznego i które mogą przyczynić się wyodrębnienia zagregowanych zmiennych decyzyjnych.

Tab. 1.10 Dobre praktyki w PMBoK

DOBRA PRAKTYKA	UZASADNIENIE
Ustalenie wejścia i wyjścia do/z każdego procesu	Pozwala na doprecyzowanie oczekiwanych parametrów wejściowych i wyjściowych w realizacji, przez co wszyscy uczestnicy od początku wiedzą, jakie są spodziewane efekty pracy.
Definiowanie wszystkich procesów realizacji projektu	Pozwala określić precyzyjnie, jakie procesy powinny zostać uruchomione w projekcie w celu wytworzenia oczekiwanego produktu.
Definiowanie sposobów przetwarzania wejść do zadań	Pozwala określić zapotrzebowanie na określone metody i narzędzia (technologie) używane do realizacji zadań w projekcie.
Ustalenie odpowiedzialności za każdy proces	Pozwala przyporządkować każdemu działaniu osobę odpowiedzialną celem zapewnienia mechanizmu kontroli.
Przygotowanie dokumentacji z realizacji projektu	Pozwala na zapis doświadczeń związanych z realizacją określonego projektu
Koncentrowanie działania na harmonogramie i budżecie	Pozwala na podejmowanie działań związanych wyłącznie z podstawowymi ograniczeniami projektowymi (harmonogram, budżet, zakres).

Źródło: opracowanie własne na podstawie [71]

Ocena

Po dokonaniu analizy dobrych praktyk zawartych w PMBoK warto zwrócić uwagę na ich praktyczny charakter. Porządkują one terminologię projektową i dają kierownikom projektów bardzo solidne podstawy wiedzy z zakresu zarządzania projektami. Z drugiej strony ich liczba, procesy i ich

podział powodują, że kierownicy projektów mogą mieć problem z wybraniem tych praktyk, które są przydatne dla realizowanych własnych projektów.

Znaczna liczba dobrych praktyk zawarta w PMBoK wskazuje także na problem dobru dla potrzeb własnych projektów tych najważniejszych. Można w oparciu o PMBoK stworzyć repozytorium projektu, a następnie wprowadzić te dobre praktyki do odpowiednich baz wiedzy. Pojawia się jednak pytanie, na ile te dobre praktyki będą dostępne w sytuacjach, kiedy kierownik projektu potrzebuje jednej kluczowej wytycznej w konkretnej sytuacji decyzyjnej. Stąd też należy zastanowić się, czy w przypadku PMBoK nie przekroczono rozsądnej granicy dostępności tych dobrych praktyk lub też czy nie należy dekomponować tych dobrych praktyk pomiędzy członków zespołu projektowego, tak aby z jednej strony rozwijać zespół, a z drugiej wspólnie podejmować decyzje zarządcze.

Problem wykorzystania PMBoK i jego dobrych praktyk sprowadza się także do jego wykorzystania jako zbioru dobrych praktyk, a nie metody zarządzania projektem. Podejście, w którym zarządza się projektem z wykorzystaniem PMBoK, jest typowe dla dojrzałych kierowników projektów. Dysponując szeroką wiedzą o zarządzaniu projektami, wybierają te praktyki, które są niezbędne w danej sytuacji decyzyjnej. Takie podejście oznacza, że można zarządzać projektem informatycznym tylko w oparciu o dobre praktyki, ale przy wysokim poziomie dojrzałości zespołu dostawcy.

1.3 Podsumowanie — Stan badań projektów informatycznych

Z przeprowadzonego w niniejszym rozdziale przeglądu metod zarządzania projektami informatycznymi oraz z omówionych przykładów rzeczywistych projektów wynika, że stosowanie metod zarządzania projektami nie gwarantuje powodzenia projektu informatycznego. Podczas omawiania specyfiki projektów informatycznych zwrócono szczególną uwagę na ich dużą dynamikę i zmienność (zwłaszcza w obszarze wymagań klienta). Dynamika ta powoduje, że dobór metody zarządzania projektem jest jednym z podstawowych problemów decyzyjnych kierowników. Niewłaściwe stosowanie metod zarządzania projektami powoduje, że projekt kwalifikuje się jako nieudany z powodu nie dotrzymania przynajmniej jednego z klasycznych ograniczeń projektowych.

Przeprowadzona dekompozycja metod zarządzania projektami informatycznymi prowadząca do wyodrębnienia dobrych praktyk wydaje się ważnym działaniem w kierunku poszukiwania przyczyn niepowodzeń projektów informatycznych. Skoro bowiem stosowanie metody zarządzania projektem nie gwarantuje jego sukcesu, warto spróbować łączyć dobre praktyki z różnych metod, odnosząc je do specyfiki projektu. Zastosowana w tym rozdziale dekompozycja metod do poziomu dobrych praktyk stwarza warunki do diagnozy stanu projektów, ale też stwarza warunki do diagnozy zmiennych decyzyjnych tak ważnych w realizacji projektów.

Ocena rynku technologii informatycznych pozwala zwrócić uwagę, że obecnie dostarcza on szeregu narzędzi wspomagających metody zarządzania projektami informatycznymi. Oferowane narzędzia do zarządzania projektami informatycznymi posiadają funkcjonalności wspomagające dobre praktyki nakierowane na prace planistyczne, jak również umożliwiające organizowanie zadań projektowych czy też kierowanie zespołem. Warto zaznaczyć także, że istnieją narzędzia, które nieodłącznie związane mogą być z konkretną metodą zarządzania projektami informatycznymi (jak np. IBM Rational Method Composer, który wspiera metodę RUP) albo stanowią rozwiązania o charakterze kompleksowym (jak Microsoft Project, który wspomaga dowolny rodzaj projektu). Poza ofertą koncernów informatycznych kierownicy mają do dyspozycji dodatkowo szereg narzędzi oferowanych na zasadzie *Open Source* (portal 4pm.pl pozwalał na darmowe użytkowanie kilkuset darmowych narzędzi do budowania harmonogramów projektów).

Podstawową kategorią dla narzędzi wspomagających kierowników projektów są narzędzia dedykowane kierownikom projektów informatycznych. Zadaniem takich narzędzi jest zapewnienie

kierownikowi projektu wsparcia w realizacji zadań planistycznych, organizacyjnych oraz kontrolnych. Takie narzędzia wspomagają szczególnie kierownika w prowadzeniu określonych działań w projekcie informatycznym. Pozwalają np. zaplanować zadania i je opisać, zdefiniować zespół poprzez przyporządkowanie ról do uczestników albo pozwalają stworzyć strukturę przepływu pracy (ang. *Work Breakdown Structure*). Narzędzia wspomagające kierowników pozwalają także najczęściej na przygotowanie stosownej dokumentacji projektowej oraz usprawniają obieg dokumentów poprzez dostępność repozytorium projektowego. Przykładem takiego narzędzia jest IBM Rational Method Composer, który umożliwia kierownikom definiowanie własnych projektów opartych na dobrych praktykach metody RUP.

Drugą kategorią narzędzi wykorzystywanych w realizacji projektów są narzędzia wspomagające wszystkie procesy związane z wytwarzaniem produktu informatycznego. Z racji tego, że produktem jest przeważnie oprogramowanie, narzędzia wspierające wytwarzanie posiadają szereg funkcjonalności zapewniających realizację zadań programistycznych. Stąd też takie narzędzia pozwalają na wytwarzanie oprogramowania, zarządzanie wersjami, kontrolę zmian w wytwarzanym oprogramowaniu.

Ocena metod zarządzania projektami informatycznymi, zbioru dobrych praktyk będących efektem dekompozycji tych metod oraz narzędzi informatycznych wskazuje, że problem zarządzania projektami informatycznymi może sprowadzać się do potrzeby wyodrębnienia innych niż dotychczas zmiennych decyzyjnych kierownika projektu. Przeprowadzona dekompozycja miała na celu wybór takich dobrych praktyk, których celem będzie wspomaganie wyboru tych zmiennych. Kolejny rozdział pracy poświęcony jest procesom identyfikacji tych zmiennych dla potrzeb proponowanego w pracy modelu systemu agentowego.

Rozdział II

Identyfikacja zmiennych decyzyjnych w projektach informatycznych

Przeprowadzona w poprzednim rozdziale analiza dobrych praktyk zarządzania projektami informatycznymi wynikająca z dekompozycji metod zarządzania projektami informatycznymi wykazała brak ich dopasowania do istniejącego do tej pory trójkąta ograniczeń. Oznacza to, że należy poszukiwać nowych zmiennych decyzyjnych, w oparciu o które kierownicy projektów dokonywać będą ich doboru do realizowanych projektów informatycznych. Podstawowym pytaniem, na które należy odpowiedzieć w tym rozdziale, jest to, czy mimo że każdy projekt jest z definicji inny (cecha niepowtarzalności projektów), istnieje taki zestaw zmiennych. Jeżeli tak, należy ten zestaw określić oraz przeprowadzić badanie dopasowania dobrych praktyk do nowego trójkąta ograniczeń zmiennych decyzyjnych.

Do odpowiedzi na tak postawione pytanie wykorzystano kompendium wiedzy zaprezentowanej w poprzednim rozdziale. Zaobserwowano bowiem po analizie wybranych dobrych praktyk z każdej z metod zarządzania, że pewne dobre praktyki są typowe dla wszystkich rodzajów projektów informatycznych. Praktyki te można agregować w ramach pewnych kategorii, które w tym rozdziale zostały określone mianem zagregowanych zmiennych decyzyjnych. To właśnie w oparciu o te zagregowane zmienne decyzyjne menedżerowie powinni dobierać metody zarządzania projektami lub poszczególne dobre praktyki, a następnie realizować projekt w oparciu o klasyczny trójkąt ograniczeń (zakres, budżet, harmonogram). Stąd też nadrzędnym celem niniejszego rozdziału jest zidentyfikowanie i opisanie takich zmiennych decyzyjnych. Na podstawie tak przeprowadzonej analizy wyodrębniono trzy zagregowane zmienne decyzyjne. Dobrane zmienne poddano procesom kwantyfikacji dla potrzeb menedżerów projektów oraz przeprowadzono procesy dopasowania wybranych w uprzednim rozdziale dobrych praktyk do nowego trójkąta ograniczeń zmiennych decyzyjnych.

2.1 Zmienne decyzyjne w projekcie informatycznym

Przeprowadzona w rozdziale I charakterystyka projektów informatycznych, metod zarządzania nimi oraz wykorzystywanych przez kierowników projektów dobrych praktyk wykazała, że jednym z podstawowych problemów, przed jakimi stają kierownicy projektów, jest wybór odpowiedniej metody lub odpowiedniego zestawu dobrych praktyk zarządzania takim projektem. Z punktu widzenia realizacji projektu wybór metody zarządzania jest kluczowy dla poprawnego przebiegu projektu. Wydaje się, że projekt informatyczny uznaje się za zakończony sukcesem pod warunkiem spełnienia klasycznych ograniczeń projektowych (zrealizowanie projektu w ramach harmonogramu, budżetu oraz w pełnym zakresie). Zgodzić się można z tak postawioną tezą jednak tylko częściowo. Dlatego, że klasyczny trójkąt ograniczeń powstaje w następstwie wcześniejszych działań i decyzji prowadzonych przez kierownika projektu. Zakres projektu określany jest najczęściej na podstawie rozmów z klientem, harmonogram ustala się na podstawie oceny możliwości zasobowych firmy realizującej projekt informatyczny (dostawca), budżet jest najczęściej wynikiem oceny pracochłonności zespołu realizującego projekt po stronie dostawcy oraz możliwości finansowych klienta. Stąd też można wysnuć wniosek, że zanim kierownik projektu zacznie podejmować decyzje odnośnie realizacji

projektu w ramach zmiennych klasycznego trójkąta ograniczeń, konieczny staje się proces analizy wstępnej tych zmiennych.

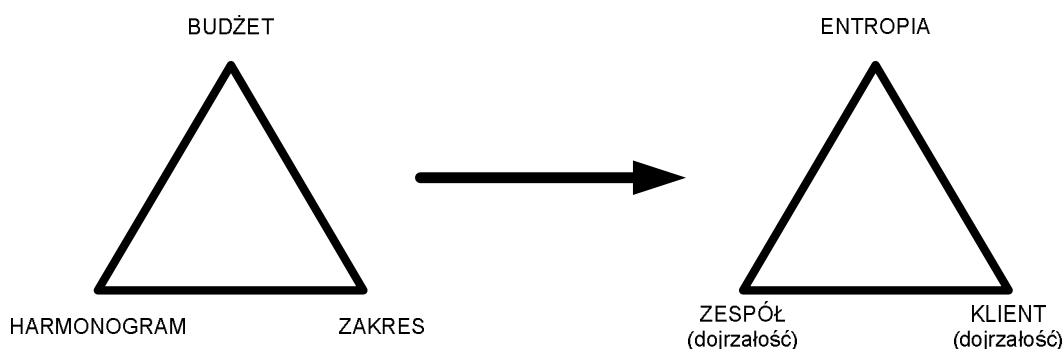
Biorąc pod uwagę dobre praktyki zawarte w metodach zarządzania projektami (zaprezentowane w uprzednim rozdziale) oraz konieczność uzupełnienia klasycznego trójkąta ograniczeń dodatkowym zestawem zmiennych decyzyjnych, wyodrębniono trzy obszary projektu niezbędne do analizy przed jego rozpoczęciem.

Pierwszy z tych obszarów obejmuje procesy klienta w projektach informatycznych. Klient jest najczęściej głównym odbiorcą produktu powstałego w wyniku projektu informatycznego. Wcześniejsza dekompozycja metod zarządzania projektami na dobre praktyki zwróciła uwagę na szereg wskazówek i zaleceń związanych z obecnością klienta, w wielu wypadkach włączając go w procesy wytwórcze i zarządcze (SCRUM, XP PROGRAMMING), zalecając także jego stałą obecność w projekcie. Oznacza to zatem, że bez dogłębnego poznania stanu klienta kierownikowi może być trudno podejmować decyzje dotyczące sposobu prowadzenia projektu. Jeżeli bowiem klient będzie często zmieniał wymagania, stosowanie metod opartych na sztywnym planowaniu i procesach może okazać się niewydajne. W dalszej części niniejszego rozdziału przedstawione zostaną sugestie dotyczące określania stanu klienta w projekcie oraz analiz, jakie powinien przeprowadzić kierownik projektu, aby podjąć właściwe decyzje związane z doбором dobrych praktyk zarządzania projektem w zależności od stanu klienta. Wszystkie czynniki związane z analizą stanu klienta określono mianem zagregowanej zmiennej decyzyjnej — **DOJRZAŁOŚĆ KLIENTA**.

Drugim obszarem procesów decyzyjnych omawianych szeroko we wszystkich metodach zarządzania projektami są procesy zespołowe organizacji dostawcy. To zespół dostawcy pod nadzorem kierownika projektu odpowiada za właściwy przebieg prac, za to jak i kiedy wykonywane są określone prace. Wydaje się zatem słuszne, aby kierownik projektu przed rozpoczęciem realizacji projektu dobrze rozpoznał możliwości realizacji projektu przez zespół, którym kieruje. Rozpoznanie to powinno odbywać się pod kątem kompetencji projektowych, ale również pod kątem kompetencji związanych z pracą grupową. Taka analiza pozwoli kierownikowi projektu na właściwe przydzielenie każdemu pracownikowi odpowiedniej roli w projekcie. Właściwy przydział ról zwiększa wydajność prowadzonych prac. Biorąc pod uwagę również to, że decyzje odnośnie doboru metody zarządzania projektami informatycznymi albo wyboru dobrych praktyk powinny być uzależnione od stanu zespołu dostawcy (o czym była mowa w rozdziale I), uznano, że kolejną zagregowaną zmienną decyzyjną, jest **DOJRZAŁOŚĆ ZESPOŁU DOSTAWCY**.

Trzeci obszar procesów decyzyjnych wiąże się z decyzjami kierowników projektów odniesionych do realizowanego projektu informatycznego, a dotyczących oceny jego złożoności, zmienności wymagań oraz niepewności otrzymania końcowego produktu. W przypadku większości projektów informatycznych realizowanych w sposób przyrostowy wymagania się zmieniają się w czasie, a ostateczny produkt nie jest dokładnie określony, stąd precyzowanie zadań i procesów wytwarzania jest trudne. Brak wystarczającej wiedzy o stanie projektu implikuje niepewność kierownika, utrudniając podejmowanie decyzji planistycznej (krótko- jak i długofalowych). W związku ze zjawiskiem niepełnego (niewystarczającego) poziomu informacji w projekcie informatycznym kierownicy projektów powinni dopasowywać takie dobre praktyki do bieżącego stanu projektu. Uznano zatem, że bieżący stan projektu należy zawrzeć w zmiennej decyzyjnej nazwanej **ENTROPIĄ PROJEKTU**.

Wyodrębnienie tych trzech zagregowanych zmiennych decyzyjnych wydaje się kluczowe z punktu widzenia doboru dobrych praktyk zarządzania przez kierowników projektów informatycznych do swoich projektów przed ich realizacją w ramach klasycznego trójkąta ograniczeń.



Rys. 2.1 Nowy trójkąt ograniczeń a klasyczny trójkąt ograniczeń

Źródło: opracowanie własne

Nowy trójkąt ograniczeń prezentowany na rys. 2.1 stanowi rozszerzenie istniejącego trójkąta ograniczeń projektowych i umożliwia wstępną analizę możliwości realizacji projektu. Aby taka analiza była możliwa oraz wykorzystywana była przez kierowników projektu, zostanie ona wsparta agentowym modelem zarządzania projektami informatycznymi, który jest celem tej pracy. Agentowy adaptacyjny model zarządzania projektami informatycznymi stworzy warunki do wstępnej analizy projektu, doboru dobrych praktyk i metod zarządzania projektami oraz zapewni wielowymiarową analizę zmiennych decyzyjnych w ramach nowego trójkąta ograniczeń. Tym samym rozszerzy dotychczasowe traktowanie projektów informatycznych jako trójwymiarowych przedsięwzięć (budżet, harmonogram, zakres).

2.1.1 Dojrzałość klienta w projekcie informatycznym

W związku z koniecznością uwzględnienia dojrzałości klienta jako zmiennej niezbędnej dla podejmowania decyzji kierowników projektów odnośnie doboru dobrych praktyk zarządzania, w tej części omówiono te stany klienta, które decydują o jego poziomie dojrzałości.

Klient jest podstawowym podmiotem każdego realizowanego przedsięwzięcia. Zgodnie z definicją struktury projektu klient jest podstawowym elementem tej struktury. Klient w projektach informatycznych postrzegany jest najczęściej jako osoba (fizyczna lub prawna) zamawiająca określony produkt projektu informatycznego. Nie zawsze klient jest tożsamy z użytkownikiem końcowym wytworzonego w projekcie informatycznym produktu. Nie ulega jednak wątpliwości, że rola klienta w projektach informatycznych jest nieodzowna. Przyjęło się bowiem traktować klienta jako podmiot inicjujący projekt (jak w metodach klasycznych) albo wręcz członka zespołu (jak w metodach lekkich). Jeżeli klientem jest podmiot prawny, to oczekuje się w projektach informatycznych oddelegowania odpowiedniej osoby lub grupy osób, która będzie reprezentować klienta w projekcie.

Rola klienta w projektach

Obecność klienta w projektach informatycznych może jednak przybierać różne formy. Odnosząc się do dobrych praktyk związanych z klientem zawartych w metodach lekkich (omówionych w pierwszej części pracy), zaleca się, aby klienta traktować jako członka zespołu, włączać do prowadzonych prac i zapewniać możliwie duży przepływ informacji pomiędzy członkami zespołu a klientem. Dzięki temu uzyskuje się szybką weryfikację tworzonego produktu przez osobę (podmiot) bezpośrednio zainteresowaną wynikami projektu. W metodach klasycznych stosunek do klienta jest nieco inny. Dobre praktyki zawarte w metodach klasycznych traktują bowiem klienta jako osobę

niezbędną do zainicjowania projektu, jako osobę, z którą tworzy się wizję projektu i ustala pewne ogólne założenia. Nie traktuje się jednak klienta jako podmiot bezpośrednio zaangażowany w prace projektowe.

Bez względu jednak na miejsce klienta lub przedstawiciela klienta w projektach informatycznych pewne jest to, że klient jest reprezentantem strony biznesowej i należy względem niego stosować wszelkie zasady gry rynkowej. Innymi słowy, działania prowadzone w ramach projektu powinny prowadzić do zaspokojenia oczekiwań klienta. Takie podejście (zgodne z zasadami orientacji pro klienckiej) adaptowane jest do dzisiejszych projektów informatycznych z pełną świadomością, co zaobserwować można również w dobrych praktykach zarządzania projektami omówionych w rozdziale II.

Warto w tym miejscu przypomnieć pewne dobre praktyki z metody XP, gdzie zakłada się, że codzienną pracę zaczyna się od spisania oczekiwań klienta, a ostateczny produkt próbuje się przełożyć na metaforę systemu, gdzie nie wymaga się od klienta wiedzy technicznej, a jedynie umiejętność wyartykułowania swoich oczekiwań. Takie podejście, związane z dążeniem do pełnego zrozumienia potrzeb klienta, jest niezbędne dla obu stron projektu. Wymusza jednak na kierowniku szereg decyzji związanych z dążeniem do usatysfakcjonowania klienta. Stąd też dostawca oprogramowania (produktu) powstałego w trakcie realizacji projektu informatycznego jest w stanie wykonać dobry produkt pod warunkiem, że oczekiwania klienta zostaną bardzo dobrze rozpoznane. Klient natomiast wtedy będzie usatysfakcjonowany, kiedy otrzyma produkt spełniający jego oczekiwania. Z uwagi na fakt, że klient nie musi znać specyfiki projektów informatycznych, albo wręcz nie znać się na aspektach technicznych (informatycznych), istnieje ogromna potrzeba zapewnienia spójności porozumiewania się strony wytwórczej i strony klienckiej. W dobrych praktykach zawartych w metodach zarządzania projektami (np. RUP) mówi się o poświęceniu szczególnej uwagi na przełożenie oczekiwań klienta na język zrozumiały dla zespołu wytwórczego. Takie działania nazywa się analizą biznesową, a odpowiedzialność za zrozumienie klienta powierza się analitykom biznesowym. Stąd też kierownik projektu zobowiązany jest podjąć stosowne decyzje odnośnie tego, jak prowadzić rozmowy z klientem i komu powierzyć gromadzenie wymagań pozyskiwanych od klienta. Bez względu jednak na to, jakie dobre praktyki zarządzania projektami zostaną zastosowane w projekcie, faktem jest, że wpływ klienta na realizację projektu jest bardzo istotny.

Wpływ klienta na projekt i decyzje względem klienta

Z punktu widzenia oceny dojrzałości klienta należy przyjrzeć się, jaka jest jego rola w projekcie informatycznym oraz jak obecność klienta wpływa na przebieg projektu. Obecność klienta wiąże się bowiem przede wszystkim z inicjowaniem projektu, ale również obecność klienta może mieć wpływ na przebieg całego projektu.

Włączanie klienta do prac związanych z projektem informatycznym zaczyna się od najczęściej od negocjacji biznesowych odnośnie tego, co ma być produktem projektu. Kierownicy projektów dokonują ustaleń finansowych z klientem, a także podejmują próbę określenia celu oraz zakresu projektu. Stąd też projekty informatyczne zaczynają się od zlecenia klienta, po czym następuje „wyciąganie wartości” [99] w oparciu o oczekiwania klienta określane mianem wymagań. Dzięki temu możliwe jest zdefiniowanie zakresu albo części zakresu projektu informatycznego.

Klient zatem jest ważnym ogniwem w procesie inicjowania projektu, ale również jego rola jest nieodzowna w określaniu wizji produktu, który ma zostać wytworzony w projekcie informatycznym. To poprzez rozmowę z klientem możliwe jest zdefiniowanie wymagań, które następnie zostaną zamienione w konkretne funkcjonalności systemu. Rolą kierownika projektu jest w tym wypadku podjęcie odpowiednich decyzji zapewniających jak najlepsze zrozumienie oczekiwań klienta.

Kierownik projektu podejmuje również szereg decyzji związanych z zarządzaniem relacjami z klientem. Pierwsza decyzja powinna dotyczyć udziału klienta w projekcie. Kierownik projektu

powinien zdecydować, czy udział klienta będzie stały, czy też będzie weryfikował prace w określonych odstępach czasu. Kolejne decyzje powinny dotyczyć, jak często powinno się klientowi raportować stan zaawansowania prac projektowych. Wreszcie kierownik powinien również sprawdzić, na ile klient jest skłonny do zmiany uprzednio zdefiniowanych wymagań, aby podjąć decyzje pozwalające radzić sobie w sytuacji zmienności wymagań. Z powyższego wynika, że kierownik projektu informatycznego powinien podjąć szereg decyzji związanych z obecnością klienta w projekcie informatycznym. Stąd też analiza stanu profilu klienta prowadząca do określenia jego dojrzałości pozwoli na podjęcie trafnych decyzji związanych z obecnością klienta w projekcie informatycznym.

Zdarzają się również w projektach informatycznych sytuacje, kiedy pewnych oczekiwań klienta nie jesteśmy w stanie zrealizować albo klient oczekuje czegoś, co z punktu widzenia analizy biznesowej jest mu niepotrzebne. W takich sytuacjach również kierownik projektu powinien podjąć stosowne decyzje prowadzące do przekonania klienta, że z niektórych funkcjonalności systemu należy zrezygnować. Bez znajomości stanu klienta kierownik projektu może podjąć decyzje niewłaściwe. Ostatnim zjawiskiem obserwowanym w projektach informatycznych odnoszącym się do stanu klienta jest to, że klient nie posiada najczęściej sprecyzowanych, skonkretyzowanych oczekiwań odnośnie produktu, który ma powstać w projekcie informatycznym. Klient potrzebuje „jakiegoś” rozwiązania informatycznego, ale nie jest w stanie określić parametrów tego rozwiązania. Klient nie potrafi zdefiniować dokładnie swoich potrzeb, co dodatkowo utrudnia przełożenie jego oczekiwań na wymagania zrozumiałe przez zespół dostawcy. Jest to sytuacja stosunkowo trudna zwłaszcza dla menedżera projektu. Po pierwsze bardzo trudno jest określić zakres projektu. Po drugie w trakcie realizacji projektu może okazać się, że klient zaczyna dopiero precyzować swoje oczekiwania, które okażą się odmienne od założonych na początku. Skutkuje to oczywiście koniecznością dokonywania zmian w projekcie. Zaprezentowana sytuacja obrazuje klienta o małej dojrzałości. Nie jest jednak winą takiego klienta, że brakuje mu znajomości zasad, jakimi kierują się projekty informatyczne. Rolą jednak kierownika jest przewidzieć takie zachowanie klienta poprzez przeprowadzenie odpowiedniej analizy jego dojrzałości.

Jak zatem rozumieć rolę klienta w przedsięwzięciu informatycznym? Na pewno jako kluczową dla wszystkich prowadzonych prac. Klient może zmienić wymagania, może zmodyfikować zestaw oczekiwanych funkcjonalności. Może również w trakcie projektu żądać zmian lub uznać, że realizowane prace nie pokrywają się z jego własną wizją danego rozwiązania informatycznego. Dodatkowo należy zaznaczyć, że zespół stający przed koniecznością realizacji projektu nie potrafi przewidzieć zachowania klienta. Niestety, każde takie zachowanie klienta może mieć wpływ na przebieg prac projektowych powodując zmiany w harmonogramie czy zakresie. Oznacza to zatem, że menedżer projektu powinien wziąć pod uwagę rolę klienta przystępując do planowania projektu. Oznacza to również, że zachowania klienta powinny być uwzględnione przy dobieraniu metody realizacji projektu i dobrych praktyk z nią związanych. Może okazać się, że w zależności od stanu klienta lepiej stosować dobre praktyki zawarte w metodach lekkich albo klasycznych. Stąd też wniosek, aby kierownik, podejmując decyzje odnośnie doboru dobrych praktyk zarządzania projektem, uwzględniał stan klienta.

Potrzeba analizy stanu klienta

Biorąc pod uwagę przedstawioną powyżej charakterystykę klienta w projektach informatycznych oraz uwzględniając jego wpływ i znaczenie dla przebiegu/realizacji projektu informatycznego, uznano, że analiza stanu klienta jest niezbędna kierownikowi projektu informatycznego z punktu widzenia podejmowania decyzji odnośnie dobierania dobrych praktyk zarządzania projektem informatycznym.

W sytuacji, w której klient nie potrafi przedstawić swoich wymagań precyzyjnie, istnieje duże prawdopodobieństwo, że w trakcie trwania projektu będzie chciał zmienić część ze swoich wymagań. Stosowanie sztywnych planów zalecanych w metodach klasycznych mogłoby znacznie skomplikować realizację takiego projektu. Każda bowiem zmiana wymagań implikowałaby modyfikację całego uprzednio zaplanowanego procesu realizacji projektu. Rolą kierownika jest zatem przyjąć takie dobre praktyki zarządzania projektem, które pozwolą sprostać zmienności wymagań. Stąd też wydaje się, że zanim kierownik projektu podejmie decyzje odnośnie dobrych praktyk zarządzania projektem i zanim zdecyduje, w jaki sposób prowadzić projekt, powinien dokonać analizy stanu klienta, określając wszystkie jego cechy niezbędne dla podjęcia odpowiednich decyzji co do sposobu prowadzenia projektu (czyli wybrania metody lub dobrych praktyk zarządzania projektem). Szereg cech klienta, które mogą mieć wpływ na decyzje kierownika odnośnie dopasowania dobrych praktyk do zarządzania projektem informatycznym, postawiono zagregować w zmiennej nazwanej DOJRZAŁOŚCIĄ KLIENTA.

Wcześniejsze rozważania pokazały, jak ważna jest rola klienta w projektach informatycznych. Wykazano również, że klient jest w stanie wywierać (świadomie bądź nieświadomie) wpływ na przebieg projektu, na sposób organizacji zadań, a również na zakres realizowanych prac. Zadaniem kierownika jest natomiast określenie (analiza) stanu klienta określanego poziomem jego dojrzałości.

Definicja 2.1

Dojrzałość klienta to zestaw cech klienta oraz jego kompetencji umożliwiających wspólną (wraz z zespołem dostawcy) realizację projektu. Ocena dojrzałości umożliwia kierownikom projektów dobór odpowiednich dobrych praktyk zarządzania dla potrzeb wspólnie realizowanego projektu

Uznano, że dokonując analizy dojrzałości klienta należy zwrócić uwagę na określone zestawy cech i kompetencji, jakie posiada klient. Podzielono te cechy i kompetencje na dwie kategorie — kompetencje merytoryczne związane ze znajomością specyfiki projektów informatycznych oraz cechy związane z umiejętnościami współpracy. Taki podział wynika z faktu, że klient przystępujący do projektu informatycznego nie musi orientować się w metodach zarządzania projektami informatycznymi, nie musi znać technologii informatycznych, a przy tym może mieć precyzyjnie określone wymagania dotyczące wytworzenia oprogramowania. Taki zestaw kompetencji klienta związanych z merytorycznymi aspektami projektu informatycznego określono mianem ODPOWIEDNIOŚCI KLIENTA. Natomiast wszystkie predyspozycje osobowościowe oraz umiejętności współpracy w trakcie realizacji projektu określono mianem DOPASOWANIA KLIENTA. Zaprezentowana ocena dojrzałości klienta bazuje na pracach [49,50].

Do kompetencji merytorycznych klienta określających poziom jego ODPOWIEDNIOŚCI można zaliczyć poziom wiedzy o realizowanym projekcie, umiejętność artykułowania wymagań, znajomość branży IT, znajomość specyfiki projektów informatycznych oraz ocena możliwości zmiany stawianych przez siebie wymagań.

Pierwszą kompetencję, którą powinni zbadać kierownicy projektów informatycznych u swoich klientów, jest znajomość branży informatycznej. Można tutaj zapytać klienta, na ile zorientowany jest w nomenklaturze stosowanej w projektach informatycznych. Sprawdzić, czy klient zna rodzaje systemów informatycznych, czy posiada wiedzę z zakresu technicznych aspektów związanych z realizacją projektów informatycznych (np. bazy danych, serwery itp.) oraz czy zna środowiska/języki wytwarzania oprogramowania. Kierownicy powinni również zorientować się, na ile klient potrafi odnaleźć się w strukturze projektowej. Sprawdzeniu należy poddać, czy klient uczestniczył w projektach informatycznych i jakie doświadczenia z nich wyniósł. Warto również sprawdzić, jakie metody zarządzania projektami informatycznymi zna klient. Kierownicy projektów powinni również zorientować się, na ile klient zna zasady iteracyjnego (przyrostowego) wytwarzania.

Klient powinien liczyć się z tym, że produkt w projekcie informatycznym przechodzi szereg iteracji i warto klienta zapytać, czy znane są mu te zasady.

Kolejna analiza dojrzałości klienta dokonywana przez kierownika powinna być związana z umiejętnościami artykułowania wymagań. Należy sprawdzić, czy klient zna języki modelowania, czy woli opisy słowne, czy umie przekazać precyzyjnie, czego oczekuje w projekcie.

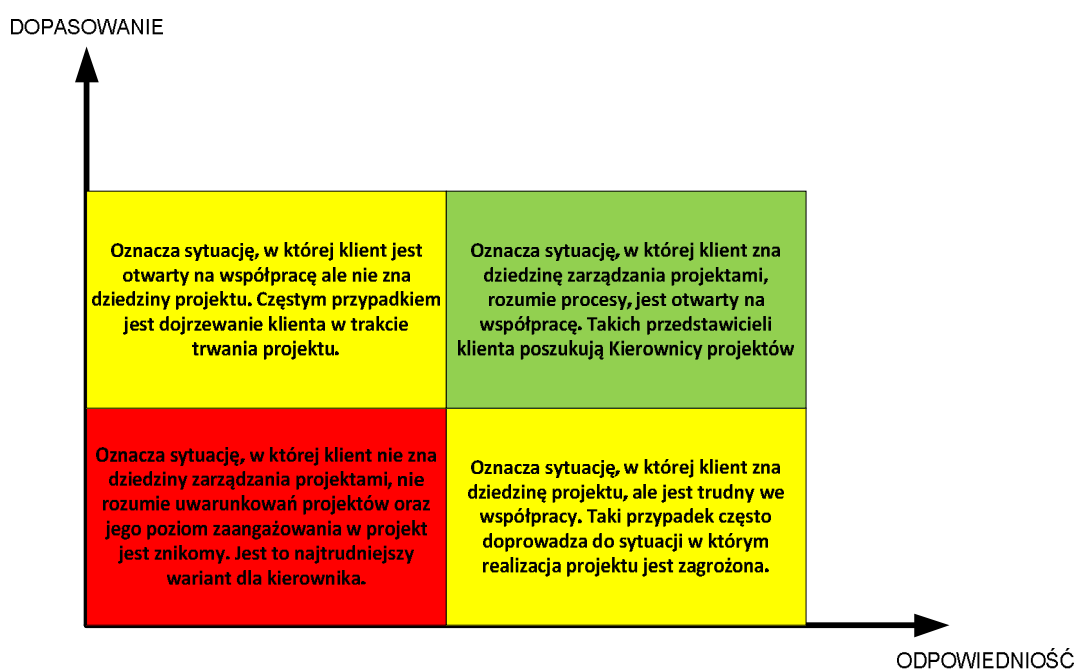
Do oceny odpowiedniości klienta potrzebne jest również sprawdzenie, w jakim stopniu klient będzie skłonny implikować zmienność w projekcie. Kierownik powinien przewidzieć, czy klient ma tendencję do zmiany zdania. Tę cechę najlepiej ocenić na podstawie obserwacji zachowań klienta.

Kompetencje związane ze współpracą w zespole określone mianem DOPASOWANIA obejmują umiejętność pracy w zespole, umiejętność dzielenia się wiedzą (współpracy), komunikatywność i zdolność do przepływu informacji, dyspozycyjność oraz znajomość technik pracy grupowej (zdolność do integracji).

Sprawdzając poziom dopasowania klienta należy zwrócić uwagę na to, czy cechy klienta pozwalają wprowadzić klienta w prace zespołu projektowego. Stąd też badając kompetencje związane z pracą w zespole kierownik powinien sprawdzić, czy klient lepiej czuje się w pracy wykonywanej samodzielnie, czy preferuje wykonywanie zadań w grupie. Należy sprawdzić, czy klient jest w stanie traktować uczestników projektu równorzędnie. Na ile też klient przejawia cechy współpracy, to znaczy, że chętnie będzie się dzielił wiedzą z uczestnikami. Ważne jest także określenie poziomu jego komunikatywności. Ta cecha jest skorelowana z umiejętnością dzielenia się wiedzą.

Biorąc pod uwagę fakt, że obecność klienta w projekcie jest niezbędna dla poprawnej realizacji projektu, należy również określić dyspozycyjność klienta. Kierownik powinien sprawdzić, na ile klient jest zdolny uczestniczyć w spotkaniach projektowych, czy może w sytuacjach awaryjnych włączyć się do prac zespołu.

Dojrzałość klienta stanowi zatem grupę cech i kompetencji, które mają znaczący wpływ na przebieg projektu oraz wymuszają dobranie odpowiedniej metody realizacji tego projektu. Wówczas dojrzałość klienta można zaprezentować w postaci układu macierzowego, który umożliwi sprecyzowanie dojrzałości klienta jako reprezentacji cech i kompetencji, czyli dopasowania i odpowiedniości.



Rys. 2.2. Macierz dopasowania i odpowiedności klienta

Źródło: opracowanie własne

Na rysunku 2.2. przedstawiono zależność pomiędzy dwoma składowymi dojrzałości klienta — odpowiednością i dopasowaniem. Na zielono zaznaczono sytuację najbardziej korzystną, gdy klient jest zarówno dopasowany, jak i odpowiedni. Jest to sytuacja najbardziej pożądana, a jednocześnie najrzadziej spotykana podczas realizacji zleceń projektowych. Kierownik projektu powinien uwzględnić poziom dojrzałości klienta zanim podejmie decyzje odnośnie doboru dobrych praktyk zarządzania. W sytuacji, gdy klient jest odpowiedni i dopasowany, można z powodzeniem stosować dowolne metody zarządzania projektami, gdyż ryzyko zmienności jest niewielkie. W pozostałych wypadkach (zwłaszcza w sytuacji klienta nieodpowiedniego i niedopasowanego) istnieje duże prawdopodobieństwo częstych zmian, do których lepiej dostosowane są metody lekkie. Największym zatem zadaniem dla menedżerów projektów jest dokładne poznanie poziomu dojrzałości klienta. Kierownik projektu, badając profil klienta, powinien pozycjonować go w jednym z zaprezentowanych na rysunku pól macierzy. Kombinacje odpowiedności i dopasowania pozwalają na następujące możliwości:

NO-ND (klient nieodpowiedni i niedopasowany)

Oznacza sytuację, w której klient nie zna dziedziny zarządzania projektami, nie rozumie uwarunkowań projektów oraz jego poziom zaangażowania w zespół jest właściwie zerowy. Jest to sytuacja najtrudniejsza dla kierownika projektu.

O-ND (klient odpowiedni i niedopasowany)

Oznacza sytuację, w której klient dobrze orientuje się w specyfice projektów informatycznych, ale współpraca z nim może być trudna. Klientowi brakuje predyspozycji do pracy grupowej, przez co zachodzi obawa, że będzie ciężko włączyć go do projektu jako stałego uczestnika.

NO-D (nieodpowiedni i dopasowany)

Oznacza sytuację, w której klient jest bardzo otwarty na współpracę i gotowy włączyć się do pracy zespołu, ale jego poziom znajomości specyfiki projektów informatycznych jest niewielki. Jednak należy przypuszczać, że dopasowanie klienta przejawiane przez zaangażowanie i chęć współpracy może spowodować dojrzewanie klienta w trakcie projektu i rozwijanie kompetencji merytorycznych.

O-D (klient odpowiedni i dopasowany)

Oznacza sytuację, kiedy klient doskonale wie, czego potrzebuje i potrafi to dookreślić precyzyjnie na każdym etapie projektu. Może wynikać to np. z faktu, że doskonale zna branżę informatyczną lub ma świadomość, że jego precyzyjne oczekiwania służą wszystkim — jemu samemu i wykonawcy projektu. Dodatkowo posiada odpowiednie predyspozycje do współpracy. Jest to sytuacja najbardziej pożądana w projekcie informatycznym.

Tak przedstawione kombinacje dwu kluczowych parametrów odpowiedności i dopasowania mogą stanowić punkt wyjścia do określenia dojrzałości klienta. Ocena poziomu dojrzałości staje się ważna dla wprowadzonego nowego trójkąta ograniczeń i potrzeby jego implementacji za pośrednictwem agentowego adaptacyjnego modelu zarządzania projektami informatycznymi.

2.1.2 Dojrzałość dostawcy w projekcie informatycznym

Kolejną zagregowaną zmienną decyzyjną nowego trójkąta ograniczeń, która powinna zostać poddana analizie z punktu widzenia wpływu na decyzje kierownika projektu, jest dojrzałość zespołu dostawcy. Dostawcą [40] w przypadku projektu informatycznego nazywa się organizację, która podejmuje się realizacji projektu informatycznego na zlecenie klienta. Najczęściej jednak mówiąc

o dostawcy w projekcie informatycznym, należy mieć na myśli zespół projektowy lub grupę zespołów projektowych w ramach organizacji dostawcy.

O tym, jak ważna jest rola zespołu projektowego/zespołów projektowych w projekcie informatycznym, wspomniano w rozdziale I. Trzeba jednak w tym miejscu podkreślić, że zespół jest podstawowym elementem każdej struktury projektowej i od sposobu jego zorganizowania zależy w dużej mierze późniejszy przebieg prac projektowych. Zadaniem kierownika projektu informatycznego jest bowiem podjąć odpowiednie decyzje związane z planowaniem tych zasobów (w tym ludzkich), które powinny być wykorzystywane w trakcie trwania projektu informatycznego. Takim podstawowym zasobem jest niewątpliwie zespół i ludzie go tworzący. Analizując zatem poziom dojrzałości zespołu (dostawcy) mającego realizować projekt informatyczny, należy de facto analizować poziom kapitału ludzkiego, jaki jest reprezentowany przez poszczególnych członków. **Kapitał ludzki** rozumiany jest jako kapitał „w nierozłączny sposób zintegrowany z człowiekiem (pracownikiem), jego wiedzą, doświadczeniem, obecnością i możliwościami działania w przedsiębiorstwie” [7]. Oznacza to zatem, że duży wpływ na powodzenie wszelkich prowadzonych działań zależy w dużej mierze od jakości tegoż kapitału. Nie ulega zatem wątpliwości, że w przypadku realizacji projektów informatycznych rola zespołu jest szczególnie istotna. Od tego, jak wydajny i skuteczny będzie zespół, zależy również powodzenie projektu informatycznego. Rolą menedżera jest w taki sposób kierować pracami i podejmować takie decyzje, aby zapewnić wysoki poziom dojrzałości zespołu

Znaczenie zespołu dostawcy w realizacji projektu

Liczebność zespołu

Zespół może składać się z kilku albo kilkunastu osób. W przypadku dużych projektów liczebność zespołu może być dużo większa, jednak zgodnie z dzisiejszymi zaleceniami zarządzania zespoły w organizacjach nie powinny być zbyt liczne [97]. Zaleca się w takich sytuacjach, aby duży (kilkudziesięcioosobowy) zespół podzielić na mniejsze zespoły, określając w każdym z nich lidera — osobę nadzorującą prace całego zespołu. Pierwszą zatem cechą związaną z zarządzaniem zespołem jest jego liczebność. Okazuje się bowiem, że w przypadku dużych zespołów przepływ informacji może być dość trudny. W zespołach kilkusobowych poziom komunikacji jest dużo lepszy. Drugą cechą związaną z liczebnością zespołu jest poziom integracji takiego zespołu. Zespoły o mniejszej liczebności integrują się łatwiej, przez co prowadzenie prac jest bardziej wydajne. Można w tym miejscu przywołać niektóre dobre praktyki zarządzania projektami, które wprost sugerują liczebność zespołu projektowego (np. w metodzie SCRUM zaleca się, aby wielkość zespołu projektowego liczyła od 4 do 6 osób)

Podział na role

Kolejną cechą, którą powinien uwzględnić kierownik projektu zarządzający zespołem, jest podział na role. Każda z metod zarządzania projektami informatycznymi sugerowała przypisanie uczestnikom dedykowanych ról. W zależności od metody zarządzania projektem, role te były mniej lub bardziej sprecyzowane. Nie ulega wątpliwości, że w celu lepszej organizacji pracy zespołu, należy te role wyodrębnić, a następnie przypisać odpowiednich członków do odpowiednich ról. Kierownik projektu, podejmując decyzje odnośnie przypisywania ról, powinien rozpatrywać role w dwóch kategoriach:

- **Role projektowe** — są to role związane z bezpośrednią realizacją projektu i związane z zakresem działań prowadzonych w ramach projektu. Typową rolą projektową jest np. kierownik projektu, analityk biznesowy itp. Rola projektowa jest zatem związana z wykonaniem określonych zadań, odpowiedzialnością za przewidziane produkty tych zadań.

- **Role zespołowe** (pozycja w zespole) — są to role określające pozycję w zespole danego uczestnika projektu. Wiązą się z preferowanym typem zadań, hierarchią w zespole, umiejętnością włączania się w pracę zespołu. Przykładem może być lider zespołu, który lubi nadzorować, albo skrupulatny wykonawca [7], który niekoniecznie chętnie kieruje innymi, ale lubi sumiennie wykonywać powierzone mu zadania.

Zadaniem kierownika jest zatem analizowanie i podejmowanie decyzji nie tylko odnośnie ról projektowych, ale także pozycji w zespole. Analiza ról projektowych powinna być dla kierownika stosunkowo łatwa i wywnioskowana na podstawie oczekiwań (wymagań) klienta. Kierownik bowiem, znając zakres projektu, może ocenić, czy większe zapotrzebowanie będzie na specjalistów od programowania czy raczej modelowania. Kierownik projektu powinien także umieć ocenić, czy jego programiści powinni znać dobrze wiele języków programowania, czy też wystarczy znajomość jednego. Trudniejszym zadaniem kierownika projektu jest ocenienie, która rola zespołowa jest preferowana przez pracownika. Niektórzy wolą pracować samodzielnie, kontaktując się z pozostałymi członkami tylko od czasu do czasu. Inni wolą poszukiwać nowych, nieszablonowych rozwiązań, do których następnie starają się przekonać innych. Wydaje się więc, że problematyka dopasowania pracowników do ról jest również ważnym czynnikiem w ocenie dojrzałości zespołu i wpływa na podejmowane przez kierownika projektu decyzje. Okazuje się bowiem, że tylko wtedy, gdy kierownik będzie miał dokładny obraz swojego zespołu, czyli podział na role projektowe, oraz będzie potrafił określić, jakie pozycje zespołowe preferują jego członkowie, można się spodziewać, że praca zespołowa spełni swoje zadanie, prowadząc do wzrostu wydajności, efektywności oraz uzyskania efektu synergii. Zespół, w którym wszystkie role są dopasowane do oczekiwań członków, można nazwać zespołem o dużej dojrzałości.

Dopasowanie pracowników do roli (pozycji w zespole)

Na podstawie powyższych rozważań należy uznać za celowe, aby kierownicy projektów zwracali uwagę na to, czy pracownicy są dopasowani do odpowiednich ról. Kierownik projektu powinien wiedzieć, jakie role projektowe mogą pełnić jego pracownicy. Aby jednak dobrze przypisać pracowników do ról projektowych, kierownik powinien mieć pełną wiedzę o członku zespołu. Wiedza ta w przypadku projektów informatycznych powinna dotyczyć nie tylko kompetencji informatycznych (znajomość programowania, języków modelowania itp.), ale również znajomości zasad zarządzania projektami informatycznymi, czy też umiejętności dostosowania się do reguł narzuconych przez dobre praktyki zarządzania projektami.

Kolejną trudnością związaną z dopasowaniem pracowników do roli jest sprawdzenie, do jakich ról zespołowych predysponowani są pracownicy. Niedopasowanie pracownika do roli w zespole może nieść ze sobą daleko idące konsekwencje związane z późniejszą realizacją projektu informatycznego. Powierzenie zadań monottonnych, powtarzalnych osobie, która lepiej czuje się w tworzeniu nowych rozwiązań i generowaniu pomysłów, wywoła u niej frustrację i niechęć do realizacji zadań. Kierownik zatem organizując prace projektowe (czyli planując zadania) powinien podjąć takie decyzje, aby zadania były dopasowane do predyspozycji osobowościowych członków zespołu, czyli ról zespołowych, jakie mogą pełnić.

Czerpanie wiedzy na temat członków zespołu

Aby jednak kierownik mógł dopasować role do projektu, niezbędny jest szereg działań poprzedzających, czyli analiz pozwalających na stworzenie dokładnego profilu uczestnika projektu. Podobnie jak w przypadku klienta, gdzie zalecano zbadanie jego profilu, tak w przypadku zespołu kierownik projektu powinien również dysponować pełnym obrazem swoich współpracowników. Stąd

też kierownik projektu powinien korzystać z wszystkich dostępnych źródeł wiedzy o członkach zespołu. Zasadniczo źródła wiedzy o członkach zespołu można podzielić na dwie kategorie [62]:

- **Źródła zewnętrzne** — to takie źródła, które są powszechnie dostępne dla kierownika, np. CV pracownika, opinie o nim z poprzedniego miejsca pracy, informacje publikowane w serwisach społecznościowych
- **Źródła wewnętrzne** — dotyczą wiedzy o pracowniku, jaką kierownik posiada na podstawie już odbytych projektów, wiedza organizacyjna (np. w firmie wiadomo, że z tym pracownikiem zawsze dobrze się współpracuje).

Z powyższego wynika, że kierownik projektu, chcąc określić poziom dojrzałości swojego zespołu w celu dobrania odpowiednich dobrych praktyk zarządzania projektem, powinien poddawać analizie swoich pracowników korzystając zarówno ze źródeł zewnętrznych jak i wewnętrznych. Obserwacje branży IT (i nie tylko tej branży) pokazują, że jedną z najbardziej precyzyjnych metod określenia pozycji w zespole jest stosowanie dedykowanych testów psychologicznych albo symulowanie zachowań zespołowych (np. przez powierzenie pracownikom wykonania jakiegoś zadania zespołowego, niekoniecznie dotyczącego projektu informatycznego, a następnie obserwacja zachowania współpracujących ze sobą osób). Sprawdzenie kompetencji zespołowych przy konstruowaniu zespołu jest dla kierownika znacznym ułatwieniem w związku z późniejszym określaniem ról. Jednak jest to możliwe w sytuacji, gdy kierownik od początku samodzielnie buduje zespół. Nie jest to jednak sytuacją typową.

Pochodzenie zespołu

Na podejmowane przez kierownika decyzje odnośnie zespołu wpływ ma niewątpliwie wiedza o członkach zespołu, którą czerpie z różnych źródeł. Jednak nie zawsze kierownik zna swój zespół od początku, albo od początku go konstruuje dobierając dowolnych współpracowników. Przyjmuje się bowiem dwa podstawowe źródła pochodzenia zespołów [62] — zespoły budowane od początku (konstruowane) oraz zespoły narzucone kierownikowi projektu.

Pierwsza sytuacja jest idealna dla kierownika, gdyż oznacza sytuację, w której kierownikowi projektu powierza się zadanie dobrania członków zespołu do projektu. Kierownik może zatem samodzielnie sprawdzać kompetencje osób, z którymi zamierza współpracować, i na tej podstawie przydzielać im role projektowe oraz zadania. W sytuacji drugiej nie jest już to tak oczywiste. W zespołach narzuconych najczęściej mówi się o sytuacji, w której zespół został powołany (np. przez zarząd organizacji dostawcy), a kierownikowi powierzono zarządzanie projektem w ramach tego powołanego zespołu. W tej sytuacji kierownik projektu również powinien określić poziom kompetencji tych członków. W przypadku zespołu konstruowanego od podstaw kierownik ma jednak możliwość komponowania zespołu według własnego uznania. W drugim przypadku musi zaakceptować powierzony mu zespół. Nie zmienia to jednak faktu, że zarówno w jednym jak i drugim przypadku niezbędne jest określenie poziomu kompetencji poszczególnych członków zespołu w celu określenia poziomu dojrzałości tego zespołu. To w odniesieniu do dojrzałości zespołu projektowego możliwe jest podejmowanie wszelkich decyzji związanych z doбором dobrych praktyk zarządzania projektami.

Wpływ dojrzałości zespołu na przebieg projektu i decyzje z nim związane

Powyższe czynniki zostały omówione w celu wskazania związku dojrzałości zespołu projektowego występującego z ramienia dostawcy z realizacją projektu informatycznego. Nie ulega również wątpliwości, że poziom dojrzałości zespołu projektowego przekłada się na szereg decyzji kierowniczych w projektach informatycznych. Kierownik projektu powinien bowiem uwzględnić poziom dojrzałości zespołu dobierając określone dobre praktyki zarządzania projektami informatycznymi. Okazuje się bowiem, że bardzo często zespoły mniej dojrzałe nie radzą sobie dobrze

w projektach realizowanych z wykorzystaniem dobrych praktyk zawartych w lekkich metodach zarządzania projektami. Zespołom o niskim poziomie dojrzałości ciężko samodzielnie podjąć decyzje planistyczne związane np. z planowaniem iteracji. Analogicznie, zespołom bardzo doświadczonym łatwiej jest samodzielnie organizować sobie pracę, niż wpasowywać się w dość sztywne ramy projektowe zalecane w metodach klasycznych. W celu zaprezentowania, w jaki sposób dobór dobrych praktyk może oddziaływać na projekt w odniesieniu do dojrzałości zespołu, warto posłużyć się odpowiednim przykładem.

Przykład 1. Wpływ dobrych praktyk na procesy zarządzania zespołem

Projekt: Pilotażowa integracja technologii Rational Team Concert (RTC) z innymi technologiami posiadanymi przez klienta (BMC Remedy). Opracowania interfejsu wymiany danych między RTC a BMC Remedy.

Klient: PSE INFO, Warszawa.

Zespoły: studenci ostatniego roku studiów Wydziału Zarządzania i Ekonomii Politechniki Gdańskiej, specjalności Zarządzanie Technologiami Informatycznymi.

Termin realizacji: kwiecień–lipiec 2009 r.

Potrzeba dobierania dobrych praktyk do zarządzania projektem w odniesieniu poziomu dojrzałości zespołu została zaobserwowana w trakcie realizowanego projektu. Na początku realizacji projektu zaproponowano mało doświadczonemu zespołowi złożonemu z 18 członków podział się na mniejsze zespoły, które będą miały za zadanie samodzielnie realizować zadania projektowe w ramach przyjętego zakresu projektu. Zastosowano lekkie podejście do zarządzania projektem i dobrą praktykę — realizację prac w krótkich sprintach (tabela 1.7). Niestety, taka decyzja nie przelożyła się na znaczny postęp prac. Gdy kierujący projektem zauważył, że brak odgórnego przyjęcia zadań i harmonogramu ich realizacji nie przekłada się na prace zespołu, postanowiono stworzyć harmonogram. Następnie zaprezentowano uczestnikom zadania i ich sposób realizacji (dobra praktyka — Planowanie i harmonogram, tabela 1.3). Takie podejście spotkało się z akceptacją członków zespołów i stanowiło przełom w projekcie, dając szansę efektywnego kontynuowania prac w ramach wyodrębnionych odgórnego zadań. Należy zatem przyjąć, że kierownicy projektów powinni uzależniać decyzje odnośnie stosowania określonych dobrych praktyk zarządzania projektami od poziomu dojrzałości zespołu projektowego.

Prezentowany przykład pokazuje, jak ważne jest określenie poziomu dojrzałości i jaki może mieć wpływ na realizację projektu. Niedopasowanie dobrych praktyk zarządzania projektami (czyli błędnie podjęte decyzje kierownika projektu) mogą skutkować negatywnie dla postępów w realizacji projektu, ale mogą również powodować niezadowolenie zespołu. Dwa główne obszary odpowiedzialności kierownika projektu — zarządzanie zadaniami (realizacją) projektu informatycznego oraz zarządzanie zespołem — bardzo wyraźnie zależą od decyzji związanych z poziomem dojrzałości zespołu.

Ponadto kolejny raz uwidocznione zostało, że niedopasowanie dobrych praktyk zarządzania projektem (czyli podjęcie błędnych decyzji) skutkuje negatywnie dla realizacji projektu. Tym samym po raz kolejny potwierdzona została teza, że przed przystąpieniem do realizacji projektu i podjęciem decyzji w ramach klasycznych ograniczeń zasobowych (harmonogram, budżet, zakres) potrzebna jest analiza zmiennych decyzyjnych wpływających na dobór dobrych praktyk zarządzania projektami, którą reprezentuje nowy trójkąt ograniczeń (dojrzałość klienta, dojrzałość zespołu i entropia projektowa).

Pozostaje jeszcze do rozstrzygnięcia, jak w sposób jednoznaczny definiować dojrzałość zespołu projektowego. Nie ulega wątpliwości, że dojrzałość zespołu uzależniona jest od doświadczeń poszczególnych członków takiego zespołu.

Definicja 2.2

Dojrzałość zespołu projektowego to zestaw doświadczeń poszczególnych członków zespołu w realizacji projektów informatycznych oraz zdolności do pełnienia określonych pozycji w zespole projektowym.

Potrzeba analizy dojrzałości zespołu

Powyższe rozważania wykazały, że zmienna zagregowana, jaką jest dojrzałość zespołu dostawcy, jest jedną z podstawowych zmiennych dla podejmowania decyzji kierowniczych w projektach informatycznych. Obserwacja wielu kompetencyjnych problemów związanych z dojrzałością zespołu projektowego, a konkretniej rzecz ujmując, z dojrzałością jego uczestników, skłoniła do poszukiwań metodycznego rozwiązania pozwalającego menedżerom na analizę tego poziomu dojrzałości. Pierwszym założeniem przy pomiarze dojrzałości zespołu jest pominięcie kompetencji merytorycznych członków zespołu. Przyjmuje się, że umiejętności informatyczne są znane (wynikają z analizy CV czy wcześniejszej współpracy z tymi osobami). Takie założenie jest niezbędne dla prowadzenia dalszych działań związanych z analizą dojrzałości zespołu. Kierownik bowiem podejmując się realizacji projektu musi przyjąć założenie, że osoby realizujące projekt posiadają odpowiednią wiedzę. Tym bardziej, że w organizacji dostawcy (czyli organizacji IT) pracują przeważnie osoby znające języki programowania i posiadające odpowiednią wiedzę z zakresu informatyki. Stąd też poziom wiedzy merytorycznej jest stały i znany kierownikowi. Kierownik nie przeprowadza analizy umiejętności informatycznych (wytwórczych) swoich członków. Natomiast poziom znajomości zasad, jakie obowiązują podczas realizacji projektów informatycznych (np. metod zarządzania projektami informatycznymi), nie musi być znany członkom zespołu i stan ich wiedzy i kompetencji w tym obszarze powinien być sprawdzony. Stąd też pierwsze zalecenie odnosi się do zmierzenia poziomu przygotowania do realizacji projektów informatycznych. Tę część analizy dojrzałości nazwano analizą doświadczenia.

- *Analiza doświadczenia* polega na zbadaniu, na ile członek zespołu projektowego rozumie specyfikę projektów informatycznych oraz zasady, jakimi kierują się projekty. Jeżeli występował wcześniej w roli kierownika, należy przypuszczać, że zna metody zarządzania projektami. Jeżeli uczestniczył w niewielkiej liczbie projektów lub w ogóle, na początku dużo czasu trzeba będzie poświęcić na przygotowanie takiego uczestnika do prac związanych z realizacją projektu.

Druga grupa analiz, jaką powinien przeprowadzić kierownik projektu, wiąże się ze sprawdzeniem, jaki charakter zadań jest najbardziej oczekiwany przez danego członka zespołu. Jeżeli uda się określić, jakie zadania preferuje dana osoba, łatwiej będzie przypisać jej określoną rolę projektową i powierzyć zakres obowiązków. Ta analiza wiąże się niewątpliwie z czynnikami psychologicznymi w zarządzaniu zespołem i można ją nazwać analizą kompetencji zespołowych.

- *Analiza kompetencji zespołowych* — Psychologia dostarcza szeregu narzędzi, które mogą być wykorzystywane do pomiaru poziomu kompetencji zespołowych. Uznano, że przydatnym narzędziem, z którego mogą korzystać kierownicy przy określaniu pozycji w zespole swoich członków, jest test Belblina [7]. W teście tym, w zależności od odpowiedzi na pytania poszczególnych członków zespołów, możliwe jest pozycjonowanie osób w zespole. Test Belblina pokazuje, jaki charakter zadań (bardziej twórcze czy powtarzalne, bardziej zarządcze czy operacyjne) jest preferowany przez osobę badaną.

Test Belblina jest o tyle wygodnym narzędziem, że z danych jakościowych oceny organizacji pozwala przejść na dane ilościowe takiej oceny. Przy czym warto zaznaczyć, że test Belblina ma charakter ogólny, tzn. sprawdza predyspozycje uczestników do pracy grupowej. Nie bada

podstawowych kompetencji związanych z projektami informatycznymi. W związku z tym uznano, że pytania testu Belblina są niewystarczające do pomiaru dojrzałości zespołu projektowego, ale mogą być częścią takiego pomiaru. O ile pozwalają one określić preferowane zajęcia uczestników projektu (ktoś woli kierować, ktoś inny realizować zadania), o tyle dla pełnego obrazu dojrzałości zespołu potrzebne jest kilka innych, kompetencyjnych pytań związanych ze specyfiką projektów informatycznych. Przy tworzeniu instrumentu pomiarowego do oceny dojrzałości zespołu (dostawcy) uznano więc za celowe uwzględnić zarówno pytania odnośnie doświadczenia projektowego, jak również kompetencji zespołowych.

Dokonując pomiaru dojrzałości zespołu, zaleca się dokonanie pomiaru kompetencji zespołowych oraz doświadczenia pojedynczych członków, a następnie stworzenie zestawienia zbiorczego przez kierownika, które pozwoli na ogólną ocenę poziomu dojrzałości zespołu. Proponuje się przyjęcie dojrzałości zespołu na pięciostopniowej skali. Poziomy dojrzałości proponowane do określenia poziomu dojrzałości zespołu zaczerpnięto z modelu dojrzałości organizacji (ang. *Capability Maturity Model Integration*) (CMMI) [37,80]. Model CMMI odnosi się co prawda do całych organizacji, ale traktując zespół jako część organizacji dostawcy przyjęcie takiej miary wydaje się zasadne. Poziomy dojrzałości zespołu określano na podstawie przyjętych w pracy miar dojrzałości oraz poziomów zgodnie z CMMI.

- **POZIOM 1 — WYKONYWALNY** Zespół charakteryzuje się niskim poziomem doświadczenia, jednocześnie brakuje wyraźnych cech do współpracy. Zespół jest w stanie wykonywać wyłącznie narzucone odgórnie zadania.
- **POZIOM 2 — ZARZĄDZANY** Zespół ma niski poziom doświadczenia, ale zarysowują się wyraźne predyspozycje do pełnienia różnych ról w zespole. Procesy takiej organizacji są planowane i egzekwowane zgodnie z ustaloną przez kierownika projektu strategią. Realizacja procesu jest poddawana ocenie zgodności z planem. Akcje naprawcze w przypadku niezgodności realizacji procesów z harmonogramem są podejmowane. Obserwuje się zaangażowanie członków zespołu, którzy wyposażeni są w odpowiednie technologie informatyczne do kontrolowania otrzymywanych produktów.
- **POZIOM 3 — DEFINIOWALNY** Zespół charakteryzuje się średnim poziomem doświadczenia projektowego oraz zarysowują się przynajmniej u połowy członków zespołu wyraźne predyspozycje do pełnienia określonych ról zespołowych.
- **POZIOM 4 — ILOŚCIOWO ZARZĄDZANY** Zespół charakteryzuje wysoki poziom doświadczenia potwierdzony uczestnictwem w wielu projektach informatycznych oraz kompetencje do pracy zespołowej. Są one wyraźne u większości członków.
- **POZIOM 5 — OPTYMALIZACJI** Zespół cechuje wysoki poziom doświadczenia w realizacji projektów, a jednocześnie widoczne są typowe role zespołowe jak lider, wykonawcy, poszukiwacze źródeł. Taki zespół jest w stanie nie tylko samodzielnie organizować sobie pracę, ale również samodzielnie wyciągać wnioski z własnych działań, korygować te działania i usprawniać.

O zakwalifikowaniu zespołu do jednego z tych poziomów decyduje kierownik na podstawie analiz dojrzałości. Warto zwrócić uwagę, że sprawdzenie poziomu doświadczenia oraz kompetencji zespołowych poszczególnych członków nie prowadzi jeszcze do stwierdzenia, jaki jest poziom dojrzałości takiego zespołu. W dalszej części zaproponowano autorską interpretację wyników otrzymanych w wyniku analizy dojrzałości pojedynczych członków i sposób przełożenia takiej analizy na dojrzałość całego zespołu występującego z ramienia dostawcy w projekcie informatycznym.

2.1.3 Entropia projektu informatycznego

Dwie uprzednio omówione zmienne, których analiza jest niezbędna z punktu widzenia decyzji podejmowanych przez kierowników w projektach informatycznych, wiązały się przede wszystkim z zespołem i zachowaniem jego członków. Analiza dojrzałości klienta i zespołu dostawcy dotyczyła przede wszystkim analizy kompetencji i umiejętności. Rodzi się jednak pytanie, czy to wystarczająca liczba zmiennych przy podejmowaniu decyzji przez kierowników projektów. Do jakiej bowiem kategorii zaliczyć decyzje odnośnie analizy ryzyka projektowego niezbędnego do oceny w wielu dobrych praktykach prezentowanych w rozdziale I metod? Wreszcie, jak ująć stopień złożoności procesów projektowych? Te pytania są niezbędne z punktu widzenia zarządzania projektem informatycznym i wiążą się z immanentnymi cechami tego typu projektu.

Uznano zatem, że niezbędna jest analiza również stanu realizowanego projektu. Pierwsze spostrzeżenia po analizie stanu dobrych praktyk (rozdział I) wiążą się z zakresem. Ale analiza zakresu projektu nie jest wystarczająca do tego, aby dobierać dobre praktyki (dlatego buduje się nowy, uzupełniający trójkąt ograniczeń). Rodzi się jednak pytanie, jak powinien zachować się kierownik projektu w sytuacji, kiedy nie posiada pełnej wiedzy o zakresie projektu, stopniu jego złożoności i niepewności jego realizacji, i jakie decyzje powinien wtedy podejmować. Stąd też zaproponowano przyjęcie zagregowanej zmiennej entropii projektu.

Termin „entropia” funkcjonuje w teorii informacji i odnosi się do niepewności wystąpienia danego zdarzenia elementarnego w następnej chwili [20]. Przyjmuje się również, że entropia w odniesieniu do informacji to średnia ilość informacji przypadająca na dany moment danego wydarzenia, niezbędna, aby zanikła wszelka niepewność związana z tym wydarzeniem.

W powyższych definicjach entropii uwidaczniają się pewne analogie w stosunku do projektów informatycznych, w których poziom precyzyjnej wiedzy o projektach decyduje o sukcesie ich realizacji. Opierając się na pojęciu entropii informacji oraz na potrzebie uwzględniania niepewności, dla stanu projektu zaproponowano następującą definicję entropii projektu informatycznego:

Definicja 2.3

Entropia projektu informatycznego p_t to miara stopnia niepewności realizacji projektu związana z ogólnym poziomem wiedzy o precyzji sformułowanych wymagań, pewności zdefiniowanego zakresu projektu oraz poziomie złożoności zadań projektowych i ryzyka wykonawczego.

$$\mathbf{p}_t = \begin{bmatrix} ps_t \\ pz_t \\ pp_t \\ pr_t \end{bmatrix} \quad (1)$$

- \mathbf{p}_t — czterowymiarowa ocena entropii projektu informatycznego,
- p_t — skalarna ocena entropii projektu informatycznego, $p_t \in \langle \text{niska, średnia, wysoka} \rangle$,
- ps_t — zmienna służąca do oceny stopnia precyzji sformułowania wymagań, $ps_t \in \langle \text{mały, średni, duży} \rangle$,
- pz_t — zmienna służąca do oceny stopnia pewności zdefiniowanego zakresu projektu, $pz_t \in \langle \text{mały, średni, duży} \rangle$,
- pp_t — zmienna służąca do oceny stopnia złożoności zadań projektowych, $pp_t \in \langle \text{mały, średni, duży} \rangle$,
- pr_t — zmienna służąca do oceny stopnia ryzyka wykonawczego, $pr_t \in \langle \text{mały, średni, duży} \rangle$,

t — czas realizacji projektu, $t \in \langle 1, t_{kon} \rangle$.

Założono klasyfikację entropii projektu na trzech poziomach:

- PROJEKTY O MAŁEJ ENTROPII — projekty o wysokim stopniu precyzji sformułowania wymagań, wysokim stopniu pewności zdefiniowanego zakresu projektu, niskim stopniu złożoności zadań projektowych, niskim ryzyku wykonawczym.
- PROJEKTY O ŚREDNIEJ ENTROPII — do projektów o średniej entropii należy zakwalifikować wszystkie nie mieszące się w klasyfikacji dla projektów o małej i dużej entropii.
- PROJEKTY O DUŻEJ ENTROPII — projekty o niskim stopniu precyzji sformułowania wymagań, niskim stopniu pewności zdefiniowanego zakresu projektu, wysokim stopniu złożoności zadań projektowych, wysokim ryzyku wykonawczym.

W zależności od stopnia entropii menedżer powinien podjąć decyzję co do wyboru dobrych praktyk zarządzania projektem informatycznym. Projektom o małej entropii sprzyjają właściwie wszystkie metody zarządzania projektami, projektom o dużej entropii zalecane są podejścia lekkie.

Wpływ entropii na decyzje kierowników projektów

Każdy projekt informatyczny poprzedzony jest szeregiem odpowiednich analiz związanych z określeniem zakresu projektu. Przed zawarciem umowy na realizację projektu klient określa najczęściej swoje oczekiwania, definiuje się podstawowe wymagania, próbuje się przyjąć ramy finansowe dla realizowanego projektu. Jednak nieodłącznym elementem tych działań inicjujących projekt jest zjawisko niepełnej wiedzy i informacji o projekcie. Wynika ono po pierwsze z nadmiaru informacji. Próbuje się bowiem przekonywać klienta do określonego typu rozwiązań, klient z kolei próbuje zarysować swoje oczekiwania najczęściej w mało uporządkowany sposób. Podczas odbywania (najczęściej cyklu) takich spotkań również strona dostawcy przedkłada szereg możliwych rozwiązań, oczekując od klienta wskazania tego, które będzie dla niego najlepsze.

Nieuporządkowanie informacji o projekcie informatycznym jest zjawiskiem, w ramach którego kierownicy projektów muszą podejmować określone decyzje. Brak o projekcie pełnych i wyczerpujących informacji powoduje, że kierownicy projektów podczas rozmów z klientem nie potrafią sprecyzować wyraźnie misji projektu (zarysu działań) oraz celu projektu i kształtu ostatecznego produktu powstałego w wyniku realizacji projektu. Te wszystkie problemy przekładają się na także na niepewność kierownika projektu dotyczącą struktury realizowanego projektu oraz efektów pracy, za które kierownik odpowiada. Trudno jest w takiej sytuacji podejmować jakiegokolwiek decyzje planistyczne.

Wydaje się również, że stopień takiej entropii projektu jest istotny z punktu widzenia doboru dobrych praktyk zarządzania projektami informatycznymi. Łatwo bowiem zauważyć, że jeżeli entropia projektu jest duża, czyli nieuporządkowanie informacji w projekcie jest duże, kierownikom będzie trudno planować procesy realizacji projektu. Tym samym dobre praktyki zarządzania procesem (jak w PMBoK czy RUP) mogą okazać się nieefektywne.

Poddając analizie poziom entropii projektowej w projektach informatycznych warto również zwrócić uwagę na konsekwencje dla kierowników projektów płynące z analizy tej zmiennej. Opisane bowiem powyżej składowe entropii (jak poziom złożoności zadań, stopień ryzyka itp.) powinny zostać uwzględnione przez kierownika projektu lub osobę przez niego wyznaczoną. Ze względu na szereg zagadnień pokrewnych tematyce zarządzania wymaganiami może być to np. analityk biznesowy. Faktem jednak jest, że wysoki poziom entropii projektu (a co za tym idzie, wysoka niepewność kierownika co do sposobu realizacji projektu) może istotnie wpłynąć na przebieg prac projektowych.

Jedną z podstawowych konsekwencji wysokiej entropii projektowej jest możliwość opóźnień w realizacji zadań. Wynika to z faktu, że brak precyzyjnych wymagań utrudnia kierownikowi zaplanowanie zadań. Wysoka entropia projektu może również wydłużyć etap inicjacji projektu. Aby się przed tym zabezpieczyć, kierownik projektu powinien podjąć odpowiednie decyzje. Zadaniem kierownika jednak nie jest zmniejszenie poziomu entropii, a raczej podjęcie takich decyzji, aby dostosować się do przewidywanego poziomu entropii.

Kolejną bardzo ważną konsekwencją realizacji projektu (przy wysokiej jego entropii) jest możliwość całkowitego zaprzestania prac. Nieuporządkowanie informacji płynących od klienta i zespołu może bowiem prowadzić do konfliktu interesów i niemożności prowadzenia prac. Potwierdza to poniższy przykład zaczerpnięty z doświadczeń autora niniejszej rozprawy.

Przykład 2. Wpływ entropii na przebieg projektu

Projekt: Budowa witryny WWW (z wbudowanym systemem zarządzania treścią) na potrzeby konferencji KKIO i SCR 2010.

Zespoły: Zespół Zakładu Zarządzania Technologiami Informatycznymi Wydziału Zarządzania i Ekonomii Politechniki Gdańskiej, Katedra Informatyki i Automatyki Wydziału Elektrotechniki i Informatyki Politechniki Rzeszowskiej (członkowie komitetów organizacyjnych obydwu konferencji KKIO i SCR).

Termin realizacji: listopad 2009–luty 2010.

Projekt dotyczył wdrożenia i adaptacji technologii do zarządzania treścią na potrzeby organizowanej przez obydwa ośrodki organizujące konferencję. Okazało się jednak, że nieuporządkowany przepływ informacji pomiędzy zespołami z tych dwóch ośrodków a dostawcą rozwiązania informatycznego był tak duży, że zaprzestano prac projektowych. Dopiero po uporządkowaniu relacji pomiędzy wszystkimi uczestnikami (poprzez uporządkowanie obiegu informacji) prace wznowiono.

2.2 Stosowanie dobrych praktyk zarządzania projektami informatycznymi w kontekście zmiennych decyzyjnych

Przeprowadzona analiza zmiennych decyzyjnych, względem których kierownicy projektów podejmują decyzje w projektach informatycznych, wykazała konieczność dokonywania pomiarów wartości tych zmiennych. Stwierdzono, że kierownicy projektów informatycznych powinni dobierać dobre praktyki zarządzania projektami informatycznymi w zależności od wartości zmiennych decyzyjnych.

Nawiązując do wniosku z rozdziału I, gdzie wykazano wyraźny brak kontekstu stosowania dobrych praktyk zarządzania projektami przy stosowaniu metod zarządzania projektami, należy stwierdzić, że taki kontekst nadają właśnie te trzy podstawowe zmienne decyzyjne: dojrzałość klienta, dojrzałość zespołu i entropia projektu. To właśnie w zależności od tych zmiennych kierownicy powinni dobierać dobre praktyki zarządzania projektami. W ten sposób uniknie się błędów niedopasowania (tak jak w przedstawionych przykładach), gdzie na skutek błędnego stosowania dobrych praktyk (bez uwzględnienia zmiennych decyzyjnych nowego trójkąta ograniczeń) prace projektowe były przerywane.

Aby jednak taki dobór był możliwy, postanowiono opracować odpowiedni mechanizm doboru dobrych praktyk w zależności od wartości przyjętych przez zmienne nowego trójkąta ograniczeń. Poszukując odpowiedniego algorytmu, postanowiono stworzyć funkcję dopasowania, która będzie prezentować, w jakim stopniu dobra praktyka pasuje do wartości poszczególnych zmiennych decyzyjnych. Taką funkcję dopasowania dobrych praktyk nazwano funkcją A_MATCH.

Przeznaczenie tej funkcji ma podwójne zastosowanie. Może być zestawem wskazówek dla kierowników projektów przy dobieraniu dobrych praktyk do swoich projektów oraz funkcja ta jest elementem adaptacyjnego agentowego modelu zarządzania projektami informatycznymi.

Funkcja skonstruowana jest w postaci tablicy, gdzie wybranym dobrym praktykom zarządzania projektami (wyodrębnionymi z poszczególnych metod zarządzania projektami w rozdziale I) przyporządkowuje się wartość dopasowania. Wartości dopasowania podawane są w skali trójstopniowej, gdzie:

- 1 — oznacza słabe dopasowanie dobrej praktyki do danej wartości zmiennej decyzyjnej,
- 2 — oznacza średnie dopasowanie dobrej praktyki do danej wartości zmiennej decyzyjnej,
- 3 — oznacza dobre dopasowanie dobrej praktyki do danej wartości zmiennej decyzyjnej,

Przyporządkowania określonych wartości dopasowania dobrej praktyki do zmiennej decyzyjnej ustalono podczas eksperymentu metodą ekspercką. Czterej eksperci zarządzania projektami informatycznymi (kierownicy projektów oraz pracownicy naukowcy zajmujący się zarządzaniem projektami informatycznymi) określali, na ile ich zdaniem dobra praktyka dopasowana jest do danej wartości zmiennej decyzyjnej. Przyjmowano wartości najczęściej podawane przez ekspertów, a w kwestiach spornych starano się określić wspólne stanowisko czwórki ekspertów.

Jednocześnie należy zaznaczyć, że to przyporządkowanie ma charakter poglądowy i może być dowolnie modyfikowane. W tym celu przygotowano odpowiedni arkusz kalkulacyjny pozwalający kierownikowi samodzielnie określać wartości dopasowania (gdyby kierownik czuł się wystarczająco kompetentny samodzielnie określić wartości dopasowania).

Przygotowano również odpowiedni interfejs do przetwarzania zapytań kierowników projektów, gdyby poszukiwali najlepszych dopasowań do swoich projektów. Arkusz został zaimplementowany w narzędziu do zarządzania projektami informatycznymi (co opisano w rozdziale VI niniejszej rozprawy) jako algorytm adaptacyjny na najwyższym poziomie adaptacyjnego agentowego modelu zarządzania projektami informatycznymi.

Tab. 2.1 Funkcja A_MATCH

#	DOBRA PRAKTYKA	POCHODZENIE	DOJRZAŁOŚĆ KLIENTA				DOJRZAŁOŚĆ ZESPOŁU					ENTROPIA PROJEKTU		
			D-O	ND-O	D-NO	ND-NO	1-wykon.	2-zarz	3-def	4-ilosc	5-opty	MAŁA	ŚREDNIA	DUŻA
1	Podziel projekt na cztery fazy	RUP	1	2	2	3	3	3	2	2	1	3	2	1
2	Definiuj produkty pracy jako wej/wyj z zadań	RUP	1	3	2	2	3	3	2	2	2	3	3	2
3	Definiuj dokładnie role projektowe	RUP	1	2	1	3	3	3	3	3	2	3	3	1
4	Planowanie struktury przepływu pracy	RUP	2	3	2	3	3	3	3	2	2	1	2	2
5	Stosuj modele wysokopoziomowe	RUP	2	2	2	3	2	2	2	1	1	1	2	3
6	Gromadź zadania w ramach dyscyplin	RUP	1	3	1	3	3	2	2	2	1	1	1	1
7	Ustal strukturę zarządzającą w postaci Komitetu	PRINCE2	1	2	2	2	2	2	1	1	1	2	2	3
8	Określ dokładnie cele projektu	PRINCE2	1	2	2	3	3	3	2	2	2	2	2	1
9	Ustal wsparcie projektu	PRINCE2	1	3	2	3	3	3	3	2	2	2	2	3
10	Przygotuj produkty pośrednie projektu	PRINCE2	1	2	1	2	3	3	2	2	2	2	2	3
11	Przygotowuj dokumentację na każdym etapie	PRINCE2	2	3	2	3	3	2	2	1	1	2	2	1
12	Definiuj wszystkie procesy do realizacji projektu	PRINCE2	1	2	2	3	3	3	2	1	1	2	2	1
13	Realizuj pracę w krótkich sprintach (przebiegach)	SCRUM	3	2	3	1	1	2	2	2	3	2	2	3
14	Spotykaj się z uczestnikami codziennie	SCRUM	2	3	2	3	3	3	2	2	1	2	2	3
15	Wyznacz osobę do nadzoru pracy zgodnie z metodą	SCRUM	2	2	2	2	3	2	2	1	1	1	1	2
16	Sporządź rejestr produktu	SCRUM	3	3	3	3	3	3	2	2	2	2	2	1
17	Sporządź rejestr sprintu (przebiegu)	SCRUM	2	2	2	2	3	3	2	2	2	2	2	1
18	Monitoruj projekt za pomocą wykresu spalania	SCRUM	1	2	1	3	2	2	2	3	3	3	3	1
19	Zaczynaj pracę od opowieści klienta	XP	3	1	2	1	2	2	2	3	3	3	2	1
20	Zastosuj pracę w parach	XP	3	2	3	2	3	3	3	2	2	2	2	1
21	Realizuj zadania w oparciu o testy	XP	2	1	2	1	1	1	2	3	3	3	2	2
22	Planuj krótkie przebiegi	XP	3	2	3	2	2	2	2	3	3	2	2	3
23	Zawrzyj dokumentację w kodzie	XP	3	1	3	1	1	1	1	2	2	1	1	2
24	Twórz metaforę systemu	XP	3	1	2	2	2	2	2	1	1	2	2	3
25	Ustal wejścia i wyjścia do każdego procesu	PMBok	1	2	2	3	3	3	2	2	1	2	2	1
26	Wyodrębnij wszystkie procesy realizacji projektu	PMBok	1	2	2	2	3	3	3	2	2	2	2	1
27	Precyzuj sposoby przetwarzania wejść do zadań	PMBok	1	2	1	2	3	2	2	2	1	2	1	1
28	Ustal odpowiedzialność za każdy proces	PMBok	1	2	2	3	3	3	3	2	1	1	1	2
29	Przygotuj odpowiednią dokumentację realizacji	PMBok	2	3	2	3	3	3	2	2	2	2	2	2
30	Koncentruj działania na harmonogramie i budżecie	PMBok	1	2	2	2	2	2	2	1	1	3	2	1

Źródło: opracowanie własne

2.3 Podsumowanie

W niniejszym rozdziale zaprezentowano trzy podstawowe zagregowane zmienne, które powinny być uwzględniane przez menedżerów projektów podczas podejmowania decyzji dotyczących doboru metody zarządzania projektem. Dobór ten będzie tylko wtedy efektywny, gdy kierownik dokona analizy dojrzałości klienta, dla którego projekt jest realizowany, analizy dojrzałości własnego zespołu, który ma realizować projekt, oraz analizy entropii projektu. Te trzy zmienne zaprezentowane w niniejszym rozdziale układają się w nowy trójkąt ograniczeń. Analogicznie do klasycznego trójkąta ograniczeń składającego się z harmonogramu, budżetu i zakresu projektu, w niniejszym rozdziale wyodrębniono trójkąt składający się z dojrzałości klienta, dojrzałości zespołu i entropii projektowej. Oznacza to zatem, że w zależności od wartości tych zmiennych powinna być dobierana metoda zarządzania projektem lub poszczególne dobre praktyki zarządzania projektem. Jednakże określenie poziomów wartości tych zmiennych wymaga zastosowania odpowiednich instrumentów pomiarowych, dlatego zostały one również zaprezentowane w niniejszym rozdziale. Jeżeli kierownikom projektów uda się pomierzyć tak ważne parametry jak dojrzałość klienta, dojrzałość zespołu i entropia projektu, ryzyko niedopasowania metody lub dobrych praktyk będzie zminimalizowane. Największym problemem dla kierowników projektów związanym z wymienionymi zmiennymi decyzyjnymi jest ich pomiar. Nie każdy kierownik projektu będzie w stanie samodzielnie dokonywać szeregu analiz stanu swoich projektów, a tym bardziej wyciągać wnioski z tej analizy pozwalające mu na efektywny dobór metody zarządzania projektem. Stąd też uznano, że należy opracować formalne rozwiązanie, które pozwoli na analizę tych zmiennych i dobór metody lub dobrych praktyk do specyfiki danego projektu. Takim rozwiązaniem powinien być odpowiedni model zarządzania projektem informatycznym.

Wydaje się zatem, że opracowując model zarządzania projektami informatycznymi należy uwzględnić w nim wszystkie problemy związane z doбором metod i dobrych praktyk zarządzania projektem omówione w rozdziale I, jak również zagadnienia zmiennych decyzyjnych zaprezentowane w tym rozdziale. Podstawowym wnioskiem płynącym z analizy tych dwóch głównych zagadnień (czyli doboru metod i dobrych praktyk zarządzania projektami do danego projektu oraz analizy nowego trójkąta ograniczeń) jest to, że należy opracować taki model zarządzania projektem informatycznym, który będzie uwzględniał owe zagadnienia. Stąd też zrodziła się koncepcja opracowania adaptacyjnego agentowego modelu zarządzania projektem informatycznym, którego nadrzędnym zadaniem będzie dokonywanie pomiaru wartości zmiennych w nowym trójkącie ograniczeń oraz dopasowywanie dobrych praktyk zarządzania projektami w oparciu o te ograniczenia. Założenia do budowy takiego modelu zostały przedstawione w rozdziale III.

Rozdział III

Założenia do budowy adaptacyjnego agentowego modelu zarządzania projektami informatycznymi

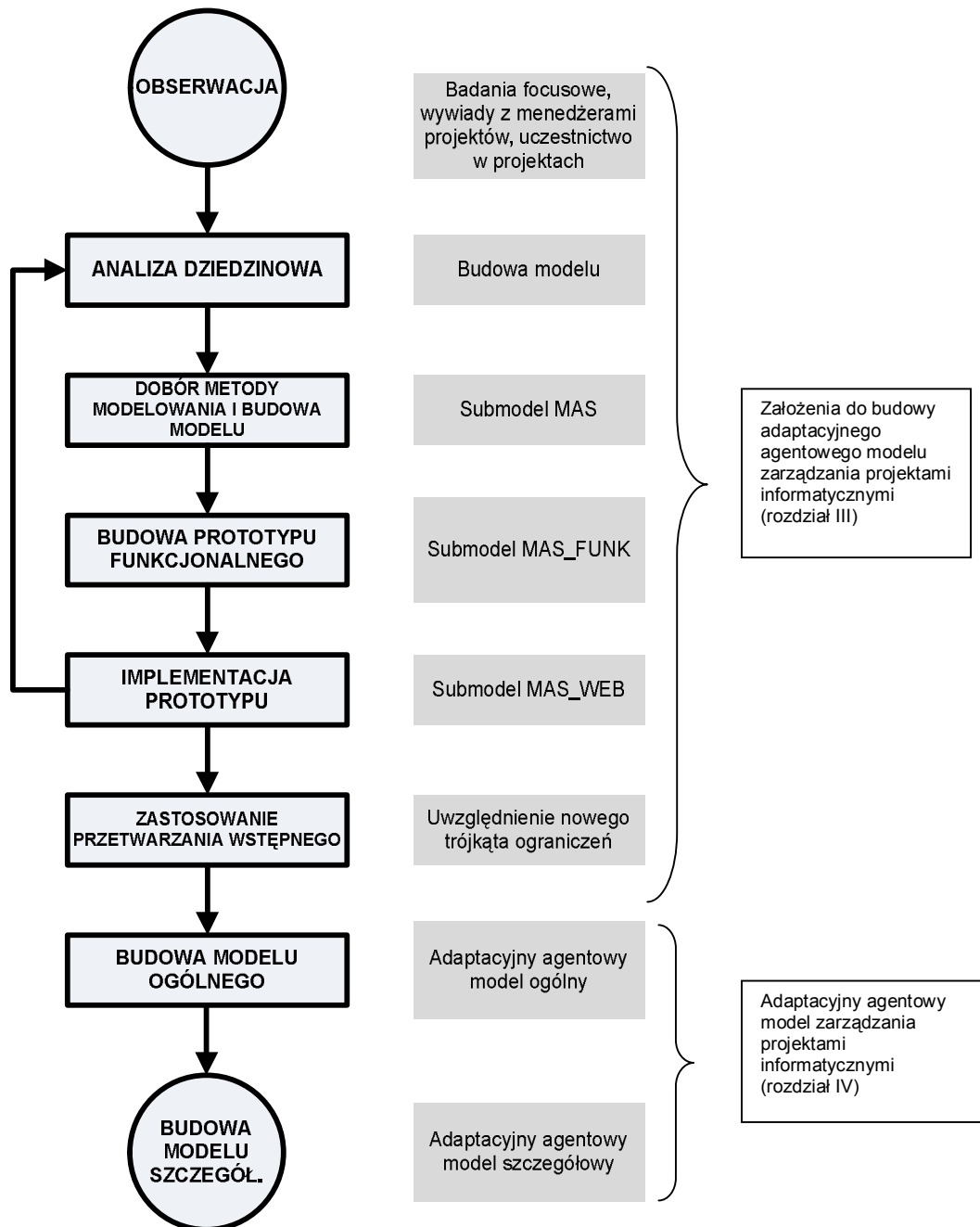
Na podstawie wniosków wyciągniętych z analiz przeprowadzonych w dwóch poprzednich rozdziałach zauważono brak sformalizowanych rozwiązań pozwalających kierownikom projektów informatycznych dobierać dobre praktyki zarządzania projektami w zależności od struktury projektowej oraz zmiennych decyzyjnych nowego trójkąta ograniczeń. W poprzednich dwóch rozdziałach zauważono również, że podejmowanie decyzji przez kierowników projektów informatycznych odnośnie tego, które dobre praktyki zarządzania projektem stosować, jest problemem bardzo złożonym. Uznano więc, że po dekompozycji metod zarządzania projektami do postaci dobrych praktyk oraz przyjęciu założenia, że kierownicy projektów powinni dobierać pojedyncze dobre praktyki w miejsce całych metod zarządzania projektami, należy opracować model, który będzie wspierał kierowników w doborze dobrych praktyk do projektów. Uznano również, że z uwagi na fakt, że stosowanie metod zarządzania projektami okazuje się nieefektywne, należy opracować taki model, który umożliwi adaptowanie dobrych praktyk w zależności od wartości zmiennych decyzyjnych nowego trójkąta ograniczeń. W związku z tym zrodziło się pytanie, jak wspomagać dobieranie tych dobrych praktyk przez kierowników projektów. Zgodnie z teorią zarządzania mówiącą, że jedną z głównych cech dobrego menedżera jest posiadanie odpowiedniej intuicji, wydawałoby się, że sama znajomość dobrych praktyk przez kierownika projektu będzie wystarczająca do ich dopasowywania do projektu. Jednak aktualny stan wiedzy o zarządzaniu projektami informatycznymi wskazuje, że podejmowanie decyzji kierowniczych w projektach informatycznych jest zagadnieniem na tyle złożonym, że nie powinno opierać się wyłącznie na intuicji. Potwierdza to także rozwój metod zarządzania projektami, ich ciągłe ewolucje, jak również pojawianie się nowych metod zarządzania projektami oraz narzędzi informatycznych wspierających kierowników w działaniu. Podejmowanie decyzji co do sposobu kierowania projektem w oparciu o dobre praktyki powinno być więc wsparte odpowiednim rozwiązaniem formalnym. Uznano zatem, że należy opracować taki model zarządzania projektem informatycznym, który będzie łączył zarówno problematykę pomiaru zmiennych decyzyjnych (nowego trójkąta ograniczeń), jak również problematykę doboru (adaptacji) dobrych praktyk zarządzania projektami informatycznymi w zależności od tych zmiennych decyzyjnych. Celem niniejszego rozdziału jest zatem omówienie podstawowych założeń do budowy takiego modelu. W rozdziale niniejszym zaprezentowano ogólną koncepcję modelu zarządzania projektami informatycznymi, przedstawiono możliwe podejścia do budowy modelu oraz wybrano podejście najbardziej odpowiadające koncepcji wsparcia kierowników projektów informatycznych w doborze dobrych praktyk do ograniczeń projektowych. W rozdziale omówiono również metodę prowadzenia badań nakierowanych na powstanie adaptacyjnego agentowego modelu zarządzania projektami, którego opracowanie jest głównym celem niniejszej rozprawy. Badania prowadzące do powstania ostatecznej wersji modelu przedstawiono w postaci etapów rozwojowych, które kończyły się stworzeniem pośrednich instancji modelu określonych mianem submodeli.

3.1 Koncepcja budowy modelu

Zaobserwowany wcześniej problem stosowania metod zarządzania projektami i dobrych praktyk w nich zawartych skłonił do podjęcia badań nad poszukiwaniem modelu pozwalającego usprawnić zarządzanie projektami informatycznymi. Uznano, że w celu wsparcia kierowników projektów informatycznych odnośnie doboru dobrych praktyk zarządzania projektami do ograniczeń

projektowych (wynikających z uprzednio zdefiniowanego nowego trójkąta ograniczeń) należy opracować model agentowy. Podjęto decyzję o konieczności budowy właśnie modelu, a nie typowej metody zarządzania projektami z dwóch powodów. Po pierwsze model jest pojęciem bardziej ogólnym, przez co możliwe jest integrowanie większej liczby elementów, które są istotne z punktu widzenia zarządzania projektami informatycznymi. Drugim powodem skłaniającym do poszukiwania właśnie modelu, a nie metody, jest wniosek płynący z prezentacji narzędzi informatycznych wykorzystywanych w zarządzaniu projektami informatycznymi (dokonany w rozdziale I). Skoro bowiem narzędzie informatyczne jest nieodłącznie związane z pracą kierowników projektów, opracowując model można dokonać późniejszej jego implementacji właśnie w przykładowym narzędziu, dając w ten sposób kierownikom dodatkowe wsparcie przy podejmowaniu decyzji w projekcie. Zgodnie z obecnym podejściem architektur MDA (ang. *Model Driven Architecture*) [10,59] jeżeli model odzwierciedla cechy narzędzia i metody, jest niezależny od środowiska implementacji i może być zastosowany w dowolnym środowisku. Tworząc model, a nie metodę, tworzone jest rozwiązanie o charakterze uniwersalnym.

Droga do opracowania adaptacyjnego agentowego modelu zarządzania projektami została poprzedzona analizą dziedzinową. Analiza dziedzinowa rozumiana jest jako badanie stanu 12 projektów informatycznych, w których autor miał okazję uczestniczyć. Dodatkowo analizę tę uzupełniały spostrzeżenia innych kierowników projektów informatycznych, którzy podczas zogniskowanych wywiadów dzielili się swoimi doświadczeniami zdobytymi podczas realizacji własnych projektów informatycznych. Obserwacje płynące z uczestnictwa w wymienionych projektach oraz wiedza zdobyta na podstawie wywiadów z kierownikami projektów pozwoliły na określenie kierunku prac badawczych. Przed zaprezentowaniem ostatecznej koncepcji adaptacyjnego agentowego modelu zarządzania projektami informatycznymi postanowiono więc przedstawić także efekty (produkty) pośrednie prowadzonych badań, zwane submodelami. Każdy bowiem submodel pozwalał na zdobycie doświadczeń prowadzących do powstania ostatecznej wersji modelu. Zarówno przy pracach koncepcyjnych (prototypowanie), jak również przy opracowaniu ostatecznej wersji adaptacyjnego agentowego modelu zarządzania projektami informatycznymi (prezentowanego w rozdziale następnym), przyjęto metodę badawczą przedstawioną na rysunku 3.1.



Rys. 3.1 Metoda prowadzenia badań dla uzyskania modelu

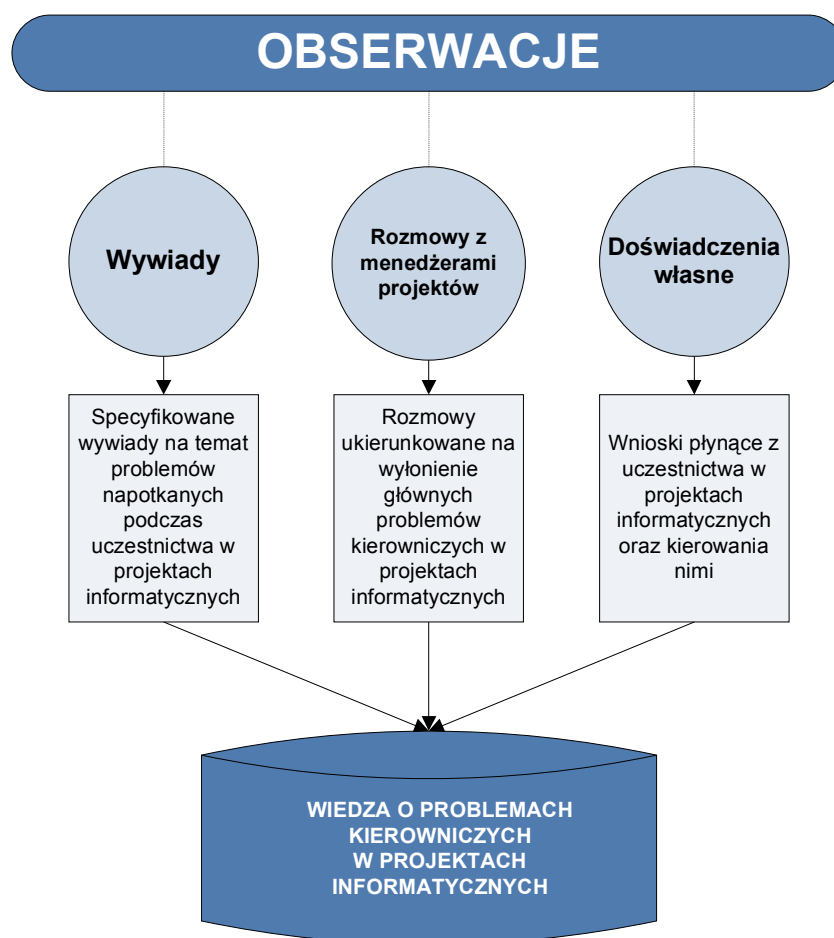
Źródło: opracowanie własne

Jak wynika z rysunku 3.1, punktem wyjścia do prowadzonych badań były wyniki obserwacji poczynionych podczas uczestniczenia w realizowanych projektach informatycznych. Na obserwacje składały się trzy podstawowe działania. Pierwsze stanowiły zogniskowane badania prowadzone z uczestnikami projektów, podczas których autor zadawał pytania istotne z punktu widzenia zarządzania projektami. Drugim źródłem pozyskiwania wiedzy były wywiady prowadzone z menedżerami projektów (głównie podczas konferencji branżowych i spotkań biznesowych). Jako, że jednostka naukowo-badawcza jaką jest Zakład Zarządzania Technologiami Informatycznymi, w której pracuje autor niniejszej rozprawy, współpracuje z wieloma korporacjami informatycznymi (jak IBM Polska, HP Polska czy Intel Polska) oraz organizacjami z innych branż, ale realizujących szereg projektów informatycznych na własne potrzeby (np. GE Money Bank), wywiady przeprowadzono głównie z kierownikami projektów informatycznych realizowanych w tych firmach. Dzięki temu

zgromadzona została wiedza na temat specyfiki prowadzenia projektów przez tych kierowników oraz wiedza o podstawowych problemach tych firm. Dodatkowo, w celu uzupełnienia wiedzy o projektach realizowanych w mniejszych firmach, zasięgnięto opinii kierowników projektów z niedużych firm świadczących usługi informatyczne.

Trzecim źródłem pozyskiwania wiedzy były własne doświadczenia autora płynące z realizacji trzech projektów informatycznych (m.in. na potrzeby firmy sektora energetycznego oraz projektu finansowanego przez Ministerstwo Nauki i Szkolnictwa Wyższego w ramach grantu badawczego).

Ten zbiór wiedzy oparty na wywiadach i własnych doświadczeniach (zgodnie z rysunkiem rys. 3.2) stanowił punkt wyjścia do opracowywania modelu pozwalającego wspomagać menedżerów projektów informatycznych w obszarze podejmowania decyzji co do metod zarządzania projektami oraz tworzenia poprawnych struktur projektowych.



Rys. 3.2 Źródła wiedzy o projektach informatycznych wykorzystane w opracowaniu modelu a2M

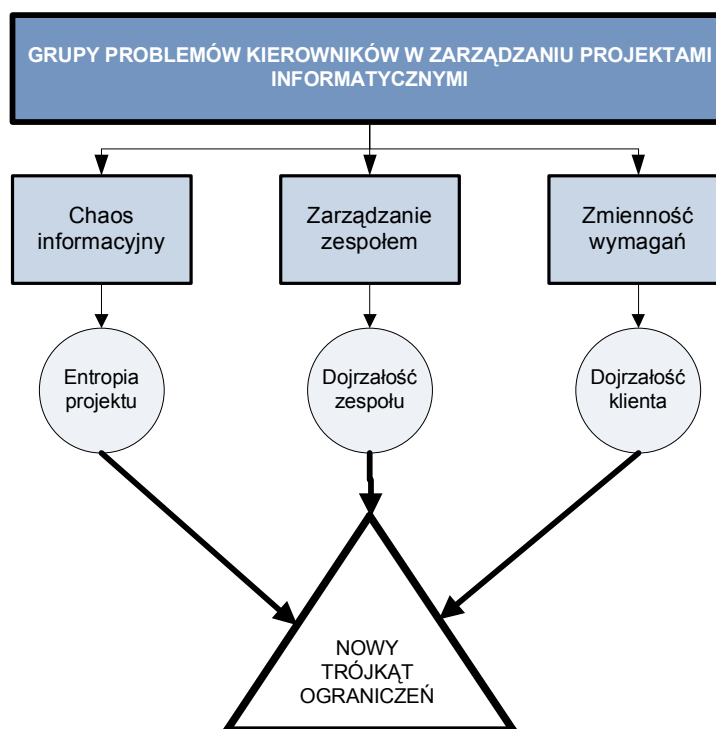
Źródło: opracowanie własne

Po zgromadzeniu odpowiedniej wiedzy o realizowanych projektach (w ramach obserwacji) zgodnie z przyjętą metodą badawczą (rys. 3.1) przystąpiono do opracowywania submodeli. Submodele poddawano weryfikacji poprzez dedykowane im eksperymenty przeprowadzane w określonym, uprzednio zdefiniowanym środowisku. Po każdym z eksperymentów wyciągano wnioski oraz uogólnienia pozwalające na tworzenie kolejnego submodelu zgodnie z podejściem iteracyjnym (rys. 3.1). Poniżej omówione zostały poszczególne etapy rozwojowe począwszy od analizy dziedzinowej, poprzez kolejne etapy prowadzące do tworzenia prototypów (submodeli). Każdy prototyp stanowił kolejne przybliżenie do opracowania adaptacyjnego agentowego modelu zarządzania projektami informatycznymi.

3.2 Analiza dziedzinowa

Podczas przeprowadzania obserwacji rzeczywistych projektów informatycznych gromadzono wiedzę o metodach zarządzania i dobrych praktykach kierowników projektów informatycznych (podobnie jak wykonano to w rozdziale I). Proces przetwarzania wiedzy (na potrzeby budowy modelu) został określony mianem **analizy dziedzinowej**. Analiza dziedzinowa stanowi zatem pierwszy etap rozwojowy budowy modelu wspomagającego zarządzanie projektami informatycznymi. Pozostałe etapy omówione w dalszej części prowadziły do powstania poszczególnych modeli pośrednich, sprowadzając modele do postaci systemu agentowego realizującego zamodelowane funkcjonalności.

Analiza dziedzinowa stanowiła więc podstawowy etap badań nad problematyką projektów informatycznych, w efekcie których rozpoczęto opracowywanie submodeli prowadzących do powstania późniejszego adaptacyjnego agentowego modelu zarządzania projektami informatycznymi. Zgromadzona uprzednio wiedza dotycząca problemów kierowników w zarządzaniu projektami informatycznymi (wykorzystywania dobrych praktyk zarządzania projektami) posłużyła w tym etapie do zdefiniowania zagregowanych zmiennych dojrzałości klienta i dostawcy oraz entropii projektu. Zmienne te wraz z opisami zostały zaprezentowane na rysunku 3.3. Do opisu poszczególnych zmiennych posłużono się odpowiednimi przykładami zaczerpniętymi z rzeczywistych projektów informatycznych.



Rys. 3.3 Transformacja głównych problemów w projektach na zmienne decyzyjne

Źródło: opracowanie własne

Poniżej omówiono podstawowe problemy związane ze zmiennymi decyzyjnymi wchodzącymi w skład nowego trójkąta ograniczeń.

Grupa problemów związanych z chaosem informacyjnym w projektach a entropia projektu

Jednym z podstawowych problemów wskazywanych podczas rozmów przez kierowników projektów okazał się być chaos informacyjny. Zwrócono uwagę, że największe nieuporządkowanie

informacji występuje z reguły przed rozpoczęciem projektu. Niektórzy kierownicy projektów informatycznych przyznali, że oczekuje się od nich rozpoczęcia prac projektowych przy niewystarczającej liczbie danych dotyczących projektu. Ten niewystarczający poziom informacji dotyczył zarówno celów, jakie stawiano zespołowi realizującemu projekt, jak również specyfiki produktu, jaki miał powstać w wyniku prac projektowych. Poniższe przykłady zaprezentowano w celu potwierdzenia tezy, że chaos informacyjny można uznać za jedną z głównych grup problemów, z jakimi borykają się kierownicy projektów i że chaos informacyjny przekłada się na entropię projektu informatycznego.

Przykład 3. Brak jasno wyznaczonych celów projektu a zarządzanie zespołem

Projekt: Integracja technologii Rational

Klient: IBM

Zespoły: studenci ostatniego roku studiów, specjalności Zarządzanie Technologiami Informatycznymi,

Termin realizacji: kwiecień 2010–lipiec 2011

W ramach tego projektu powołano kilka niezależnych zespołów projektowych. Każdy zespół zajmował się możliwościami integracyjnymi innych narzędzi z rodziny Rational. Na początku projektu jego cel nie był jasny dla członków zespołów projektowych. Wynikało to między innymi z faktu, że pojęcie integracji nie zostało dokładnie sprecyzowane (nie było wiadomo, czy integracja ma następować na poziomie danych, czy na wyższym poziomie analizowanych narzędzi). Członkowie zespołów nie umieli tym samym określić zadań analitycznych (związanych z rozpoznawaniem technologii) i zwracali uwagę na duży dyskomfort spowodowany brakiem wizji ostatecznego celu projektu. Menedżerowie prowadzący ten projekt również nie potrafili precyzyjnie tych celów wskazać. Zarys finalnego produktu projektu integracji był co prawda przedstawiony (technologie miały być zintegrowane w cyklu wytwarzania oprogramowania), ale uczestnicy projektu nie potrafili dokładnie zrozumieć, jak ma ten finalny produkt wyglądać. Brak wystarczających danych o celach realizowanego projektu przyczynił się do dużego zniechęcenia i spadku poziomu motywacji wśród wykonawców (niektórzy członkowie zespołu chcieli się wycofać z realizacji zadań). Na podstawie tego przykładu okazuje się, że niedobór informacji rodzi szereg problemów zarówno dla osób kierujących projektami, jak również dla wykonawców, a planowanie zadań w warunkach niskiego poziomu informacji jest znacznie utrudnione.

Podsumowanie:

- projekt o niskim stopniu precyzji sformułowania wymagań,
- projekt o niskim stopniu pewności zdefiniowanego zakresu projektu,
- projekt o wysokim stopniu złożoności zadań projektowych,
- projekt o wysokim ryzyku wykonawczym.

Ocena: Projekt o wysokiej entropii.

Przykład 4. Wpływ nadmiaru informacji na realizację zadań w projekcie informatycznym

Klient: Politechnika Gdańska

Dostawca: ZETO Olsztyn

Projekt: wdrożenie aplikacji webowej do zarządzania publikacjami naukowymi

Termin realizacji: styczeń-kwiecień 2010

Drugim przykładem obrazującym problematykę chaosu informacyjnego jest obserwacja poczyniona podczas uczestnictwa w projekcie realizowanym na potrzeby jednej z uczelni wyższych. Celem projektu było wytworzenie systemu wspomagającego zarządzanie publikacjami naukowymi i ich udostępnianie środowisku biznesowemu. Jednym z podstawowych problemów związanych z realizacją zadań w tym projekcie był nadmiar informacji. Prace nie były wykonywane terminowo głównie ze względu na różnorodność informacji (często sprzecznych) napływających do osób wytwarzających system. Informacje napływały z dwóch źródeł umiejscowionych na różnych szczeblach w strukturze organizacyjnej, przez co wykonawcy nie byli w stanie określić, które informacje są ważniejsze. Stąd też nadmiarowość i nieuporządkowanie informacji stanowiło również problem dla kierowników projektów oraz wykonawców zadań w projekcie. Napływ nowych informacji z różnych źródeł stanowił w omawianym projekcie poważne wyzwanie dla kierownika projektu, który zmuszony był do porządkowania zadań dla swojego zespołu wykonawców. W sytuacji, w której informacje są niepełne, nieprecyzyjne i występują w bardzo dużej ilości, wyłonienie informacji kluczowych z punktu widzenia produktu stało się poważnym problemem dla menedżerów.

Podsumowanie:

- projekt o niskim stopniu precyzji sformułowania wymagań,
- projekt o niskim stopniu pewności zdefiniowanego zakresu projektu,
- projekt o średnim stopniu złożoności zadań projektowych,
- projekt o średnim ryzyku wykonawczym.

Ocena: Projekt o średniej entropii.

Na podstawie powyższych przykładów można stwierdzić, że brak analizy poziomu entropii własnego projektu przez menedżera projektu zazwyczaj skutkuje negatywnie z punktu widzenia realizacji zadań. Łatwo zauważyć więc, że jednym z kluczowych działań, jakie musi przeprowadzić kierownik projektu, jest zmniejszenie albo przynajmniej określenie poziomu entropii projektowej poprzez np. uporządkowanie informacji niezbędnych do realizacji projektu.

Grupa problemów związanych z funkcjonowaniem zespołu a dojrzałość zespołu dostawcy

Drugim ważnym problemem związanym z realizacją projektów informatycznych jest problem budowy zespołu realizującego projekt — zespołu dostawcy. Problem ten wpisuje się w klasyczny dla nauk o zarządzaniu problem zarządzania zasobami ludzkimi. Jednakże w przypadku projektów informatycznych nabiera szczególnego znaczenia. Jak wspomniano w rozdziale I, zespoły w projektach informatycznych charakteryzują się tym, że występuje w nich wyraźne rozróżnienie ról i wąska specjalizacja zadań. Stąd też dobór członków zespołu projektowego, zbudowanie poprawnych relacji pomiędzy tymi członkami oraz zapewnienie właściwej komunikacji stanowią podstawowe wyzwanie dla menedżerów projektów informatycznych.

Przykład 5. Wpływ dojrzałości zespołu na realizację zadań w projekcie informatycznym

Projekt: Integracja technologii IBM Clear Case oraz IBM Clear Quest

Klient: Avnet

Zespoły: studenci ostatniego roku studiów specjalności Zarządzanie Technologiami Informatycznymi,

Termin realizacji: 2009–2010

W projekcie dotyczącym zastosowania technologii IBM Clear Case oraz IBM Clear Quest do wspomagania zarządzania incydentami w oprogramowaniu klienta projektu wystąpiła konieczność integracji tych technologii z innymi technologiami wykorzystywanymi u klienta. W trakcie obserwacji

zespołów realizujących projekt zwrócono uwagę na wyraźne problemy dotyczące komunikacji pomiędzy członkami zespołu. W projekcie powołano trzy zespoły. Zespoły te miały zorganizować swoją pracę poprzez odpowiedni podział zadań. Powołane zespoły składały się z osób o małym doświadczeniu projektowym. Efektem takiego podejścia było to, że na samym początku realizacji projektu zaobserwowano problemy z podziałem zadań pomiędzy poszczególnych członków. Liderzy zespołów nie potrafili jasno zdefiniować zadań i zlecić ich wykonania pozostałym członkom zespołu. Kontrola przeprowadzona po kilku tygodniach od uruchomienia projektu pokazała, że prace projektowe nie są realizowane terminowo. Podczas wywiadów z członkami zespołów najczęściej zwracano uwagę na niski poziom komunikacji pomiędzy liderami a pozostałymi członkami, przez co wykonawcy zadań nie do końca wiedzieli, co powinni zrobić. Liderzy natomiast zapytani o sposób prowadzenia prac, czyli o zastosowaną metodę zarządzania projektem (wcześniej byli szkoleni z formalnych metod zarządzania projektami), wskazali na problem dopasowania takiej metody do swoich zespołów. Po spotkaniu kontrolnym z liderami włączono do projektu dwie osoby z większym doświadczeniem w zarządzaniu projektami, które pomogły liderom w doprecyzowaniu zadań dla poszczególnych członków. Od tego momentu prace były realizowane terminowo. Podobna obserwacja potwierdziła się również w kilku innych badanych zespołach projektowych, gdzie uczestnicy z niewielkim stażem projektowym mieli zazwyczaj duże problemy z właściwym definiowaniem zadań, a młodzi menedżerowie nie potrafili dobrać metody zarządzania projektami, co znacznie utrudniało realizację zadań projektowych.

Dalsze obserwacje wynikające z udziału w projektach informatycznych pokazały, że na podobne problemy, jak opisane w przykładzie piątym, zwraca uwagę większość menedżerów IT, z którymi prowadzono wywiady. Nie oznacza to oczywiście, że zespoły złożone z niedoświadczonych członków nie powinny realizować projektów informatycznych. Ważne jest jednak, aby w zależności od poziomu dojrzałości zespołu dostawcy dopasowywać dobre praktyki zarządzania oraz zapewnić odpowiednią organizację zespołu — czyli taki podział ról, który odpowiada kompetencjom i poziomowi doświadczenia poszczególnych członków. Ta obserwacja pokazała również, że poziom dojrzałości zespołu powinien być uwzględniany podczas tworzenia zespołów projektowych i w zależności od tego poziomu menedżerowie powinni dobrać dobre praktyki zarządzania projektem. Wydaje się również słuszne, aby poziom dojrzałości zespołu dostawcy poddawać stałej kontroli, a zespoły powinny dobrać sposoby realizacji prac projektowych w zależności od poziomu swojej dojrzałości.

Powyższy przykład wskazał również na dodatkowy czynnik, który wydał się być nieodzowny w zarządzaniu projektem. Mowa tutaj o obecności ekspertów dziedzinowych, osób z doświadczeniem i kompetencjami, które mogą wspomagać menedżerów w zarządzaniu projektem. Jak pokazał przykład, młody zespół (złożony z niedoświadczonych uczestników i liderów o niedużym doświadczeniu w zarządzaniu projektami) czy też początkująca firma informatyczna może na pewnych etapach realizacji projektu potrzebować konsultacji z ekspertami dziedzinowymi. Włączenie do projektu osób o dużym doświadczeniu projektowym opisane w przykładzie piątym usprawniło realizowanie zadań.

Definicja 3.1

Ekspert dziedzinowy rozumiany jest jako doradca, osoba, która służy pomocą zespołowi, wypowiadająca swoje opinie, wskazująca kierunki lub sugerująca rozwiązania. Charakteryzuje się dużym doświadczeniem w danej dziedzinie, umiejętnościami interpersonalnymi oraz gruntowną wiedzą z dziedziny, z którą związany jest projekt lub elementy projektu.

Ekspert dziedzinowy stanowi element struktury projektu informatycznego, co oznacza, że menedżerowie projektów powinni określać zapotrzebowanie na takich ekspertów. Opisany wcześniej przykład wyraźnie pokazał, że dzięki uwzględnieniu eksperta w projekcie, udało się zespołowi

rozwiązać bieżące problemy z planowaniem zadań. Zastosowanie obecności ekspertów w projektach staje się coraz częstszym zjawiskiem i wpisuje się w bieżące w naukach o zarządzaniu zjawisko mentoringu (albo nawet coachingu). [7]. Na potwierdzenie istotności ekspertów w projektach można podać przykład firmy IBM Polska, gdzie podczas realizowanych projektów przydziela się do zespołu tzw. mentora, który służy swoją wiedzą w sytuacji, gdy zespół napotyka problemy podczas realizacji zadań. Stąd też coraz większy nacisk należy położyć na uwzględnianie takich właśnie ekspertów w strukturze projektu. Ekspert nie musi być osobą zatrudnioną na stałe do projektu, ale powinien być dostępny w sytuacji wymaganej przez zespół.

Grupa problemów związanych ze zmianami w projekcie a dojrzałością klienta

Trzecią grupą problemów dostrzeżonych w trakcie uczestnictwa w projektach informatycznych, jak również w trakcie prowadzonych rozmów z menedżerami projektów, jest dynamika zmian w trakcie realizacji projektu. Najczęstszą przyczyną niepożądanego zjawiska zmienności w projektach, na którą wskazali menedżerowie, są nieporozumienia podczas gromadzenia wymagań klienta. Co prawda dzisiejsze podejście do klienta w projektach informatycznych charakteryzuje duża elastyczność (Manifest Zwinności omawiany w rozdziale I dosłownie wskazuje w punkcie „chętnie widzimy zmiany w projekcie” na potrzebę dużej elastyczności względem klienta projektu informatycznego), to jednak każda zmiana wymagań projektowych implikuje szereg zmian w przebiegu całego projektu. Tym samym skutkuje dla kierowników projektów koniecznością podejmowania kolejnych decyzji dotyczących dalszej realizacji projektu. Każda zmiana wiąże się z koniecznością reorganizacji przyjętych ram pracy (w postaci np. harmonogramu albo planu sprintu). Prowadzi również do zniechęcenia zespołu wytwórczego. Problemem jednak nie wydaje się samo występowanie zmienności, ale raczej trudności w jej przewidzeniu. Jeżeli zespół będzie przygotowany na zmiany, będzie łatwiej radził sobie z dynamiką zmian w projekcie.

Przykład 6. Wpływ zmienności wymagań na realizację zadań w projekcie informatycznym

Projekt: Budowa systemu informatycznego do obsługi rozliczeń z klientami

Klient: Firma konsultingowa LeanTeam

Zespół: Studenci kierunku informatyka i ekonometria oraz ich opiekun naukowy

Termin realizacji: 2010–2011

Budowany system miał klasyczną architekturę trójwarstwową, typową dla aplikacji bazodanowych (czyli dane, logika biznesowa, interfejs). Przedstawiciel desygnowany przez firmę nie posiadał wystarczającej wiedzy z zakresu projektów informatycznych. Nakreślił analitykowi szczegółowe zadania, jakie miały być zrealizowane w trakcie budowy systemu. Okazało się jednak, że w trakcie realizacji systemu wymagania zmieniły się diametralnie, podczas gdy wytwórcy systemu wydawało się, że zakres przygotowany na początku budowy systemu jest kompletny. Jedną z zmian w wymaganiach pociągała za sobą reorganizację struktury tabel w budowanej aplikacji, co dla dostawcy oznaczało wytwarzanie baz od nowa. Ta nagła zmiana wymagań wynikająca z niezrozumienia przez klienta specyfiki prowadzenia projektu informatycznego doprowadziła do trzymiesięcznego opóźnienia w dostarczeniu kompletnego rozwiązania. Jednocześnie brak uwzględnienia zmian wymagań przez wytwórcę spowodował duże problemy w realizacji projektu do tego stopnia, że wytwórca chciał odstąpić od jego realizacji.

Problematyka związana z dynamiką zmian wskazuje jednoznacznie na konieczność bardzo dogłębnej analizy dojrzałości klienta. Nie oznacza to oczywiście, że zmienność w projekcie powinna być eliminowana. Natomiast powinna być przez menedżera przewidywana. Prognozy zmienności można dokonać wtedy, gdy klient zostanie poddany odpowiedniej analizie prowadzącej do określenia jego poziom dojrzałości. Stąd też przed rozpoczęciem realizacji projektu menedżer powinien znać

poziom dojrzałości klienta, aby wiedzieć, na ile możliwe jest generowanie przez niego zmian w projekcie. Taka wiedza pozwoli na lepsze dostosowanie sposobu prowadzenia prac (dobrych praktyk zarządzania projektem), jak również pozwoli lepiej przygotować zespół na nieprzewidziane zmiany.

Podsumowanie etapu związanego z analizą dziedzinową

Jak łatwo zauważyć, większość problemów menedżerskich obserwowanych w projektach oraz wskazanych przez kierowników podczas wywiadów można zaliczyć do problemów związanych z dojrzałością klienta/dostawcy lub entropii projektu (omawianych w rozdziale II). Tym samym można zauważyć, że to właśnie te trzy zagregowane zmienne mają istotny wpływ na późniejsze decyzje dotyczące sposobu realizacji zadań projektowych. Oznacza to, że kierujący projektami w oparciu o wprowadzanie nowego trójkąta ograniczeń powinni dokonywać doboru dobrych praktyk zarządzania czy konstruowania zespołu. Analiza i pomiar tych zmiennych dla dowolnego realizowanego projektu powinny poprzedzać jego realizację w oparciu o klasyczny istniejący trójkąt ograniczeń (zakres, budżet, harmonogram). Wyodrębnienie tych trzech zmiennych (rozdział II) i analiza możliwości ich oceny pozwoliło zakończyć etap analizy dziedzinowej.

Jak zaznaczono we wstępie do niniejszego rozdziału, jego celem jest pokazanie poszczególnych etapów budowy adaptacyjnego agentowego modelu zarządzania projektami informatycznymi. Na podstawie zaprezentowanej analizy dziedzinowej rozpoczęto prace nad doбором metody modelowania istniejącego trójkąta ograniczeń.

3.3 Dobór metody modelowania

Najczęściej stosowanym podejściem do modelowania systemów społeczno-technicznych (zarządzanie projektem informatycznym) jest **podejście procesowe**. Podejście procesowe zakłada, że wszelkie działania konstrukcyjne prowadzą do modelu opierającego się na czynnościach wymagających na wejściu określonego, zdefiniowanego wkładu, mechanizmów przetwarzania oraz określonych rezultatów na wyjściu. Rezultaty powinny natomiast stanowić nową wartość. Cechą podejścia procesowego jest również obecność procesów pomocniczych wspierających realizację procesu głównego, jak również konieczność kwantyfikacji poszczególnych parametrów procesu — zarówno podawanych na wejściu, jak i tych wyjściowych.

Drugim podejściem do modelowania rozważanym w tej pracy jest **podejście agentowe**, które opiera się na wyodrębnieniu z rzeczywistości określonych encji opisywanych w postaci stanów i akcji, jakie mogą być podejmowane w określonym celu. Agenty są typowym rozwiązaniem zaliczanym do rozwiązań opartych na sztucznej inteligencji.

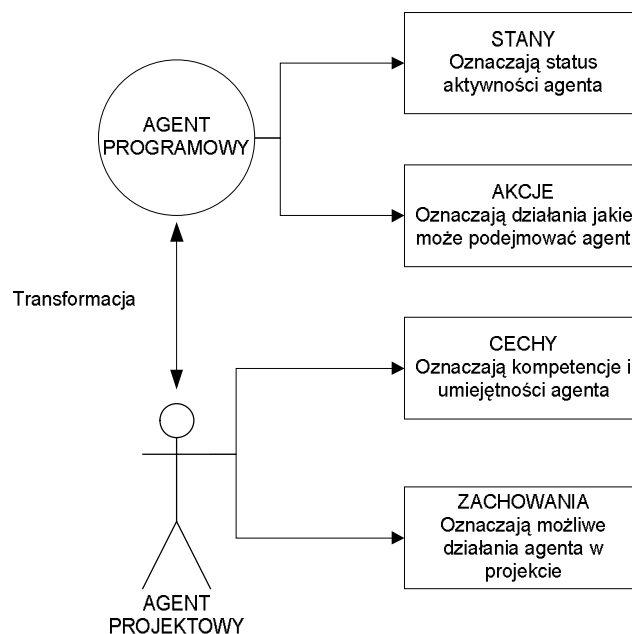
W celu przybliżenia podejścia agentowego do modelowania systemów poniżej przedstawiono podstawowe cechy agentów i systemów agentowych. Jest to konieczne także z punktu widzenia ostatecznego modelu nazwanego w niniejszej rozprawie adaptacyjnym agentowym modelem zarządzania projektami, który wykorzystuje właśnie podejście agentowe.

Agent programowy

Na przestrzeni lat powstało wiele definicji agentów i systemów agentowych[3,43,78, 100]. W 1995 Russel pisał o agentach jako „czymkolwiek, co może być uznane jako obserwujące otoczenie poprzez sensory i działające w ramach tegoż otoczenia poprzez efekторы”. Druga ze znanych definicji (opracowana przez IBM) mówi o agentach jako jednostkach programowych, które podejmują działania w imieniu użytkownika lub innych programów i które są w pewnym stopniu niezależne i autonomiczne oraz stosują w swoim działaniu pewną wiedzę lub reprezentację celów i potrzeb użytkownika[13,82]. W pracy Wooldridge’a z 1997 roku można przeczytać o systemie agentowym

jako pewnym zamkniętym systemie komputerowym znajdującym się w pewnym otoczeniu, który to (system) posiada umiejętności elastycznego działania w tymże otoczeniu oraz działania polegającego na wypełnieniu celów, dla jakich został stworzony. Najbardziej precyzyjną definicją systemu agentowego spotykaną w literaturze[2,41,92] jest powstała w 1998 definicja Maesa, mówiąca o systemie agentowym jako autonomicznym systemie znajdującym się w dynamicznym otoczeniu, działającym niezależnie od narzucanych przezeń ograniczeń i wypełniającym w jego ramach zbiór celów lub poleceń, dla których został stworzony.

Przez systemy wieloagentowe (ang. *multiagent systems*) należy rozumieć systemy komputerowe oparte i zbudowane na bazie programów — agentów. Tworzenie takich systemów — programowanie agentowe — polega na stworzeniu zbioru programów agentowych i środowiska do ich działania [6,11,25,81]. Agentom w systemach agentowych przypisuje się określony zbiór własności, które są charakterystyczne dla każdego agenta. Do takich cech należy zaliczyć m.in. reaktywność, ukierunkowanie na osiąganie celów czy autonomię. Autonomia oznacza, że dany agent powinien wykonywać zlecone czy przypisane mu zadania samodzielnie. Mówiąc o zlecaniu zadań, należy także wymienić kolejną cechę systemu agentowego, czyli działanie w imieniu zlecającego. Mówi się wtedy o agencie jako delegacie, działającym w imieniu użytkownika, innego programu czy nawet innego agenta. Wyróżnikiem agentów są także umiejętności adaptacji (dostosowywania się do otoczenia), uczenia się i komunikacji[1,8,26]. Chodzi tu głównie o porozumiewanie się z innymi agentami (ale możliwa jest również komunikacja z użytkownikiem czy bazami danych). Dzięki takiej komunikacji działanie agenta może być skuteczniejsze [4,43,66]. Warto także nadmienić, że przed agentem często stawia się wymagania mobilności, tak aby program agentowy potrafił przemieścić się w inne miejsce (na inną maszynę, do innego środowiska) i tam kontynuował wykonywanie swoich zadań — zleconych mu działań. Konieczność współpracy agentów w różnych systemach wymaga także umiejętności ich wzajemnego współdziałania i współpracy. Zgodnie z tym co zostało powiedziane wcześniej, systemy agentowe mają służyć pozyskiwaniu wiedzy, jej przetwarzaniu i wnioskowaniu, stąd tak duże znaczenie agentów także w zarządzaniu ontologiami (czyli „bazami wiedzy o wiedzy”). Obecność tych cech w systemach agentowych pozwala stwierdzić, że uzasadnione wydawało się wykorzystywanie agentów do analizy różnych zjawisk, również takich jak realia projektów informatycznych.



Rys. 3.4 Agent programowy a agent projektowy

Źródło: opracowanie własne

W związku z przedstawionymi cechami agentów[22,30] i systemów oraz zgodnie z definicją IBM, że agentem może być każda jednostka, która działa w ściśle określonym celu na bazie pewnych przypisanych parametrów, agenty programowe można odnieść do struktur projektów informatycznych.

Skoro bowiem każdy agent programowy reprezentowany jest przez atrybuty oznaczające jego stany, które może przyjmować oraz akcje, jakie podejmuje w imieniu zlecającego, istnieje analogia do uczestników projektów. Uczestnicy projektów bowiem również posiadają określone cechy (odpowiednik stanów agenta programowego) oraz przejawiają określone zachowania (np. reagują na sygnały mówiące o konieczności wykonania danego zadania). Taka transformacja stanów i akcji agentów programowych do cech i zachowań uczestników projektów informatycznych pozwala uznać podejście agentowe za odpowiednie do modelowania struktur projektów informatycznych. Na tej bowiem podstawie można uczestników projektów określić mianem agentów (projektowych), których można opisać cechami i zachowaniami. Stosując takie podejście należy uznać, że koncepcja agentów programowych zostaje przeniesiona na płaszczyznę zarządzania w projektach informatycznych. Z rysunku 3.4 wynika również, że transformacja może przebiegać w drugim kierunku. Czyli znając strukturę projektu, można zamodelować ją w systemie agentowym.

Jak wynika z powyższych rozważań, należy podkreślić także, że systemy agentowe zakładają stałą relację pomiędzy poszczególnymi agentami. To również można odnieść do typowych struktur projektowych, w których wszystkie elementy powiązane są relacjami.

Przy opracowaniu systemu agentowego założono, że wspomaganie kierowników projektów w podejmowaniu decyzji kierowniczych w projektach informatycznych będzie wtedy skuteczne, gdy uwzględni się odpowiednio dużo niezależnych bytów (agentów). Dlatego też w tym etapie (doboru metody modelowania) skoncentrowano się na zdefiniowaniu wszystkich możliwych autonomicznych jednostek wnioskujących (agentów oraz modułów sztucznej inteligencji), które będą wspomagały podejmowanie decyzji.

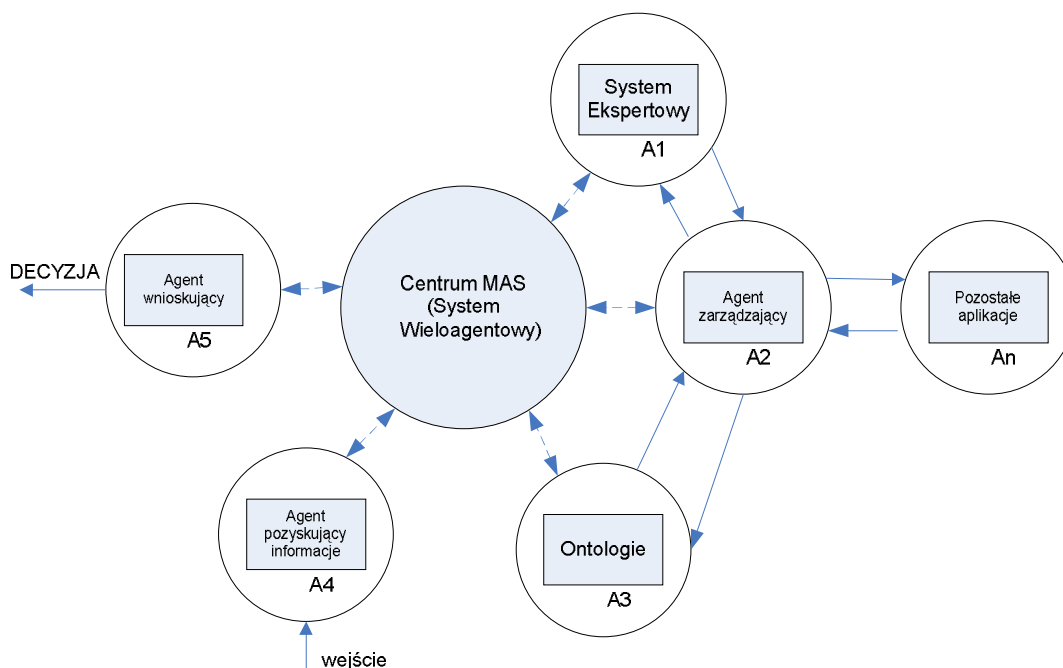
Stąd w proponowanym systemie wieloagentowym, oprócz agentów, uwzględniono dodatkowe elementy wspomagające przetwarzanie danych[29,76,83] i generowanie decyzji, takie jak:

- **System ekspertowy** — moduł wnioskujący, który na podstawie informacji zebranych o otoczeniu projektowym będzie w stanie zgromadzić dane.
- **Ontologie**, które będą czuwały nad poprawnością zadawanych pytań oraz pozyskiwanych odpowiedzi (słownik systemu).
- **Inne moduły inteligentne**, np. sztuczna sieć neuronowa, która na podstawie parametrów otrzymanych będzie w stanie dokonać analizy danych i przeprowadzić wnioskowanie.

Elementy te miały za zadanie uzupełnienie systemu agentowego o wiedzę przydatną w podejmowaniu decyzji w projektach informatycznych. Uznano, że mechanizm działania systemu będzie polegał na tym, że na podstawie kilku pozyskanych informacji od menedżerów zostanie przeprowadzony proces wnioskowania zakończony określoną decyzją. Założono na tym etapie również, że informacje będą pozyskiwane przez dedykowane agenty, a pytania i odpowiedzi tłumaczone za pomocą ontologii.

Stąd mechanizm działania systemu agentowego opierał się na funkcji pozyskiwania i przetwarzania informacji pozyskiwanych od menedżerów projektów odnośnie ograniczeń projektowych[48]. Pamiętając o cechach systemów agentowych, takich jak pozyskiwanie wiedzy, jej przetwarzanie i generowanie wniosków, uznano za konieczne zapewnić systemowi agentowemu zasilanie w wiedzę za pomocą systemu ekspertowego i ontologii. Uznano również, że w celu realizacji procesów wnioskowania (prowadzących do podania sugestii, jakie dobre praktyki oraz narzędzia wykorzystać do realizacji zadań w projekcie informatycznym) w takim systemie agentowym powinien znaleźć się także system ekspertowy, który z jednej strony przetwarza wiedzę pozyskiwaną przez agenty, a z drugiej wykorzystuje wiedzę ekspercką zawartą w bazach wiedzy do generowania wskazówek co do wykorzystania dobrych praktyk zarządzania projektami informatycznymi. Już

w trakcie pierwszych badań zauważono również konieczność uwzględnienia dodatkowych modułów mogących wspomagać generowanie decyzji, takich jak aplikacje bazujące na formalnych metodach reprezentacji wiedzy (np. aplikacji wykorzystujących modelowanie rozmyte lub sieci neuronowe umożliwiające przetwarzanie nieprecyzyjnej i niepewnej wiedzy tak typowej dla projektów informatycznych). W oparciu o te założenia powstał submodel MAS, w którym zarówno ontologie, systemy ekspertowe, jak i pozostałe aplikacje wspierające pozyskiwanie wiedzy miały wspomagać kierowników projektów w podejmowaniu decyzji. Na rysunku 3.5 została przedstawiona koncepcja struktury omawianego submodelu MAS.



Rys. 3.5 Ogólna koncepcja modelu systemu agentowego do zarządzania technologiami (architektura submodelu MAS)

Źródło: [51]

W submodelu MAS przyjęto zatem, że elementy wspierające wnioskowanie (generowanie decyzji), czyli zgodnie z założeniami — ontologie, system ekspertowy czy sztuczne sieci neuronowe, traktowane będą jako odrębne agenty całego systemu. Włączenie ich w cały system wymagało jednak zapewnienia odpowiednich mechanizmów współpracy i komunikacji. Ten wniosek wykorzystano do budowy adaptacyjnego agentowego modelu zarządzania projektami, w którym również zauważono konieczność definiowania wszystkich elementów struktury projektowej jako agenty w celu zapewnienia właściwego poziomu komunikacji między agentami.

Podczas prac nad submodelem MAS brano pod uwagę przede wszystkim tę cechę agentów, która dotyczyła samodzielnego pozyskiwania informacji i ich przetwarzania. Stąd duży ciężar podczas badań położono na rolę takich agentów, które mogą wspomagać procesy pozyskiwania takich informacji od menedżerów, które ułatwią podejmowanie decyzji co do doboru metody zarządzania projektem.

Podstawowym agentem pozyskującym informację w submodelu MAS jest przedstawiony na rysunku 3.5 agent pozyskujący informacje (A4). Zadaniem takiego agenta była bezpośrednia komunikacja z menedżerem projektu i zadanie pytań odnośnie poziomu zmiennych decyzyjnych. Pozyskane informacje powinny zostać przekazane do centrum komunikacji (Centrum MAS), gdzie agent zarządzający podejmował decyzję, który agent dokona przetwarzania (i czy będzie potrzebne uruchomienie agenta ontologii A3 lub systemu ekspertowego A1). W submodelu MAS koncentrowano się w dużej mierze także na organizacji przepływów informacyjnych, stąd wykorzystanie Centrum MAS jako bufora w komunikacji między poszczególnymi agentami. Centrum

MAS pozwalało na wymianę informacji z pozostałymi agentami (czyli np. innymi aplikacjami wspomagającymi dokonywanie ocen, takimi jak sieci neuronowe czy algorytmy genetyczne), tak aby w efekcie agent wnioskujący podejmował optymalne decyzje. Warto zwrócić w tym miejscu uwagę na fakt, iż w pierwotnej koncepcji systemu pojawiły się załączki struktury agentowej obecnej w ostatecznej wersji. Opisane centrum MAS przerodziło się w ostatecznej wersji w Agentowy System Zarządzania, czyli strukturę organizacyjną nadzorującą procesy przepływu informacji i zadań pomiędzy poszczególnymi uczestnikami.

W tym etapie badań rozwojowych nad doбором metody modelowania skoncentrowano się na wykorzystaniu cech agentów programowych w zarządzaniu projektami informatycznymi. Warto jednak w tym miejscu nadmienić, że przeprowadzone analizy (rozdział I i II) pokazały, że problem zarządzania projektami jest dużo bardziej złożony i wymaga opracowania modelu nie tylko implementowanego jako system agentowy, ale pozwalającego na usprawnienia organizacyjne w projektach informatycznych, a także uwzględnienia procesów zarządczych i wytwórczych i dobrych praktyk zarządczych. Stąd też w opracowywaniu modelu usprawniającego zarządzanie projektami informatycznymi zdecydowano się wykorzystać połączenie podejścia procesowego oraz agentowego. Uznano, że ostateczny model (zwany adaptacyjnym agentowym modelem zarządzania projektami informatycznymi) łączyć powinien cechy podejścia agentowego w celu modelowania struktury projektowej oraz cechy podejścia procesowego związanego z definiowaniem parametrów wejściowych (jak np. dojrzałość klienta), przetwarzania (z uwzględnieniem relacji między agentami) oraz wyjść w modelu (w postaci adaptowanych dobrych praktyk). Połączenie tych dwóch podejść pozwoli na przygotowanie ram dla opracowania ostatecznej wersji modelu. Uznano ponadto, że model ze względu na sugestie co do łączenia poszczególnych elementów struktury projektowej oraz łączenia (dopasowywania) dobrych praktyk zarządzania projektami informatycznymi nabywa cech adaptacyjnych. Odejście od sugerowania metody na rzecz sugerowania menedżerom dobrych praktyk dało początek podejściu, w którym na podstawie znanej (określonej) przez menedżera struktury projektu tworzony jest zestaw wskazówek, jak postępować w projekcie. Następuje zatem adaptacja dobrych praktyk do realiów projektowych. Takie podejście nazwano **podjęciem adaptacyjnym**.

Łącząc zatem podejście agentowe, procesowe oraz adaptacyjne, przyjęto, że celem modelowania będzie środowisko nazwane ADAPTACYJNYM AGENTOWYM MODELEM ZARZĄDZANIA PROJEKTAMI INFORMATYCZNYMI opatrzone akronimem **a2M** (adaptacyjny agentowy Model).

3.4 Budowa prototypu funkcjonalnego modelu agentowego — submodel MAS_FUNK

Kolejny etap badań nad modelem **a2M** miał na celu doprowadzenie do precyzyjnego określenia zadań poszczególnych agentów modelu. Stan wiedzy na tym etapie pozwalał na koncentrowanie się na specyfikacji poszczególnych agentów — ich cech i zachowań (czyli stanów, które mogą przyjmować oraz akcji, które mogą wykonywać). Wydawało się, że takie podejście odzwierciedli strukturę projektów informatycznych oraz pozwoli na usprawnienie zarządzania projektami informatycznymi poprzez wsparcie podejmowania decyzji przez kierowników projektów informatycznych.

Omawiany etap badań, którego efektem był submodel MAS_FUNK[52], stanowił więc kamień milowy w dążeniu do opracowania modelu **a2M**. Skoncentrowano się bowiem na zdefiniowaniu funkcjonalności poszczególnych agentów, które są niezbędne dla wspomagania kierowników projektów. Zakładano na tym etapie badań, że usprawnienie zarządzania projektami informatycznymi jest możliwe poprzez system agentowy[40.45], którego model należy opracować, a następnie zaimplementować. Stąd też submodel MAS_FUNK stanowił architekturę zawierającą funkcjonalności poszczególnych agentów, które powinny zostać wykorzystane do generowania decyzji wspomagających kierowników projektów.

Po zakończeniu prac związanych z analizą dziedzinową oraz wyciągnięciu wniosków z opracowania koncepcji submodelu MAS, przystąpiono do opracowania rozszerzonej wersji submodelu MAS o specyfikację poszczególnych agentów. Rozszerzenie submodelu MAS o precyzyjną definicję zadań poszczególnych agentów oraz określenie zależności (relacji) między nimi doprowadziło do powstania submodelu MAS_FUNK. W tym etapie badania ukierunkowane były więc na możliwie dokładną i pełną specyfikację poszczególnych agentów. W tym celu podjęto badania pod kątem poszukiwania zależności pomiędzy poszczególnymi typami agentów. Analizowano również możliwości przystosowania agentów do realizacji procesu wspomagania decyzji kierowniczych w projektach informatycznych. Część badań skoncentrowano jednocześnie na możliwościach wzajemnej współpracy agentów i mechanizmach komunikacji między nimi. Stąd efektem prowadzonych badań było dokonanie pełnej definicji agentów. Definicja agentów oznaczała określenie wszystkich niezbędnych agentów dla realizacji procesu wspomagania decyzji kierowniczych oraz przyporządkowanie agentom zadań, które powinny wykonywać. Kolejnym krokiem było określenie relacji pomiędzy agentami, które zostały odwzorowane w postaci zależności hierarchicznych pomiędzy agentami. Takie podejście pozwoliło na opracowanie produktu III etapu badań — submodelu MAS_FUNK, czyli modelu systemu agentowego wspomagającego zarządzanie projektami informatycznymi posiadającego w pełni zdefiniowanych agentów. W opracowanym submodelu MAS_FUNK wyodrębniono główne typy agentów oraz przydzielono im przykładowe zestawy zadań, które powinny być wykonywane, a także podjęto pracę nad określeniem przepływu informacji między agentami. Ważnym wnioskiem z tego etapu było również określenie wszystkich funkcji systemu, które obejmują:

- pozyskiwanie informacji z otoczenia (przede wszystkim od menedżera projektu, ale także od klienta czy uczestników projektów informatycznych),
- przeszukiwanie zasobów,
- wnioskowanie w oparciu o udostępnione zasoby,
- zarządzanie poszczególnymi agentami systemu.

Po zdefiniowaniu tych głównych funkcjonalności możliwe stało się przejście do pracy nad sposobem przepływu informacji w przyjętej strukturze, czyli nad komunikatami, jakie agenty sobie przekazują. Dalsze prace wykazały, że określenie mechanizmów komunikacji między agentami programowymi przełożyło się na tworzenie relacji pomiędzy agentami zdefiniowanymi w ostatecznym adaptacyjnym agentowym modelu zarządzania projektami informatycznymi.

W submodelu MAS_FUNK sposób komunikowania się między agentami systemu okazał się najbardziej newralgicznym punktem. Podjęto próby zdefiniowania możliwych komunikatów oraz uznano, że za pomocą komunikatów agenty będą przekazywać sobie zlecenia do wykonania zadań. Zgodnie więc z teorią systemów agentowych, gdzie przepływ pracy agentów determinowany jest przez odpowiednie komunikaty, uznano, że w submodelu MAS_FUNK taki przepływ powinien zostać uwzględniony. Stąd też komunikaty miały za zadanie nieść ze sobą zlecenie podjęcia określonej akcji przez agenta. W trakcie opracowywania submodelu MAS_FUNK zauważono jednocześnie, że w podejściu agentowym można znaleźć wiele analogii do rzeczywistych projektów informatycznych, w których uczestnicy pracują w środowisku rozproszonym, a swoje działania rozpoczynają po otrzymaniu określonego komunikatu (zlecenia) od innego uczestnika (np. kierownika projektu).

W trakcie pracy nad sposobami komunikowania się agentów w submodelu MAS_FUNK, który umożliwiłby wsparcie podejmowania decyzji kierowników w projektach informatycznych, podjęto próbę zdefiniowania typowych komunikatów, jakie wysyłają agenty. Ponieważ przyjęto, że każdy agent ma przypisany zakres stanów (cech, uprawnień) oraz akcji (zachowań, zadań) (rys.3.4), które może zrealizować, dokonano definicji podstawowych komunikatów-zleceń, jakie mogą być wysyłane przez agenty (w zależności od ich umiejscowienia w układzie hierarchiczno-funkcyjnym).

Jednocześnie zauważono, że odebrany komunikat wpływa na zachowanie i stan agenta, co pozwala na definiowanie jego stanów w zależności od odebranego komunikatu.



Rys. 3.6 Przepływ zadań pomiędzy agentami

Źródło: opracowanie własne

Na rysunku 3.6 zaprezentowano przepływ zadań pomiędzy agentami w postaci komunikatów. Przykładowe komunikaty wysyłane między agentami przedstawiono w tabeli poniżej.

Tab. 3.1 Wpływ komunikatów na agenty w submodelu MAS_FUNK

Przykładowy komunikat	STAN agenta programowego	AKCJA agenta programowego	CECHA agenta projektowego	ZACHOWANIE agenta projektowego
Pozyskaj wymagania od klienta	Stan początkowy: czuwanie Stan po odebraniu komunikatu: działanie	Wysłanie wiadomości do klienta	Znajomość zarządzania wymaganiami (analityk biznesowy)	Spotyka się z klientem, pozyskuje wymagania i modeluje je np. w UML

Źródło: opracowanie własne

W tabeli 3.1 pokazano analogię pomiędzy przepływem komunikatów pomiędzy agentami programowymi oraz członkami zespołu projektowego traktowanych jako agenty w strukturze projektowej. Po uznaniu konieczności uwzględniania przepływu komunikatów na działanie agentów postanowiono również zdefiniować zakres działań poszczególnych agentów, który przedstawiono na rys. 3.7

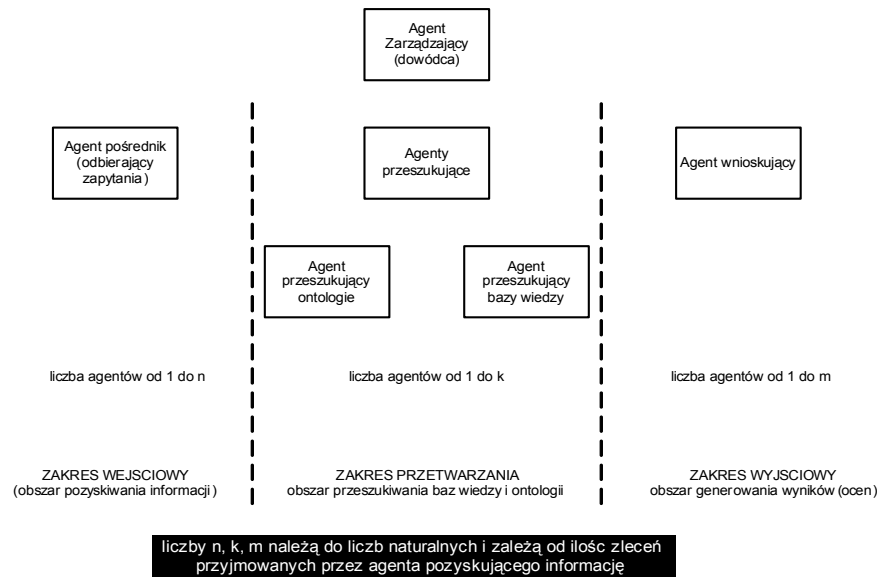


Rys. 3.7 Zakres działania agentów w submodelu MAS_FUNK

Źródło: [57]

Jak wynika z rys. 3.7, wyodrębniono zakres wejściowy, na który składać powinny się zmienne decyzyjne (czyli dojrzałość zespołu, dojrzałość klienta oraz entropia projektu). Wyodrębniono również zakres przetwarzania, (określono zależności pomiędzy agentami oraz podstawowe ich funkcjonalności) oraz zakres wyjściowy (decyzja wspomagająca kierownika projektu, np. jakiej dobrej praktyki użyć w zależności od wartości zmiennych decyzyjnych). W zależności od wyodrębnionych zakresów działania agentów postanowiono również określić główne typy agentów, przyporządkowując je do jednego z zakresów działania submodelu MAS_FUNK. Postanowiono

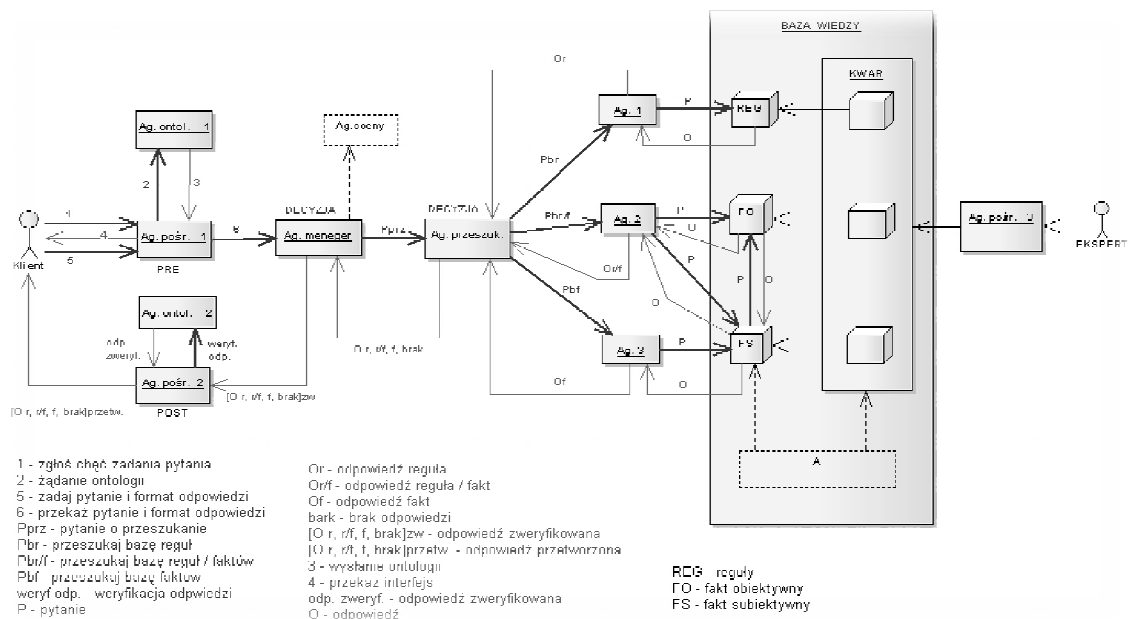
również uwzględnić układ hierarchiczny między agentami dla zaprezentowania procesów hierarchicznych procesów zarządczych. Na rys. 3.8 przedstawiono strukturę hierarchiczną agentów submodelu MAS_FUNK.



Rys. 3.8 Struktura hierarchiczna agentów submodelu MAS_FUNK

Źródło: [19]

Wstawienie agentów w strukturę hierarchiczną pozwoliło na stworzenie zależności pomiędzy poszczególnymi typami agentów. Ponadto wydzielono główne obszary zadaniowe (zachowania) dla agentów, takie jak pozyskiwanie informacji, przeszukiwanie zasobów wiedzy oraz generowanie decyzji wspomagających zarządzanie projektami. Dokładne relacje pomiędzy poszczególnymi agentami (odzwierciedlone w przepływach informacyjnych, przepływach komunikatów, przepływach zadań) zaprezentowane zostały wraz z układem hierarchicznym agentów i zgromadzone w postaci architektury submodelu MAS_FUNK.



Rys. 3.9. Prototyp funkcjonalny - submodel MAS_FUNK

Źródło: [49]

Po zdefiniowaniu funkcjonalności poszczególnych agentów możliwe stało się określenie relacji pomiędzy agentami (określenie zależności między nimi). Jest to istotne z punktu widzenia organizowania przepływu strumienia informacji pomiędzy agentami. Następnie dokonano sprawdzenia poprawności tej komunikacji. Uznano, że wstępna weryfikacja systemu powinna nastąpić w oparciu o dane z dowolnego systemu technicznego (niekoniecznie związanego z zarządzaniem projektem). Do weryfikacji wybrano system techniczny z uwagi na znaczną powtarzalność danych. Przyjęto dane systemu technicznego, w którym wszystkie analizowane parametry miały wartości liczbowe (wartości badanych parametrów w zaimplementowanym submodelu MAS_FUNK dotyczyły warunków atmosferycznych takich jak temperatura czy wilgotność, przy których określano ich wpływ na poziom zanieczyszczeń powietrza w aglomeracji Trójmiasta mierzony stężeniem takich substancji jak toluen, benzen itp.) [54,56].

Po przeprowadzeniu procesów wstępnej weryfikacji submodelu MAS_FUNK przeprowadzono weryfikację właściwą w oparciu o dane wejściowe dotyczące zarządzania projektami informatycznymi. Na podstawie tej weryfikacji stwierdzono, że agenty podejmowały działania niezależnie i autonomicznie w oparciu o przypisaną im inteligencję (inteligencję agentów stanowiły dedykowane reguły baz wiedzy). Uznano również, że tak zaprojektowany submodel oraz wstępna weryfikacja są wystarczające, aby przystąpić do wykorzystywania submodelu MAS_FUNK zgodnie z pierwotnie zakładanym przeznaczeniem, tj. do wsparcia podejmowania decyzji w projektach informatycznych. Uznano jednakże, że dla przyjętej architektury submodelu MAS_FUNK konieczne będzie dokładne określenie parametrów wejściowych. Zastosowanie języka naturalnego i tłumaczenie go za pomocą ontologii okazywało się nieefektywne. Takie podejście dało początek dalszym pracom badawczym związanym z zawężeniem pytań zadawanych kierownikom projektów przez agenta pośredniczącego (agent komunikujący się z menedżerem projektu w celu pozyskania informacji o projekcie), dzięki którym można było określić poziom dojrzałości klienta oraz poziom entropii projektowej. Uznano, że dopiero po takim zdefiniowaniu parametrów wejściowych submodel MAS_FUNK będzie w stanie przetworzyć dane wejściowe tak, aby uzyskać w efekcie konkretną decyzję, np. zalecaną metodę (lub dobrą praktykę) realizacji projektu zgodnie z zaprezentowanym ujęciem na rys. 3.8.

Podsumowując, głównym efektem (produktem) badań zrealizowanym na tym etapie budowy **a2M** było określenie zależności pomiędzy agentami i zdefiniowanie ich funkcjonalności. Zauważono również, że dla potrzeb właściwego wnioskowania (przetwarzania parametrów wejściowych) należy określić poziomy zagregowanych zmiennych wejściowych.

Po określeniu funkcjonalności agentów i relacji między nimi oraz zdefiniowaniu zmiennych decyzyjnych punkt ciężkości badań przeniesiono na proces wnioskowania, czyli generowania decyzji przez submodel MAS_FUNK. Wnioskowanie to (jak wynika z rysunku 3.9) odbywa się w zakresie przetwarzania i jest realizowane przez agenty zestawione w układzie hierarchicznym. W zależności od tego, jakie wartości przyjmują parametry wejściowe (dotyczące złożoności projektu, dojrzałości organizacji oraz dojrzałości klienta), uruchamiany jest proces generowania ocen/wniosków przez agenty submodelu MAS_FUNK. W wyniku wygenerowanych wyników użytkownik lub menedżer projektu otrzymuje wskazanie, np. jakiej metody realizacji przedsięwzięć informatycznych powinno się użyć pod potrzeby danego projektu lub jakie dobre praktyki zastosować. Podczas sesji z submodelem MAS_FUNK agent pośredniczący pozyskuje informacje wprowadzone przez użytkownika w polu zapytania. Informacje te zostają przekazane agentowi menedżerowi. W oparciu o te informacje agent menedżer podejmuje decyzje co do wykorzystania posiadanych przez submodel MAS_FUNK zasobów wiedzy. Taki przepływ zadań wynika z przyjętej architektury systemu, w którym agenty zgromadzone są według funkcji i hierarchii. W oparciu o inne źródła wiedzy, takie jak bazy wiedzy, agent wnioskujący dokonuje analizy przekazanego zapytania. Analiza może przebiegać dwutorowo. Może być to proste przeszukanie (podczas którego agent menedżer uruchamia agenta przeszukującego oraz przekazuje zlecenie dotyczące przeszukania zasobów wiedzy

znajdujących się w systemie, np. bazy faktów). W drugim przypadku, gdy zapytanie menedżera jest bardziej skomplikowane, uruchamiany jest agent wnioskujący korzystający z elementów wspomagających wnioskowanie (system ekspertowy, sieć neuronowa). Po tej akcji następuje przekazanie wyników do agenta menedżera. Pozyskane z obliczeń wyniki przekazywane są klientowi za pośrednictwem agenta pośredniczącego.

Określenie procesu przepływu informacji pomiędzy agentami w trakcie generowania decyzji wspierających menedżerów projektów dotyczące doboru metody zarządzania projektem (dobrych praktyk) zamknęło etap prac nad submodelem MAS_FUNK. Po zakończeniu tego etapu przystąpiono do prac implementacyjnych modelu w środowisku webowym.

3.5 Implementacja prototypu funkcjonalnego — submodel MAS_WEB

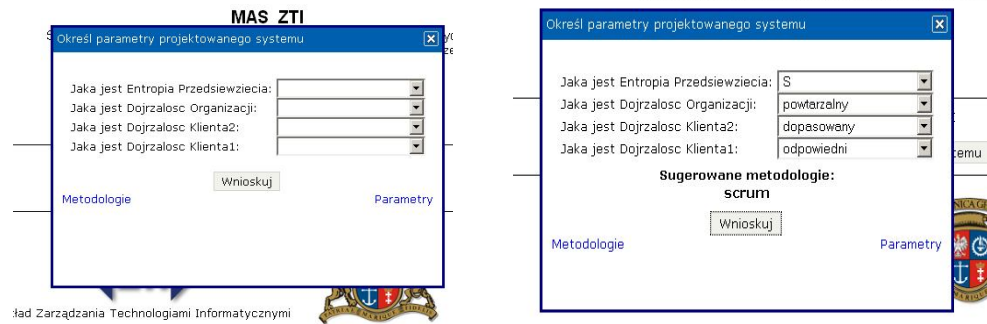
Po przeprowadzonych badaniach nad architekturą submodelu MAS_FUNK, podczas których określono parametry wejściowe, wyjściowe oraz zakres przetwarzania dla modelu, zdecydowano się na implementację tego submodelu, aby stał się dostępny dla menedżerów projektów z poziomu przeglądarek internetowych. W wyniku tej implementacji powstał prototyp decyzyjnego systemu wsparcia opartego na architekturze submodelu MAS_FUNK[47,53, 55]. Produkt tego etapu określono mianem submodelu MAS_WEB. W wytwarzaniu submodelu MAS_WEB zastosowano podejście lekkie z racji dojrzałości zespołu realizującego system, jak również krótkiego czasu trwania projektu (3 miesiące). Prace realizowano w 5 sprintach, a efektem prowadzonych prac była gotowa wersja systemu, którego mechanizm działania zgodny był z uprzednio omówionym submodelem MAS_FUNK.



Rys. 3.10 Kontakt z użytkownikiem w submodelu MAS_WEB

Źródło: opracowanie własne na podstawie zbudowanego interfejsu prototypu MAS_WEB

Na powyższym rysunku przedstawiono widoki z zaimplementowanego systemu/submodelu MAS_FUNK wspomagającego decyzje menedżerów dotyczące doboru metod zarządzania projektami w zależności od wartości zmiennych decyzyjnych. Podczas sesji z systemem agent pośredniczący prezentuje użytkownikowi (menedżerowi) zestaw zmiennych decyzyjnych nowego trójkąta ograniczeń (w postaci list rozwijalnych). Użytkownik określa wartości zmiennych: dojrzałość zespołu, dojrzałość klienta oraz poziom entropii projektowej, wybierając jedną z kilku proponowanych wartości (dla potrzeb prototypu przyjęto skale trój- lub pięciostopniowe). Po ich określeniu generowany jest był wynik (decyzja) — metoda zarządzania projektem informatycznym.



Rys. 3.11 Okno zapytań i wygenerowana sugestia dotycząca metody w submodelu MAS_WEB

Źródło: opracowanie własne na podstawie zbudowanego interfejsu prototypu MAS_WEB

Zaimplementowany prototyp MAS_WEB pozwalał na wnioskowanie (generowanie decyzji) na podstawie podanych przez menedżerów parametrów wejściowych oraz w oparciu o wiedzę ekspercką zgromadzoną w bazie wiedzy. Wiedza opierała się na regułach tworzonych przez grono ekspertów.

Poniżej przedstawiono przykład takiej reguły:

Jeżeli dojrzałość zespołu jest niska, dojrzałość klienta mała i duża entropia projektu, to zalecona metoda prowadzenia projektu to RUP. (1)

Dodatkowym doświadczeniem płynącym z implementacji submodelu MAS_FUNK było sprawdzenie mechanizmu działania agentów w odniesieniu do uprzednio określonych ich funkcjonalności podczas definiowania architektury submodelu MAS_FUNK. Wcześniej opracowane procedury przeszukiwania zasobów i generowania decyzji zostały zweryfikowane podczas wykorzystania prototypu submodelu MAS_WEB. Wnioski z przeprowadzonych testów działania prototypu przedstawiały się następująco:

- Agenty systemu zachowują się poprawnie, reagują na bodźce zgodnie z oczekiwaniami, a sekwencyjny przepływ komunikatów realizowany jest zgodnie z założeniami.
- Każdy kolejny komunikat implikuje działanie właściwego dla danego typu komunikatu agenta.
- Agenty podejmują akcję wyłącznie wtedy, gdy pojawi się sygnał od klienta (zapytanie).
- W procesach testowania zostały zastosowane wszystkie uprzednio zdefiniowane funkcjonalności agentów.
- Wykorzystana architektura systemu agentowego może być wykorzystana jako moduł, np. w aplikacjach do zarządzania projektami (np. Rational Team Concert).

Proces testowania prototypu pozwolił na dalsze usprawnienia prototypu submodelu MAS_WEB. Zaobserwowano, że procesom weryfikacji muszą zostać poddane parametry wejściowe i sposób ich określenia przez menedżerów. Nie każdy bowiem menedżer był w stanie określić wprost poziom dojrzałości swojego zespołu, a tym bardziej dojrzałości klienta. Dało to podstawę do wyciągnięcia kilku wniosków dotyczące dalszych prac w celu usprawnienia modelu. Uznano, że niezbędne będzie przetwarzanie wstępne (ang. *preprocessing*), który polegać będzie na prezentowaniu menedżerom zestawu pomocniczych pytań. Na podstawie odpowiedzi na te pytania określone zostaną poziomy zmiennych decyzyjnych (dojrzałości klienta, dojrzałości zespołu i entropii projektowej).

Na uwagę zasługuje również fakt, iż zauważono, że architektura agentowa zaprojektowana w submodelu MAS_FUNK i przetestowana w oparciu o prototyp submodelu MAS_WEB stanowi elastyczną architekturę przeznaczoną do integracji z aplikacjami wspomagającymi zarządzanie projektem. Docelowo taki prototyp jak submodel MAS_WEB może zostać poddany procesom implementacji np. w systemie Rational Team Concert służącym do kierowania pracami projektowymi

i wspomagać menedżerów projektów w doborze metod zarządzania projektami w oparciu o zmierzone wartości zmiennych decyzyjnych.

Model wsparcia zarządzania projektami wraz z mechanizmem wnioskowania zaimplementowanym do postaci aplikacji webowej (prezentowany jako prototyp MAS_WEB) został zaprezentowany grupie kierowników projektów. Prezentacja nastąpiła na jednej z konferencji. Okazało się jednak, że dla wielu menedżerów mechanizmy wnioskowania widoczne w prototypie submodelu MAS_WEB nie są tak istotne i stanowią raczej rzecz wtórną. O wiele więcej problemu sprawia odpowiedź na podstawowe pytania zadawane przez system dotyczące poziomów dojrzałości zespołu dostawcy czy klienta. Potwierdziły się zatem wcześniejsze spostrzeżenia, że niezbędne jest uzupełnienie modelu o pytania pomocnicze prezentowane w procesach przetwarzania wstępnego.

Dla wielu menedżerów problemem okazało się określenie dojrzałości swoich organizacji czy zespołów. Był to więc wyraźny sygnał że w dalszych pracach nad **a2M** należy położyć nacisk nie tyle na poprawność techniczną modelu, co na uwzględnienie w modelu zarządzania projektem informatycznym elementów organizacyjnych związanych ściśle z dyscypliną zarządzania.

Menedżerowie wskazali na brak pomiarów poziomów dojrzałości oraz na brak sposobu określania otoczenia projektowego (realiów projektowych). Wskazali dodatkowo, że jest to problem tak złożony, że wielu z nich ma problemy już na poziomie „uchwycenia” wszystkich istotnych elementów projektu i ich dopasowywania [79].

Kolejne rozmowy z menedżerami pokazały również, jak skomplikowanym problemem decyzyjnym jest dobór metod prowadzenia projektów. Okazało się, że w wielu przypadkach metodę prowadzenia projektu dobiera się w sposób wyłącznie intuicyjny na podstawie własnych przekonań, a w niektórych firmach w sposób nawet instytucjonalny — po prostu taka metoda jest stosowana w korporacji. Zwrócono również uwagę na istnienie „metody partyzanckiej” [62], gdzie projekty realizowane są właściwie bez żadnych zasad czy procedur tak skrupulatnie definiowanych w dobrych praktykach metod zarządzania projektami [16].

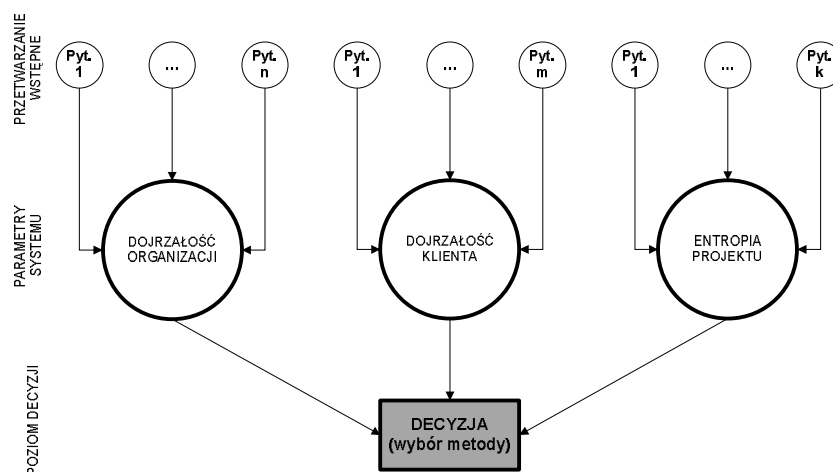
Kolejny etap budowy **a2M** zamykany budową prototypu submodelu MAS_WEB wraz z opiniami menedżerów pozwolił na zebranie kilku ważnych wniosków:

1. Po pierwsze nie należy stosować metody prowadzenia projektów wprost do organizacji. Decyzja dotycząca sposobu prowadzenia projektu musi uwzględniać analizę parametrów organizacji odniesionych do trzech głównych zmiennych zagregowanych, czyli dojrzałości zespołów klienta, dostawcy i entropii projektu.
2. Po drugie prototyp submodelu MAS_WEB jest akceptowalny tylko wtedy, gdy menedżerowie są bardzo doświadczeni i znają bardzo dobrze specyfikę zarządzania projektami. Menedżerowie mniej doświadczeni w pierwszej kolejności potrzebują usprawnień organizacyjnych, a dopiero później wykorzystują systemy wsparcia decyzyjnego (jak prototyp submodelu MAS_WEB). Od tego momentu uznano, że **a2M** musi łączyć w sobie zarówno aspekt informatyczny, jak również aspekt organizacyjny.

Rozważania po weryfikacji prototypu submodelu MAS_WEB stały się punktem wyjścia do poszukiwań rozwiązania bardziej ogólnego. Jeżeli nie każdy menedżer umie wskazać, jakie parametry decyzyjne występują w jego projekcie, jeśli nie potrafi ich zmierzyć albo niekiedy nawet nie potrafi wyodrębnić wszystkich elementów projektu, to znaczy, że poszukiwany model usprawniający zarządzanie projektami powinien mieć charakter **generyczny**. Taki model stanowi wtedy uogólnienie wszystkich składowych projektu, zawierając w sobie cechy zarówno rozwiązania technicznego — pozwalającego analizować i prognozować stany zespołów i projektu na podstawie danych, jak i rozwiązania organizacyjnego — stanowiącego zestaw procedur postępowania członków zespołu projektowego dostawcy.

3.6 Zastosowanie procesów przetwarzania wstępnego na potrzeby submodelu MAS_WEB

Na podstawie przeprowadzonych badań nad prototypem submodelu MAS_WEB uznano za konieczne opracowanie mechanizmu mierzenia zmiennych decyzyjnych (dojrzałości organizacji, klienta i entropii projektu) w projektach informatycznych. Zaobserwowano bowiem, że zmienne te stanowią zagregowane wartości znacznej liczby podstawowych parametrów/zmiennych. Okazało się także, iż menedżerom jest stosunkowo trudno odpowiedzieć na pytanie, na ile ich zespół jest dojrzały. Trudno również ocenić, jaki poziom dojrzałości reprezentuje klient. Dlatego też podjęto decyzję o zastosowaniu w a2M procesów preprocessingu zagregowanych zmiennych decyzyjnych. Poniżej, na rys. 3.12, przedstawiono submodel przetwarzania wstępnego prezentujący sposób pozyskiwania danych służących podejmowaniu decyzji kierowniczych.



Rys. 3.12 Submodel przetwarzania wstępnego dla generowania decyzji

Źródło: [57]

Dla wsparcia tych procesów przygotowano również zestaw pytań kontrolnych zadawanych menedżerom w celu ustalenia poziomów zmiennych decyzyjnych. Poniżej przedstawiono przygotowaną wstępną listę pytań[57]:

Pytania dotyczące dojrzałości organizacji:

- Czy kierownik zna i wykorzystuje model CMMI?
- Czy jest w stanie ocenić poziom dojrzałości własnej organizacji?
- Czy zespół wytwórczy realizował w przeszłości projekty? (ile?)

Pytania dotyczące dojrzałości klienta:

- Czy realizowano już przedsięwzięcia informatyczne?
- Czy projekty z udziałem klienta zakończyły się powodzeniem?
- Czy klient zna specyfikę projektów informatycznych?

Pytania dotyczące entropii projektu:

- Na ile złożone są funkcjonalności podane przez klienta?
- Czy lista funkcjonalności została zamknięta?
- Czy klient precyzyjnie zdefiniował oczekiwania odnośnie wytwarzanego systemu informatycznego?

Po spotkaniach z kierownikami projektów i ekspertami od zarządzania projektami przedstawiona lista pytań została następnie zmodyfikowana do postaci prezentowanej w załączniku. W trakcie sesji

z ekspertami zwracali oni uwagę na brak w ocenach projektu, klienta i dostawcy elementów „miękkich” — związanych z problemami organizacyjnymi projektu [5,21].

Okazało się także, że o ile rozwiązanie oparte o agenty programowe może być wykorzystywane w obszarach prognozowania stanów organizacji i projektu, o tyle jest zbyt proceduralne. Tak złożone środowisko trudno się modeluje z wykorzystaniem wyłącznie systemów wnioskujących (w oparciu o dane pozyskiwane od kierowników projektów i wprowadzane do systemu). Uznano więc, że model powinien zostać poddany zmianom i przejść od opartego na inteligentnych jednostkach wnioskujących submodelu MAS_WEB do modelu ogólnego całego przedsięwzięcia, uwzględniając poszczególne jego elementy (nazywane w niniejszej rozprawie „składnikami” projektu) traktowane jako poszczególne agenty całego systemu.

Drugim ważnym wnioskiem płynącym od ekspertów zarządzania projektami (w trakcie sesji z systemem) była informacja, że metody zarządzania projektami nie zawsze stosowane są w całości. Autor niniejszej rozprawy korzystał z takiego rozwiązania i przekonał się również o zaletach wykorzystywania wybranych dobrych praktyk, stosując w kilku projektach zasady pracy parami (dobre praktyki z metody XP Programming) oraz wysokopoziomowej prezentacji projektu (dobrej praktyki zgodnej z zasadami Rational Unified Process) [9,63].

Te obserwacje skłoniły autora do kolejnych zmian modelu. Zaczęto bowiem zastanawiać się, czy sugerowanie kierownikom dokładnie jakiej metody użyć jest rozwiązaniem wiążącym (wnioski wynikające z analizy rozdziału I wskazują na dobre praktyki). Jakie działania należy podejmować, jeżeli menedżer nie zna tej metody? Co należy zrobić, gdy menedżer stosował sugerowaną w odpowiedzi od systemu wnioskującego metodę i nie jest do niej przekonany?

Te refleksje stanowiły punkt zwrotny w rozważaniach nad ogólną koncepcją modelu. Zauważono bowiem, że jeżeli podejście agentowe wiąże się z bardzo precyzyjnym wyodrębnieniem podstawowych składników projektu, to należy również dobre praktyki metod zarządzania projektami potraktować jako agentów, które powinny być dopasowane do ogólnej struktury projektu. Jednak pamiętając, że menedżerowie nie muszą znać wszystkich metod zarządzania projektami, uznano, że rozsądnym rozwiązaniem będzie potraktowanie (zgodnie z definicją) metod zarządzania projektami jako zestawu dobrych praktyk i dobierać na potrzeby agentów nie metody, a dobre praktyki. Każda dobra praktyka może mieć swoją reprezentację w postaci określonego agenta. Dzięki temu w ostatecznej wersji modelu menedżer otrzyma nie tyle zalecenie w postaci: „korzystaj z metody X”, a raczej zestaw wytycznych postaci: „dla Twojego projektu zalecane jest realizowanie prac projektowych według dobrej praktyki X1, X2, X3”. W ten sposób powstanie procedura postępowania dedykowana konkretnemu projektowi przy zagregowanych wartościach zmiennych decyzyjnych (opisanych w rozdziale II).

Zaprezentowane etapy budowy modelu **a2M** doprowadziły do poszerzenia wiedzy odnośnie jego ostatecznej wersji. Większość dotychczasowych badań skoncentrowana była na implementacyjnym wymiarze modelu, gdyż na początku procesu wytwarzania wydawało się, że rozsądnym rozwiązaniem będzie wykorzystywanie przez kierowników tylko systemu MAS_WEB. Stwierdzono jednak, że zastosowanie modelu wspierającego tylko decyzje menedżerów projektu w oparciu o zagregowane zmienne decyzyjne jest niewystarczające. Konieczne okazało się uwzględnienie szeregu procesów zarządczych, które należy również uwzględnić w modelu. Stąd też uznano, że ostateczna wersja **a2M** powinna posiadać szerszy wymiar, wprowadzający także usprawnienia organizacyjne do zarządzania projektami informatycznymi.

3.7 Analiza możliwości adaptacji i weryfikacja modelu

Kolejny rozdział został poświęcony prezentacji możliwości adaptacji **a2M**. Badania prowadzące do opracowania tego modelu prowadzone były na przestrzeni pięciu lat zgodnie z omówioną wcześniej metodą badawczą [8]. W tym okresie koncepcja budowy **a2M** ewoluowała zgodnie z zaprezentowanym uprzednio opisem prowadzonych badań. Każdy etap prowadzonych badań

stanowił kolejne (iteracyjne) przybliżenie i był zamykany określonymi artefaktami (produktami) [8]. Do artefaktów tych należy zaliczyć także kolejno wnioski/raporty badawcze stanowiące punkt wyjścia do dalszych udoskonaleń modelu. Warto również dodać, że w trakcie prowadzenia prac badawczych nad kolejnymi artefaktami wypracowano jednocześnie dobre praktyki związane z zarządzaniem projektami informatycznymi (budowa poszczególnych submodeli stanowiła również rodzaj projektu informatycznego). Po zaprezentowaniu etapów budowy modelu **a2M** postanowiono przygotować (zebrać) wszystkie wnioski z wcześniej przeprowadzonych badań, aby przygotować założenia do ostatecznej wersji modelu.

Wcześniejsze analizy pozwoliły na wyodrębnienie zagregowanych zmiennych, które powinny zostać poddane analizie i które bezpośrednio oddziałują na przebieg projektu. Zauważono jednak, że dla poprawnej realizacji projektu należy również wyodrębnić inne elementy struktury projektu, w tym dodatkowe osoby (np. ekspertów), technologie i inne źródła wiedzy, do których menedżerowie mogą się odnosić w trakcie trwania projektu. Uznano również, że tak jak w każdym projekcie powinien istnieć centralny ośrodek decyzyjny, w skład którego, oprócz menedżera projektu, powinien wchodzić szereg podległych mu analityków, wspierających go w rozpoznaniu wszystkich składników projektu oraz w pomiarze niektórych parametrów (np. poziomów dojrzałości). Stąd uznano, że tak jak to miało miejsce w submodelach opracowanych uprzednio, tak w wersji ostatecznej **a2M** w modelu powinny wystąpić niezależne agenty ułożone w strukturze hierarchicznej (agent menedżer, agent koordynator itp). Uznano także, że zaproponowane podejście agentowe powinno zostać przełożone na płaszczyznę elementów struktury projektu. Zauważono bowiem analogię pomiędzy zachowaniami agentów w systemach agentowych a uczestnikami projektów podczas realizacji prac projektowych.

Z powyższych rozważań wynikają trzy wnioski:

1. Po pierwsze **a2M** może być traktowany jako poddane implementacji środowisko agentowe. Na podstawie **a2M** możliwe będzie zbudowanie kompletnego systemu informatycznego wspomagającego zarządzanie projektami, który będzie w stanie zasymulować konkretne środowisko projektowe.
2. Po drugie **a2M** będzie generował usprawnienia organizacyjne sugerujące kierownikom, aby wszystkie elementy struktury definiować jako niezależne agenty. Agenty te powinny współpracować ze sobą, a następnie mierzyć dojrzałość klienta i własnego zespołu. Powinny także wiązać wszystkie elementy w sprawnie funkcjonującą strukturę projektu zapewniającą właściwe zarządzanie projektem.
3. Po trzecie silne powiązanie pomiędzy **a2M** z zespołem dostawcy umożliwi uruchomienie procedur adaptacyjnych dla modelu **a2M**.

Stąd też przed stworzeniem ostatecznej wersji **a2M** opracowano jego założenia dotyczące możliwości adaptacji i weryfikacji modelu. Przygotowanie takich założeń staje się niezbędne przed rozpoczęciem prac nad budową modelu **a2M**.

Założenie 1 — możliwości skalowalności modelu i uwzględnianie elementów rozproszonych

Pierwszym założeniem dla budowy modelu jest uwzględnienie możliwości jego skalowalności. Do badań i weryfikacji będą wykorzystane dobre praktyki z czterech podstawowych metod zarządzania projektami. Uwzględnione zostaną tylko te składniki, które udało się zaobserwować na przestrzeni 5 lat badań. Rodzi się jednak pytanie, co należy zrobić, kiedy liczba metod zostanie powiększona (tym samym zwiększy się zbiór dobrych praktyk). Rodzi się także pytanie, co należy zrobić, jeżeli do zespołu zostanie dołączony nowy członek (względnie odejdzie). Jak zmieni się struktura projektu (i czy projekt będzie realizowany poprawnie), gdy będziemy dodawać nowe technologie do projektu?

Wydaje się, że stosując podejście agentowe, model może być rozbudowywany, a każdy dodatkowy element (nowy członek zespołu, nowa metoda zarządzania, nowa dobra praktyka) może być traktowany jako atrybut charakterystyczny dla agenta lub jako agent.

Założenie 2 – możliwości adaptacji modelu a2M

Możliwości adaptacji **a2M** oznaczają, że opracowywany model powinien zostać wyposażony w mechanizmy łączenia cech poszczególnych elementów struktury projektowej (agentów) oraz poszczególnych elementów struktury projektu (członków zespołu projektowego), ale również zawierać mechanizm odpowiedniego ich łączenia. Planuje się w tym celu tworzenie agentów samouczących się. Planuje się także, aby model posiadał w tym celu odpowiednie algorytmy — na przykład specyficzne funkcje dopasowania. Przykład takiej funkcji **A_MATCH** zaprezentowano w rozdziale II.

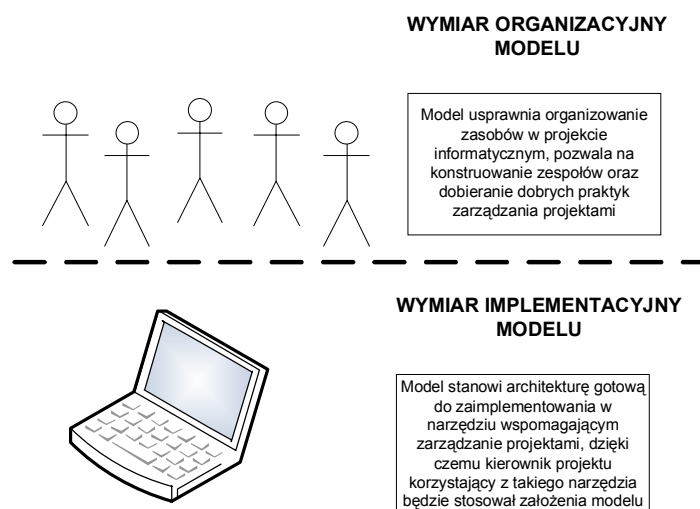
Założenie 3 — Każdy element struktury projektu powinien być traktowany jako agent

Kolejnym założeniem do opracowania modelu **a2M** jest konieczność traktowania każdego elementu struktury projektowej jako niezależnego agenta. Wszystkie uprzednio wyodrębnione elementy struktury projektowej jak członkowie zespołu projektowego, klient, eksperci, narzędzia wspomagające realizację projektu, dobre praktyki zarządzania projektami, modele ocenowe powinny zostać potraktowane jako agenty. Tylko wtedy możliwe będzie ich integrowanie w ramach struktury projektowej oraz możliwe będzie dokonanie procesów adaptacji.

Przyjmuje się także, że każdy element struktury projektowej powinien być reprezentowany przez atrybuty typowe dla agenta (cechy i zachowania) oraz uruchamiany jest w określonej sytuacji w celu wykonania powierzonego mu zlecenia. Takie założenie wydaje się być zgodne z typowymi działaniami w projektach informatycznych, gdzie kierownik przekazuje określone polecenia pozostałym agentom (uczestnikom), którzy realizują powierzone im zlecenie.

Założenie 4 — dualizm zastosowania

W trakcie prac koncepcyjnych nad ostateczną wersją modelu przyjmuje się także, że aby sprostać oczekiwaniom menedżerów projektów, model **a2M** należy rozpatrywać w dwóch wymiarach – organizacyjnym oraz implementacyjnym.



Rys. 3.13 Dualny charakter adaptacyjnego agentowego modelu zarządzania projektami informatycznymi

Źródło: opracowanie własne

Jak wynika z prezentowanego rysunku 3.13, pierwszy wymiar modelu ma na celu stworzenie kompletnego zarządzanego i dopasowanego do wszystkich zmiennych decyzyjnych środowiska projektowego, w ramach którego podejmowane są decyzje zarządcze.

Z kolei drugi wymiar to wymiar implementacyjny. Model powinien umożliwić implementację wszystkich elementów wyodrębnionych w modelu za pomocą dowolnego środowiska programistycznego. Takie podejście pozwoli na otrzymanie kompletnego systemu wspomagania decyzji kierowniczych w projektach oraz zapewni możliwość symulacji zachowania się poszczególnych składników projektu.

3.8 Podsumowanie

Celem niniejszego rozdziału było przygotowanie założeń do budowy adaptacyjnego agentowego modelu zarządzania projektami (**a2M**). Aby ten cel zrealizować, przyjęto określoną metodę prowadzenia badań, w której począwszy od obserwacji (rozumianych jako wywiady z kierownikami projektów, czy też uczestnictwo w projektach informatycznych) dokonuje się analizy dziedzinowej, a następnie przystępuje do modelowania oraz testowania zaimplementowanego modelu.

Podczas przeprowadzania analizy dziedzinowej wykazano istotność zmiennych decyzyjnych nowego trójkąta ograniczeń z punktu widzenia kierownika oraz wpływ tych zmiennych na realizację zadań w projekcie. W dalszej części zaprezentowano etapy budowy modelu usprawniającego zarządzanie projektami informatycznymi poprzez prezentację kolejnych submodeli (będących odpowiednikami prototypów). W rozdziale zawarto również wnioski płynące z opracowania poszczególnych submodeli, które okazały się niezbędne z punktu widzenia dalszych prac badawczych prowadzących do opracowania **a2M**. Działania te prowadzone były zgodnie z przyjętą metodą badawczą, również opisaną w tym rozdziale. Zebrane obserwacje oraz doświadczenia płynące z implementacji prototypów dały podstawy do wybrania metody modelowania niezbędnej dla opracowania modelu usprawniającego zarządzanie projektami informatycznymi. Na podstawie analizy możliwych podejść do budowy modelu zdecydowano się na powiązanie podejścia procesowego oraz podejścia agentowego w celu uzyskania lepszego dopasowania modelu do rzeczywistych struktur w projektach informatycznych. W rozdziale również wskazano na konieczność transformacji elementów struktur projektowych do postaci agentów. Dzięki temu możliwe będzie łączenie ich w zależności od wartości zmiennych decyzyjnych nowego trójkąta ograniczeń. Takie podejście uznano za niezbędne z punktu widzenia późniejszej adaptacji dobrych praktyk do projektu.

Rozdział zakończono prezentacją podstawowych założeń do budowy **a2M** odnośnie doboru dobrych praktyk zarządzania projektem informatycznym w zależności od wartości zmiennych nowego trójkąta ograniczeń. Wśród tych założeń przyjęto przede wszystkim możliwość skalowalności modelu (rozbudowania go o nowe agenty), możliwość adaptacji, czyli dobierania dobrych praktyk w zależności od nowego trójkąta ograniczeń. Zwrócono również uwagę na konieczność reprezentowania poszczególnych elementów struktury projektowej w postaci agentów. Przyjęto również jako założenie zasadę dualizmu modelu, czyli możliwości stosowania go zarówno do usprawnień organizacyjnych, jak i do implementowania w dowolnym narzędziu wspomagającym kierownika projektu.

Rozdział IV

Adaptacyjny agentowy model zarządzania projektem informatycznym

Efektom ewolucji submodeli (MAS, MAS_FUNK i MAS_WEB) przedstawionych w poprzednim rozdziale było przygotowanie założeń do budowy modelu **a2M**. Uznano także, po analizie możliwości adaptacji i weryfikacji, że model **a2M** umożliwił będzie kierownikom projektów dobór dobrych praktyk zarządzania projektami informatycznymi. Zdecydowano również, że model **a2M** powinien zostać zbudowany w oparciu o hybrydowe, procesowo-agentowe podejście.

Zdiagnozowano także analogię pomiędzy agentami programowymi a członkami zespołu dostawcy. Tę analogię postanowiono wykorzystać podczas opracowywania modelu usprawniającego pracę kierowników projektów. Uznano bowiem, że problem zarządzania projektem jest na tyle złożony, że opracowując model usprawniający zarządzanie projektami należy wyodrębnić, oprócz wymiaru implementacyjnego, również wymiar organizacyjny. Stąd też przy opracowaniu **a2M** konieczne jest uwzględnienie obydwu wymiarów, a nie, jak zakładano wcześniej, tylko wymiaru implementacyjnego.

Celem niniejszego rozdziału jest zaprezentowanie modelu ogólnego i szczegółowego **a2M** zbudowanego w oparciu o metodykę badawczą zaprezentowaną w rozdziale III. Model ogólny prezentuje podstawowe typy agentów oraz zależności między nimi. W modelu szczegółowym wyróżniono jego poziomy, w ramach których kierownicy projektów powinni organizować poszczególne elementy struktury projektowej i odpowiednio nimi zarządzać. W ramach każdego z poziomów wyróżniono także warstwy, które zostały wyodrębnione w celu uszczegółowienia zadań, jakie powinni wykonać kierownicy projektów informatycznych.

Podczas prezentacji modelu **a2M** starano się również podkreślić dualizm modelu, wskazując na konieczność wprowadzania usprawnień organizacyjnych, jak też implementacyjnych. Dlatego też na wstępie przedstawiono odpowiednią specyfikację **a2M**, która może zostać wykorzystana do budowy dowolnych środowisk projektowych.

4.1 Specyfikacja agentowego modelu zarządzania projektami

Na podstawie przyjętych założeń w rozdziale III postanowiono przygotować specyfikację agentowego modelu zarządzania projektami poprzez wyodrębnienie wszystkich niezbędnych z punktu widzenia zarządzania projektem agentów. Wykorzystano w tym celu doświadczenia z opracowania poszczególnych prototypów (submodeli) oraz doświadczenia wyniesione z obserwacji rzeczywistych projektów. Stwierdzono przy tym także, że wyodrębnienie wszystkich niezbędnych agentów jest kluczowe dla tworzenia struktury projektu. Dzięki takiemu wyodrębnieniu kierownik projektu może sprawdzić, czy takie agenty występują w realizowanym przez niego projekcie. W ten sposób można przyjąć, że podczas opracowywania modelu wypracowane zostają również dobre praktyki zarządzania projektami.

Zgodnie z przyjętymi założeniami odnośnie dualnego charakteru adaptacyjnego agentowego modelu zarządzania projektami informatycznymi uznano za konieczne podczas omawiania poszczególnych agentów zaznaczyć, jak działają takie agenty w wymiarze organizacyjnym modelu oraz jak powinny zachować się w wymiarze implementacyjnym. Stąd też omówione poniżej agenty w wymiarze organizacyjnym należy postrzegać jako członków zespołu realizującego projekt,

w wymiarze implementacyjnym agenty rozumiane są jako jednostki programowe funkcjonujące w oparciu o komunikaty (zlecenia) i podejmujące określone akcje.

Na podstawie doświadczeń zdobytych podczas budowy submodeli prezentowanych w rozdziale III wyodrębniono trzy podstawowe kategorie agentów dla potrzeb modelu **a2M**:

1. Pierwszą kategorię stanowią agenty funkcyjne. Ich zadaniem jest pozyskiwanie niezbędnych informacji pozwalających na tworzenie nowego trójkąta ograniczeń. Stąd też sugeruje się obecność takich agentów funkcjonalnych jak AGENT ENTROPII, AGENT KLIENTA, AGENT ZESPOŁU. Do kategorii agentów funkcjonalnych należy również zaliczyć wszystkie te agenty, które mają realizować zadania w projekcie informatycznym, czyli AGENT ZADANIOWY.
2. Drugą kategorię agentów stanowią agenty zarządzające, czyli te, które odpowiadają za strumień informacji w realizowanym projekcie. Wysyłają komunikaty (zlecenia) do agentów funkcjonalnych oraz pozyskują od nich informacje. Agenty zarządzające są więc odpowiednikiem szczebla menedżerskiego w hierarchicznej strukturze organizacyjnej. Ich zadaniem jest także sprawowanie funkcji kontrolnych, organizacyjnych czy planistycznych. Do agentów zarządzających zaliczono następujące: AGENT MENEDŻER, AGENT KOORDYNATOR, AGENT WNIOSKUJĄCY.
3. Trzecią kategorią agentów niezbędnych z punktu widzenia tworzenia struktury projektu są agenty wiedzy.

4.1.1 Specyfikacja agentów funkcjonalnych

Zgodnie z przedstawionymi wcześniej założeniami agenty funkcjonalne odpowiedzialne są za realizację podstawowych zadań w projekcie oraz zobowiązane są do dostarczania informacji agentom wyższego szczebla, czyli agentom zarządzającym. Wyróżniono więc 4 agenty funkcjonalne, których obecność w strukturze projektu jest niezbędna z punktu widzenia agentowego modelu zarządzania projektami informatycznymi. Do agentów funkcjonalnych zaliczono:

- **AGENTA DOJARZAŁOŚCI KLIENTA** — jest to agent odpowiedzialny za kontakt z klientem, pozyskiwanie informacji o jego wymaganiach, określenie jego preferencji, profilu biznesowego oraz poziomu znajomości specyfiki projektów informatycznych. W celu określenia dojrzałości klienta agent dojrzałości posługuje się odpowiednim zestawem pytań przygotowanych w procesach preprocessingu. Zestaw pytań jest uprzednio przygotowany przez menedżera (przykładowy zestaw pytań przedstawiono we wcześniejszej części rozprawy, a pełny w załączniku do pracy), a następnie wykorzystany przez takiego agenta. Na podstawie odpowiedzi uzyskanych od klienta (w przypadku wymiaru organizacyjnego — rozmowa, w przypadku implementacyjnego — zestaw pytań zadanych przez system) zostaje utworzony profil klienta pozycjonujący go w jednym z czterech pól macierzy dojrzałości klienta (rys. 2.2). Zgromadzona wiedza o dojrzałości klienta zostaje przekazana do centralnego ośrodka zarządzania nazwanego AGENTOWYM SYSTEMEM ZARZĄDZANIA (opis w części dalszej). Warto dodać, że zgodnie z zasadą skalowalności modelu zestaw zadawanych pytań przez agenta dojrzałości klienta nie musi być zestawem zamkniętym. Zaleca się pozostawienie możliwości zadania dodatkowych pytań lub odnotowania zachowań klienta. Takie podejście jest możliwe zarówno w wymiarze organizacyjnym jak i informatycznym omawianego modelu.

W wymiarze organizacyjnym agentem dojrzałości klienta powinien być analityk biznesowy, który odnotowuje swoje obserwacje odnośnie klienta podczas rozmów.

W wymiarze implementacyjnym agent programowy ma przypisaną wewnętrzną bazę wiedzy pozwalającą na samodzielne rejestrowanie zachowań klienta lub jego przedstawiciela.

- **AGENTA DOJRZAŁOŚCI ZESPOŁU DOSTAWCY** — podobnie jak w przypadku agenta dojrzałości klienta, jest dedykowaną jednostką (członkiem zespołu lub programem) mającą za zadanie zmierzenie poziomu dojrzałości poszczególnych członków zespołu. Zadaniem agenta jest zdobycie odpowiedzi na podstawowe kompetencyjne pytania dotyczące umiejętności pracy w zespole oraz odnośnie doświadczeń w pracy nad projektami. Podobnie jak we wcześniejszym przypadku, model umożliwi rozszerzenie zadawanych pytań, a omawiany wcześniej kwestionariusz ma charakter pogładowy. Na podstawie pozyskanych odpowiedzi tworzony jest obraz poszczególnych członków zespołu w podziale na preferowane zadania (zgodnie z testem Belblina). Informacja zostaje przekazana do ASZ (Agentowego Systemu Zarządzania) i na podstawie zdobytych informacji menedżer nie tylko będzie w stanie ocenić dojrzałość swojego zespołu, ale także uwzględnić preferowane zadania dla poszczególnych członków, co powinno przełożyć się na zwiększoną efektywność wykonywanych zadań.

W wymiarze organizacyjnym zaleca się, aby był to lider zespołu, osoba o największym doświadczeniu lub najlepszych predyspozycjach do kierowania grupą ludzi. Taki lider przeprowadza rozmowy lub badania kwestionariuszowe dla określenia poziomu dojrzałości zespołu.

W wymiarze implementacyjnym agent programowy ma wpisany w zakres zadań przeprowadzenie testu Belblina oraz zadania zestawu pytań kompetencyjnych zgodnie z opisanym wcześniej instrumentem pomiarowym dojrzałości zespołu.
- **AGENTA ENTROPII PROJEKTU** — w budowanym modelu przyporządkowuje się dedykowanego agenta, którego zadaniem jest określenie poziomu entropii (zgodnie z wzorem 2.1). Posiłkuje się on informacjami pozyskanymi od menedżera projektu i od innych agentów (klienta i zespołu). Agent entropii powinien być przygotowany na wyszukiwanie wszelkich luk w przepływach informacyjnych. Powinien również posiadać umiejętności identyfikowania sprzecznych lub niepełnych informacji, bo to one przyczyniają się do wzrostu entropii projektowej. Efekt pracy agenta entropii powinien nie tylko obniżać poziom entropii, ale prowadzić również do uporządkowania strumienia informacji w projekcie.

W wymiarze organizacyjnym modelu powinien być to analityk biznesowy z największym stażem, który łatwo zauważy wszystkie braki w przepływach informacyjnych.

W wymiarze implementacyjnym jest to agent zaprogramowany na ocenę poziomu entropii.
- **AGENTA ZADANIOWEGO** — oznacza agenta odpowiedzialnego za realizację poszczególnych zadań zleconych w trakcie realizacji projektu. Zadania mogą dotyczyć zarówno pozyskiwania określonych informacji, jak również wykonywania zadań operacyjnych w projekcie informatycznym, takich jak programowanie, wykonywanie testów, wdrożenie oprogramowania u klienta itp.

W wymiarze organizacyjnym modelu agentami zadaniowymi są programiści, testerzy.

W wymiarze implementacyjnym jest to typowy agent odpowiedzialny za przeszukiwanie zasobów oraz wykonywanie przyporządkowanych mu zadań.

4.1.2 Specyfikacja agentów zarządzających

Agenty zarządzające stanowią kolejną kategorię agentów niezbędnych do realizacji zadań projektowych zgodnych z założeniami a2M. Uznano bowiem na podstawie obserwacji oraz wniosków wyciągniętych po analizie submodeli, że funkcjonowanie wszystkich agentów w systemie jest możliwe jedynie dzięki zastosowaniu odpowiedniej struktury. Agenty funkcjonalne tylko wtedy wykonają swoje zadania poprawnie, kiedy ich zadania zostaną wyraźnie określone przez agenty zarządzające. Natomiast przepływ informacji będzie miał miejsce wtedy, kiedy także agenty zarządzające będą miały określony swój zakres zachowań (zadań, które mogą wykonywać, działań,

które mogą podejmować). Na tej podstawie uznano, że niezbędne jest stworzenie hierarchii dla agentów zarządzających.

Agentowy System Zarządzania (ASZ) — stanowi centralny ośrodek decyzyjny w projekcie informatycznym, odpowiada za kluczowe decyzje oraz wyznaczanie kierunków prac. Składa się z agentów ułożonych w hierarchiczną strukturę odwzorowującą zależność pomiędzy menedżerem projektu i osobami mu podległymi a wspierającymi jego pracę (np. asystenci).

Uwzględnienie struktury zarządzania w projekcie informatycznym wydaje się nieodzowne. W realizowanych projektach (z udziałem autora pracy) zawsze wyodrębniano strukturę hierarchiczną. Kierownik projektu współpracował ze swoimi podwładnymi (asystentami), którzy dostarczali mu odpowiednich informacji. Taka obserwacja skłoniła autora do uwzględnienia struktury hierarchicznej również w opracowywanym **a2M**. Stąd Agentowy System Zarządzania prezentuje hierarchiczne zależności pomiędzy decyzyjnymi agentami w projekcie (tymi, które mają wpływ na jego przebieg).

Zgodnie z dobrymi praktykami zarządzania projektami (rozdział I), decyzje są podejmowane przez menedżera reprezentowanego w modelu przez AGENTA MENEDŻERA.

- **AGENT MENEDŻER** — reprezentuje najwyższy poziom zarządzania w projekcie. Oznacza agenta odpowiedzialnego za zarządzanie wszystkimi pozostałymi agentami. Należy rozumieć agenta menedżera jako instancję kierownika projektu, osoby zarządzającej wszystkimi pozostałymi agentami, ale również sprawującej nadzór nad przebiegiem projektu oraz sprawujący wszystkie funkcje kierownicze w projekcie.
- **AGENT KOORDYNATOR** — oznacza agenta wspomagającego prace menedżera. Może być to reprezentacja asystenta, który dostarcza „na żądanie” potrzebne raporty, sprawozdania. Pełni funkcje głównie kontrolne, koordynując zbieranie informacji o pozostałych elementach struktury projektu. Komunikuje się ze wszystkimi agentami funkcyjnymi (entropii, klienta, zespołu) oraz posiada zdolność do poszukiwania dodatkowych źródeł wiedzy niezbędnych dla realizacji projektu. W zależności od sytuacji identyfikuje niezbędne technologie, ekspertów etc.

W wymiarze organizacyjnym modelu taką rolę pełni najczęściej asystent kierownika.

W modelu implementacyjnym jest to agent najbardziej ukierunkowany na pozyskiwanie informacji z otoczenia odnośnie przydatnych dla realizacji projektu zasobów wiedzy.

- **AGENT WNIOSKUJĄCY** — to typ agenta, który posiada szczególne cechy pozwalające na przetwarzanie pozyskanych informacji oraz generowanie wyników na ich podstawie. To agent wnioskujący określa ostateczne poziomy dojrzałości klienta, zespołu dostawcy i poziom entropii projektu na podstawie informacji przekazanych przez agenty funkcjonalne. Ten typ agenta przygotowuje podstawowe parametry w celu dopasowania (adaptacji) poszczególnych agentów do realiów projektowych (agent adaptacji). Na poziomie agenta wnioskującego musi zostać uruchomiony mechanizm dopasowywania wszystkich składowych projektu do siebie. Rolą agenta wnioskującego jest zatem takie zastosowanie odpowiedniego mechanizmu łączenia elementów (robocza nazwa: funkcja A_MATCH), aby w zależności od dojrzałości klienta, dojrzałości zespołu dostawcy i prognozowanej entropii projektu, zastosować odpowiednie technologie, odpowiednie dobre praktyki metod zarządzania projektami, a także w razie potrzeby wybrać odpowiedniego eksperta. Ten agent może również zostać wykorzystany w procesach dopasowywania członków zespołu do realiów projektu (być może niektóre osoby należy przesunąć do innych zadań). Przeprowadza więc wstępną weryfikację kompetencji. Agent ten odpowiada także za pomiar ryzyka. Z racji tego, że spływają do niego wszystkie informacje, powinien w kooperacji z menedżerem projektu analizować wszystkie zagrożenia, jakie mogą pojawić się w przestrzeni projektu.

W wymiarze organizacyjnym jest to członek zespołu dostawcy o największych zdolnościach analitycznych.

W wymiarze implementacyjnym oznacza moduł wnioskujący (może być to agent pracujący z bazami wiedzy — bazami reguł i faktów).

Warto zaznaczyć, że omówione agenty (wnioskujący, koordynator menedżer) wchodzi w skład Agentowego Systemu Zarządzania, czyli centrum decyzyjnego w projekcie informatycznym. Centrum to może zostać rozbudowane o dodatkowe agenty w zależności od rozmiarów projektu. W sytuacji, w której kierownik decyduje się na stosowanie dobrych praktyk z metody PRINCE2 i powołuje komitet sterujący, w strukturze ASZ należy uwzględnić dodatkowych agentów. Budowa modelu stwarza warunki, aby kierownik projektu adaptował tylko te agenty, które odpowiadają jego strukturze. Tym samym w małych projektach funkcje AGENTA MENEDŻERA oraz AGENTA KOORDYNATORA można ograniczyć do jednego agenta. Takie podejście odpowiada przyjętemu w rozdziale III założeniu skalowalności modelu (możliwość rozbudowy), jak również założeniu adaptacyjności (dołączane dobre praktyki zarządzania projektami oraz modyfikacje z nimi związane).

4.1.3 Specyfikacja agentów wiedzy

Trzecią kategorią agentów są agenty wiedzy wykorzystujące zasoby wiedzy projektu. Do takich źródeł wiedzy należą eksperci, ale również technologie informatyczne, które posiadają funkcjonalności wspierające realizację zadań (np. tworzenie wykresu Gantta wbudowane w narzędziu do harmonogramowania projektu). Ważne staje się także wyodrębnienie źródeł wiedzy, z których będzie mógł korzystać agent wiedzy podczas realizacji projektu. Wyodrębniono następujące agenty wiedzy:

- **AGENT EKSPERT** — oznacza agenta reprezentującego eksperta dziedzinowego wspomagającego prace prowadzone w ramach danego projektu.
W wymiarze organizacyjnym modelu jest to ekspert w danej dziedzinie, który zostaje włączony w prace projektowe.
W wymiarze implementacyjnym agent ten komunikuje się z ekspertem w określonej sytuacji (np. za pośrednictwem WWW), pozyskując wskazówki do projektu.
- **AGENT DOBREJ PRAKTYKI** — agent odpowiedzialny za poszukiwanie metod zarządzania projektami, które mogą być przydatne w realizacji projektu oraz odpowiedzialny za wyodrębnienie z tych metod dobrych praktyk przydatnych do realizacji projektu.
W wymiarze organizacyjnym oznacza osobę lub kilka osób odpowiedzialnych za nadzorowanie realizacji projektu zgodnie z zaleceniami dobrych praktyk zarządzania projektami. Może to być np. przyporządkowanie osobom odpowiedzialności za poprawność wykresu wypalania zgodnie z metodą SCRUM lub wyznaczenie osoby odpowiedzialnej za czuwanie nad dokumentacją projektową zgodnie z zaleceniami RUP.
W wymiarze informatycznym oznacza dobrą praktykę, która składa się z zestawu cech i zachowań (cechy to wskazówki zawarte w dobrej praktyce, zachowania to uzasadnienia stosowania dobrej praktyki). Stąd też każda dobra praktyka opisana jest cechami i jest reprezentowana przez odpowiedniego agenta. Takie podejście ułatwia późniejsze łączenie dobrych praktyk (agentów) na potrzeby projektu (przy znanych zmiennych decyzyjnych).
- **Agent NARZĘDZIA** — agent poszukujący najbardziej adekwatnych rozwiązań technologicznych związanych z prowadzeniem projektu. Dotyczy to zarówno technologii wspomagających realizację projektu (jak technologie do zarządzania zespołem, rozliczania

projektu etc.), jak również technologii niezbędnych do wytworzenia systemu informatycznego.

W wymiarze organizacyjnym — najbardziej doświadczony informatyk, specjalista do spraw infrastruktury i oprogramowania.

W wymiarze implementacyjnym — jednostka programowa odpowiedzialna za poszukiwanie nowych rozwiązań technologicznych, przeszukująca zasoby Internetu, w tym stron branżowych.

- **Agent SZTUCZNEJ INTELIGENCJI (AI)** — jest to typ agenta odpowiedzialny za weryfikację pozyskanej wiedzy, tłumaczenie i poszukiwanie alternatywnych źródeł wiedzy wspierających zarządzanie projektem i jego realizację.

W wymiarze organizacyjnym powinien być to bardzo doświadczony uczestnik projektów informatycznych, który potrafi ocenić już pozyskaną wiedzę oraz jej wykorzystanie w oparciu o mechanizmy AI.

W wymiarze implementacyjnym agenty AI należy rozumieć jako agenty, które mogą wspomagać zarządzanie projektem poprzez przetwarzanie wiedzy, generowanie decyzji z wykorzystaniem systemów sztucznej inteligencji (ontologie, sztuczne sieci neuronowe itp.).

Powyższą specyfikację agentów (kategorie agentów: funkcjonalne, zarządzające, wiedzy oraz przykłady agentów w ramach każdej z kategorii) przygotowano na podstawie obserwacji oraz doświadczeń uzyskanych przy budowie submodeli. Opracowano także dobrą praktykę dla kierowników projektów, aby przed przystąpieniem do realizacji projektów określili, które agenty są niezbędne, a następnie włączali nowe agenty podczas realizacji projektu (gdy np. zrodzi się potrzeba włączenia eksperta). W tabeli 4.1 zestawiono wyodrębnione kategorie agentów wraz z przykładami oraz przewidywanym zakresem zachowań.

Tab. 4.1 Klasyfikacja agentów modelu a2M

KATEGORIA	TYPY AGENTÓW	ZACHOWANIA
Agenty funkcyjne	Agent ENTROPII Agent KLIENTA Agent ZESPOŁU Agent ZADANIOWY	Przeprowadzenie przetwarzania wstępnego, czyli pozyskania informacji dotyczących głównych parametrów decyzyjnych w projekcie.
Agenty zarządzające	Agent MENEDŻER Agent KOORDYNATOR Agent WNIOSKUJĄCY	Zarządzanie strumieniem informacji w projekcie, prowadzenie bieżących zadań, sprawowanie funkcji menedżerskich.
Agenty wiedzy	Agent EKSPERT Agent NARZĘDZIA Agent DOBRYCH PRAKTYK Agent OCENY Agent SZTUCZNEJ INTELIGENCJI (AI)	Dostarczenie wiedzy niezbędnej dla poprawnego realizowania projektów informatycznych.

Źródło: opracowanie własne

Należy zaznaczyć również, że wyodrębnienie agentów na potrzeby modelu **a2M** jest niezbędne z punktu widzenia późniejszej ich adaptacji. Specyfikacja agentów jest pierwszym etapem budowy **a2M**. Kolejnym krokiem jest powiązanie ich relacjami. Podstawowym typem relacji, jakie należy wyodrębnić wśród zdefiniowanych agentów, są relacje określające zależności między nimi. Określenie podległości pomiędzy agentami determinuje jednocześnie sposób reagowania na komunikaty (zlecenia).

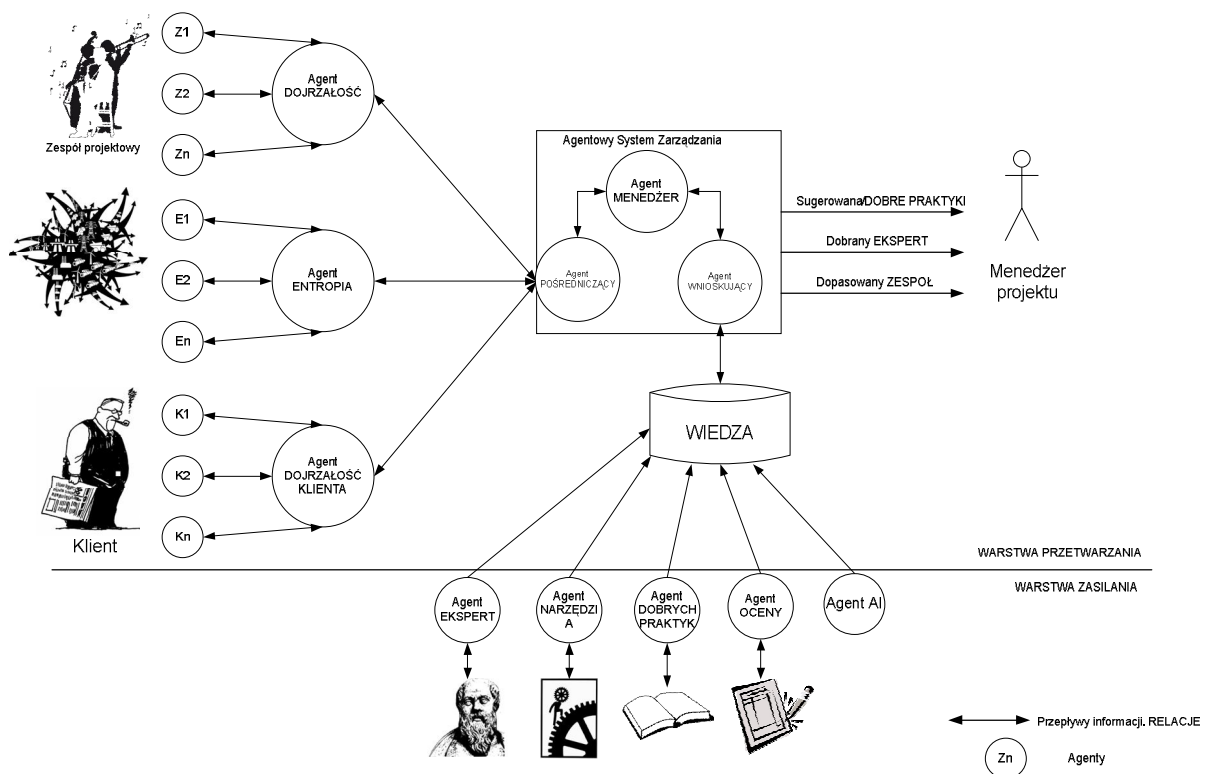
Po specyfikacji agentów i ich kategorii, można przystąpić do wyodrębnienia zależności między nimi.

4.2 Agentowy model ogólny zarządzania projektami informatycznymi

Przygotowana specyfikacja adaptacyjnego agentowego modelu zarządzania projektami pozwoliła na wyodrębnienie trzech kategorii agentów — funkcjonalnych, zarządzających oraz agentów wiedzy. Spośród agentów należących do tych kategorii możliwa jest do zdefiniowania struktura projektu. Kierownik projektu powinien wyodrębnić elementy struktury, które występują w jego projekcie. Korzystając z modelu **a2M** kierownik otrzymuje zestaw sugerowanych elementów struktury projektowej. Konieczne stają się także wyodrębnienie elementów struktury projektowej oraz określenie zarządzania tymi elementami. Ważne jest także, z punktu widzenia realizacji zadań przez agenty, dobranie odpowiednich dobrych praktyk zarządzania projektem informatycznym. Stąd też prezentowany model powinien także wskazywać na kierunki usprawnień organizacyjnych dla kierowników projektów. Model **a2M** powinien także być możliwy do implementacji w dowolnym środowisku agentowym (zgodnie z zasadami modelowania MDA). Przyjęto także, że model usprawniający zarządzanie projektami informatycznymi powinien mieć dwa wymiary — organizacyjny oraz implementacyjny.

Wymiar organizacyjny modelu należy rozpatrywać właśnie w perspektywie usprawnień zarządczych — doskonalących realizację zadań przez kierownika projektu podczas przygotowań do uruchomienia projektu. Wymiar implementacyjny modelu oznacza, że model stanowi architekturę gotową do implementacji w dowolnym środowisku agentowym.

Opierając się na zaprezentowanej uprzednio specyfikacji agentów, biorąc także pod uwagę nowy trójkąt ograniczeń, jak też uwzględniając, że realizacja projektu przebiega w ramach klasycznego trójkąta ograniczeń, na rys. 4.1 przedstawiono ogólny model **a2M**. Model ogólny posiada też reprezentację szczegółową.



Rys. 4.1 Model a2M w postaci ogólnej

Źródło: opracowanie własne

W modelu tym zintegrowano wyspecyfikowane powyżej agenty. Przy ich tworzeniu uwzględniono przyjętą kategoryzację na agenty funkcjonalne, zarządzające i wiedzy. Wyodrębniono agenty wiedzy (AGENT EKSPERT, AGENT NARZĘDZIA, AGENT METODY/AGENT DOBRYCH PRAKTYK, AGENT SZTUCZNEJ INTELIGENCJI, AGENT OCENY). Wyróżniono także agenty funkcjonalne, takie jak AGENT DOJRZAŁOŚCI KLIENTA, AGENT ENTROPII oraz AGENT DOJRZAŁOŚCI ZESPOŁU. Zadaniem tych agentów jest komunikacja z innymi agentami uczestniczącymi w projekcie (AGENTY ZADANIOWE) w celu określenia wartości zmiennych decyzyjnych. Na tej podstawie AGENT DOJRZAŁOŚCI KLIENTA, stosując instrument pomiarowy (jak opracowany na potrzeby tej pracy prezentowany w załączniku kwestionariusz), określa poziom dojrzałości klienta uczestniczącego w projekcie. Analogicznie zachowują się agenty w przypadku pomiaru entropii projektowej oraz dojrzałości zespołu. Zebrane przez AGENTY FUNKCJONALNE informacje pozwalają na określenie nowego trójkąta ograniczeń. Wyniki pomiarów przesłane zostają do Agentowego Systemu Zarządzania dla określenia stanu ograniczeń projektowych (zakłada się, że analizę ograniczeń przeprowadził AGENT MENEDŻER lub AGENT WNIOSKUJĄCY). W oparciu o te ograniczenia dobierane są dobre praktyki.

Aby jednak adaptacja dobrych praktyk do ograniczeń projektowych była możliwa, kierownik projektu powinien znać dobre praktyki, które mogą być stosowane. Dobre praktyki, zgodnie z przyjętym w modelu **a2M** założeniem, są również agentami (mają cechy i zachowania, czyli przypadki zastosowania). Stąd też AGENT MENEDŻER powinien komunikować się z AGENTEM DOBREJ PRAKTYKI w celu wybrania tych dobrych praktyk, które są najlepiej dopasowane do ograniczeń projektowych. W tym jednak momencie rodzi się pytanie, w jaki sposób dopasować dobre praktyki. Jest to możliwe z wykorzystaniem odpowiedniej funkcji dopasowania. Na potrzeby modelu **a2M** opracowano w rozdziale II funkcję `A_MATCH`. Funkcja ta pokazuje, w jaki sposób dobierać dobre praktyki w zależności od wartości zmiennych decyzyjnych. Stąd też nieodzowna jest rola AGENTA WNIOSKUJĄCEGO, który poprzez odwołanie do AGENTÓW WIEDZY powinien pozyskać sugestie dotyczące dopasowania dobrych praktyk. Funkcja `A_MATCH` może być reprezentowana przez AGENTA OCENY. W ten sposób do Agentowego Systemu Zarządzania powinien trafić zestaw niezbędnych informacji do przeprowadzenia takiej oceny. W procesie podejmowania decyzji z wykorzystaniem **a2M** z jednej strony wpływają informacje o ograniczeniach projektowych, z drugiej o dobrych praktykach wraz z uwzględnieniem ich użycia (dopasowania). Ponadto obecność AGENTA WNIOSKUJĄCEGO zapewnia adaptowanie tylko tych dobrych praktyk, które są niezbędne dla obniżenia entropii projektu, podwyższenia dojrzałości klienta oraz dojrzałości zespołu dostawcy. Ponadto w zależności od tych zmiennych decyzyjnych można określić zapotrzebowanie na inne agenty wiedzy (np. eksperta). Dzięki takiemu podejściu kierownik projektu może podjąć także decyzje odnośnie włączenia AGENTA EKSPERTA do realizowanego projektu.

Podczas omawiania modelu **a2M** należy także wskazać na skalowalność modelu ogólnego. Przyjęto bowiem zasadę, że w zależności od złożoności struktury projektowej może zachodzić zapotrzebowanie na zwiększenie bądź zmniejszenie liczby agentów. Jeżeli bowiem zespół projektowy będzie składał się z wielu członków, można dokonać multiplikacji AGENTA DOJRZAŁOŚCI ZESPOŁU do wielu takich agentów. Należy również przyjąć, że skoro członkowie zespołu stanowią AGENTY ZADANIOWE (Z_1 , Z_2 itd.), otrzymując zlecenie od AGENTA DOJRZAŁOŚCI ZESPOŁU wymagające wypełnienia określonych kwestionariuszy (instrumentów pomiarowych), prezentują efekty wykonanych zadań AGENTOWI DOJRZAŁOŚCI.

Po prezentacji modelu ogólnego należy przedstawić znaczenie tego modelu dla kierowników projektów w obydwu wymiarach: w wymiarze organizacyjnym oraz w wymiarze implementacyjnym.

Wymiar organizacyjny modelu ogólnego

Model ogólny w wymiarze organizacyjnym pełni funkcję zestawu procedur pozwalających menedżerom na usprawnienie zarządzania projektami zarządzania projektami informatycznymi

w kilku obszarach. Założony pomiar dojrzałości zespołu pozwala na trafniejsze decyzje odniesione do członków zespołu, pozwala tym samym na lepsze dopasowanie uczestników do pozycji w zespole (np. predyspozycje zgodne z testem Belblina do pełnienia roli lidera albo wykonawcy). W ten sposób kierownicy usprawniają procesy związane z zarządzaniem zadaniami. Mogą bowiem lepiej dopasowywać zadania do oczekiwań i możliwości członków zespołu. Zastosowanie modelu **a2M** pozwala także na poprawę relacji z klientem. Zbadanie poziomu dojrzałości klienta umożliwia podejmowanie właściwych decyzji co do uczestniczenia klienta w projekcie. Wreszcie pomiar entropii projektowej daje kierownikom projektów obraz ryzyka projektowego (składnika oceny entropii projektu).

Zastosowanie modelu ogólnego pozwala także na zalecenie kierownikom projektów tworzenia struktur zarządczych (określonych mianem agentowego systemu zarządzania), w skład których powinien wchodzić kierownik projektu oraz dwa agenty asystujące — AGENT KOORDYNATOR oraz AGENT WNIOSKUJĄCY. Agenty te odpowiadają funkcjom zarządczym. Liczba agentów powinna być zależna od złożoności samego projektu.

W modelu zaleca się wykorzystanie co najmniej dwóch typów agentów asystujących:

- AGENT KOORDYNATOR — nadzorujący wszystkie analizy,
- AGENT WNIOSKUJĄCY — ułatwiający podejmowanie decyzji.

Innym usprawnieniem organizacyjnym wynikającym z zastosowania modelu ogólnego jest zwrócenie uwagi kierownikom projektów na wyodrębnienie wszystkich AGENTÓW WIEDZY. Zaleca się wyodrębnienie wszystkich możliwych agentów wiedzy, które mogą wspomagać realizację zadań w projekcie. Kierownicy powinni zatem zwrócić uwagę na możliwość uwzględnienia eksperta w projekcie, ale także na konieczność uwzględnienia dobrych praktyk, które również należy traktować jako agenty dobierane do projektu przez dedykowanego agenta (AGENT DOBRYCH PRAKTYK).

Należy jednocześnie podkreślić, że podczas opracowywania modelu **a2M** stworzono zestaw zaleceń dla kierowników, jak powinni organizować elementy struktury projektowej korzystając z modelu **a2M**. Można zatem przedstawić te zalecenia w postaci dobrej praktyki składającej się z następujących po sobie kroków:

1. Wyodrębnij elementy struktury projektu.
2. Zdefiniuj te elementy jako agenty.
3. Określ zbiór dobrych praktyk potrzebnych do realizacji projektu.
4. Utwórz Agentowy System Zarządzania w projekcie.
5. Zmierz poziom dojrzałości zespołu dostawcy, dojrzałości klienta oraz entropii projektu.
6. Określ relacje między wszystkimi agentami.
7. Dobierz dobre praktyki do ograniczeń.

Poniżej przedstawiono przykłady usprawnionych decyzji przez zastosowanie modelu ogólnego

Tab. 4.2 Przykłady usprawnionych decyzji projektowych przez zastosowanie modelu ogólnego

Lp.	PRZYKŁAD DECYZJI	USPRAWNIENIE
1	Określenie pozycji w zespole dla danego członka zespołu	Mierzenie dojrzałości zespołu poprzez sprawdzenie dojrzałości poszczególnych jego członków pozwala kierownikom na podjęcie właściwych decyzji odnośnie pełnienia przez nich określonych pozycji w zespole. Celem tego usprawnienia nie jest przypisanie ról (takich jak programista czy tester), ale określenie pozycji w zespole, czyli jaki charakter zadań jest preferowany przez uczestnika (np. ktoś woli wykonywać zadania wymagające skupienia).

2	Określenie roli klienta w projekcie	Na podstawie profilu klienta określonego za pomocą poziomu dojrzałości klienta kierownik może lepiej wkomponować klienta w zespół. Klienta odpowiedniego i dopasowanego kierownik powinien włączać na stałe, lecz w przypadku niedopasowanego klienta może podjąć inne decyzje, przyjmując, że nie on będzie stałym członkiem zespołu.
---	-------------------------------------	--

Źródło: opracowanie własne

Wymiar implementacyjny modelu ogólnego

Wymiar implementacyjny modelu **a2M** stanowi ogólną architekturę systemu agentowego do zarządzania projektami informatycznymi, która może być zaimplementowana w narzędziach wspierających pracę kierowników projektów. Podstawowym założeniem przyjętym dla tej architektury jest to, że każdy element (składnik) struktury rzeczywistego projektu może zostać odzwierciedlony poprzez agenta programowego. Agentowi programowemu przypisuje się zestaw stanów, które może przyjmować (odzwierciedlenie cech uczestników projektów) oraz akcji (odzwierciedlających zachowania uczestników w projekcie). Kombinacja tych dwóch czynników określa agenta jako niezależny fragment systemu podejmujący akcję w oparciu o otrzymany komunikat od agenta zlecającego wykonanie określonego zadania.

Każdy zatem agent zdefiniowany w części poświęconej specyfikacji zostaje zaimplementowany w systemie agentowym i podejmuje akcje wyłącznie po uzyskaniu komunikatu od innego agenta. Pozwala to na pełną kontrolę przepływu informacji oraz porządkuje realizację zadań.

Zaimplementowanie modelu **a2M** w dowolnym środowisku programistycznym prowadzi do przyspieszenia analiz związanych z określeniem wartości zmiennych decyzyjnych. Zaimplementowane agenty mogą bowiem komunikować się z klientem oraz członkami zespołu w celu zdiagnozowania poziomu dojrzałości klienta czy też dojrzałości zespołu reprezentującego dostawcę w projekcie informatycznym. Ponadto implementując agenty w systemie agentowym, można korzystać z ich możliwości obliczeniowych. Oznacza to zatem, że dokonując analizy poziomu dojrzałości klienta albo dostawcy, agenty mogą przysyłać wyniki do kolejnego agenta, którego zadaniem będzie przeprowadzenie obliczeń. Takim agentem jest AGENT WNIOSKUJĄCY, który w wymiarze implementacyjnym odpowiada za przetwarzanie danych pozyskanych (przesłanych) od innych agentów. Zaimplementowane agenty mogą znacznie przyspieszyć analizy. O ile w wymiarze organizacyjnym kierownik lub jego asystent (AGENT KOORDYNATOR lub AGENT WNIOSKUJĄCY) dokonywali samodzielnie określenia poziomu dojrzałości klienta, dostawcy oraz entropii projektowej na podstawie kwestionariuszy wypełnionych przez uczestników projektów (rzeczywistego klienta, członków zespołu projektowego itp.), o tyle w wymiarze implementacyjnym wnioskowanie może odbywać się w oparciu o zaimplementowane algorytmy, przez co kierownik projektu może uzyskać takie wyniki o wiele szybciej.

Kolejną zaletą zaimplementowania modelu agentowego do postaci systemu agentowego jest możliwość przeprowadzania symulacji zachowań poszczególnych agentów. Implementując bowiem agenty jako jednostki posiadające stany i akcje odpowiadające cechom i zachowaniom rzeczywistych uczestników projektów, można zlecić agentom programowym wykonanie określonych zadań i sprawdzić ich zachowanie. Dzięki temu system agentowy stanowi jednocześnie środowisko symulacji i pozwala przewidywać, jak będzie wyglądała współpraca np. z klientem albo jak będzie przebiegać wykonywanie zadań przez członków zespołu (AGENCY ZADANIOWE).

Kolejną zaletą wymiaru implementacyjnego jest możliwość przeniesienia wszystkich założeń modelu organizacyjnego do poziomu narzędzia, które może być wykorzystywane przez kierownika projektu. Zaimplementowany system agentowy można wtedy potraktować jako moduł rozszerzający typowe narzędzie (służące np. do harmonogramowania projektów) o dodatkowe funkcje pozwalające np. analizować nowy trójkąt ograniczeń projektowych.

Zastosowanie modelu ogólnego zarówno do usprawnienia działań organizacyjnych zespołu dostawcy, jak też do tworzenia struktur agentowych, wydaje się słuszne. Rodzi się tym samym pytanie, czy model może być stosowany niezależnie od zachowań zespołu dostawcy. Jeżeli kierujący projektem nie przeprowadza procesów symulowania zachowań poszczególnych składowych struktury lub nie ma dostępu do zaimplementowanej wersji modelu, może korzystać z wariantu organizacyjnego, traktując model jako gotową metodę zarządzania projektem lub zbiór dobrych praktyk. W takim przypadku jego zastosowanie przyniesie określone korzyści w postaci uporządkowania procesów zarządzania w ramach złożonych struktur projektowych.

Oczywiście zastosowanie jedynie wymiaru organizacyjnego jest również możliwe, jednak na pewnym etapie analizy nowego trójkąta ograniczeń zaistnieje potrzeba przejścia z poziomu organizacyjnego do poziomu implementacyjnego.

Docelowym rozwiązaniem staje się wykorzystanie modelu zarówno na poziomie organizacyjnym, jak i implementacyjnym. W takim przypadku w pierwszej kolejności kierownik powinien uporządkować relacje między poszczególnymi elementami struktury w wymiarze organizacyjnym, postrzegając te elementy jako agenty. Następnie powinien przystąpić do zarządzania nimi z wykorzystaniem zaimplementowanego modelu ogólnego.

4.3 Agentowy model szczegółowy zarządzania projektami informatycznymi

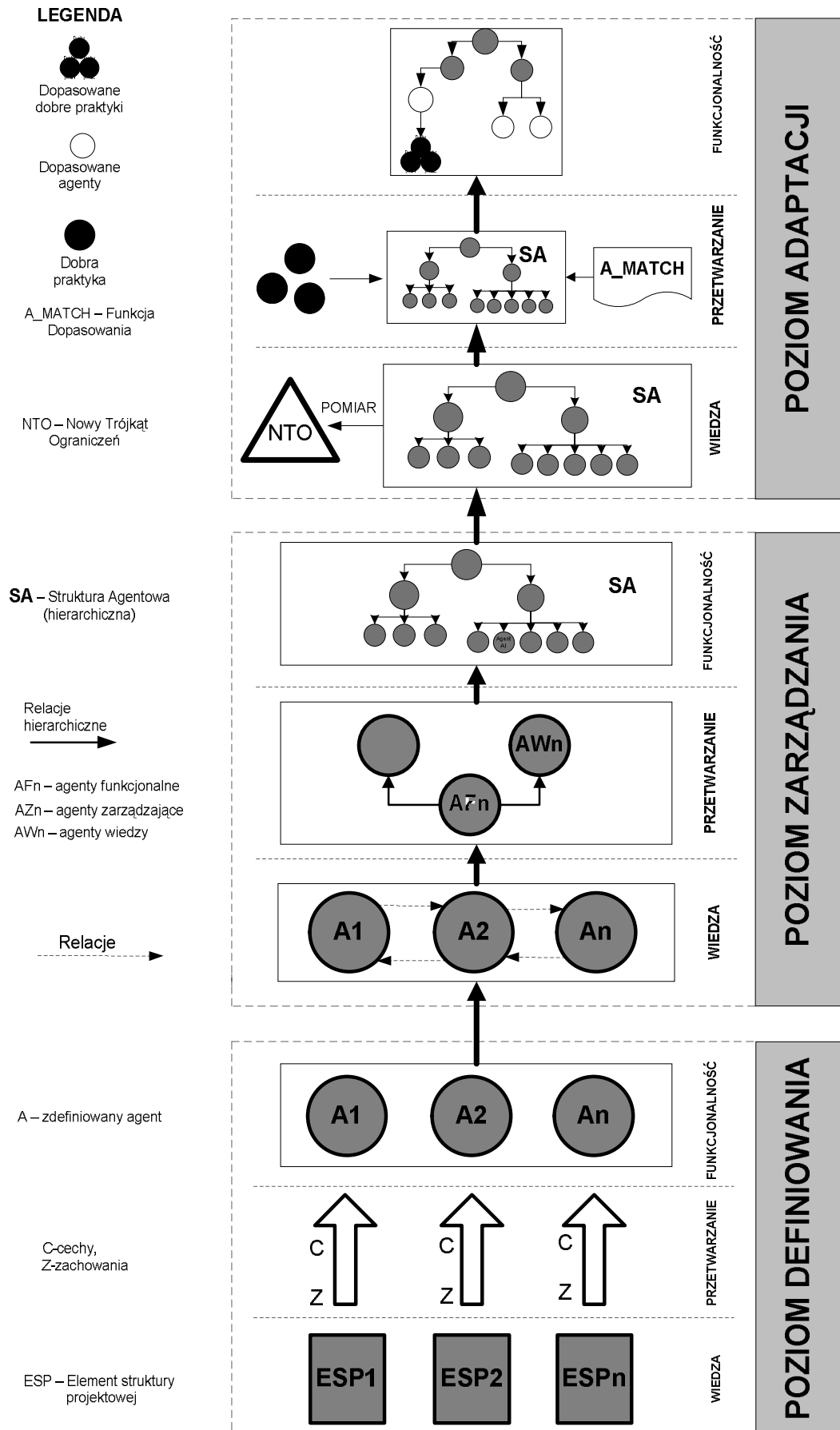
Omówiony w poprzedniej części model ogólny **a2M** zarządzania projektami uwzględniał wszystkie kategorie agentów i prezentował ogólną zasadę wykorzystania modelu, zarówno do usprawnień organizacyjnych w zarządzaniu projektami informatycznymi, jak również do usprawnień implementacyjnych. Dla pokazania sposobu wykorzystania modelu przez zespół dostawcy zaprezentowano wersję szczegółową **a2M**. Model szczegółowy powstał w efekcie dekompozycji zagregowanych zmiennych decyzyjnych, a podstawą do jego opracowania były doświadczenia wyciągnięte z badań nad submodelami (prezentowane w rozdziale III) oraz późniejsze wnioski i założenia do budowy modelu ogólnego. Stąd też model szczegółowy, podobnie jak model ogólny, utrzymuje zasadę dualizmu (organizacyjno-implementacyjną). Omówione w modelu szczegółowym poziomy oraz warstwy mogą być wykorzystane zarówno w wymiarze organizacyjnym, jako zestawy wskazówek dla kierowników jak organizować elementy struktury projektowej, jak również w wymiarze implementacyjnym, wspierając modelowanie agentów.

Konieczność zachowania zasady dualizmu modelu (czytelnego zarówno dla kierowników, jak i osób implementujących w środowiskach agentowych) pozwoliła na dekompozycję modelu do trzech poziomów, zgodnych z wcześniej przyjętymi założeniami (rozdział III). Przyjęto zatem, że skoro wstępnie kierownicy wyodrębniają elementy struktury projektowej, model na najniższym poziomie (definiowania) powinien odzwierciedlać definiowanie tych elementów i ich transformację do postaci agentów. Definicja agentów oraz określenie relacji pomiędzy nimi pozwala rozpatrywać model szczegółowy na wyższym poziomie nazwanym poziomem zarządzania. Po zdefiniowaniu agentów i utworzeniu hierarchii między nimi model szczegółowy należy rozpatrywać na najwyższym poziomie, nazwanym poziomem adaptacji. Na tym poziomie następuje dopasowanie dobrych praktyk zarządzania projektami, które są niezbędne z punktu widzenia nowego trójkąta ograniczeń oraz dopasowanie poszczególnych agentów do projektu (nie wszystkie agenty zdefiniowane wcześniej są niezbędne do realizacji zadań).

Należy zaznaczyć także, że każdy z poziomów modelu jest osiągany (powstaje) w oparciu o wykonanie określonych zadań zgodnych z logiką architektury trójwarstwowej. Aby więc osiągnąć określony poziom modelu, konieczne jest posiadanie odpowiednich danych, zapewnienie mechanizmów przetwarzania oraz uzyskanie interfejsu w postaci zestawu funkcjonalności. Takie podejście pozwoliło więc rozpatrywać każdy poziom modelu jako układ trzech warstw, gdzie w miejsce danych pojawia się wiedza, następnie uruchamiany jest mechanizm korzystania z tej wiedzy

i jej przetwarzania. Na podstawie pozyskanej i przetworzonej wiedzy uzyskiwana jest funkcjonalność modelu. Stąd też uznano, że każdy z poziomów modelu reprezentowany jest przez trzy warstwy — wiedzy, przetwarzania oraz funkcjonalności. Na rys. 4.2 przedstawiono model **a2M** w wersji szczegółowej.

Należy pamiętać, że nałożenie warstw wynika z potrzeby zapewnienia możliwości implementacji modelu szczegółowego **a2M**. W założeniach do budowy systemu przyjęto, że opracowywany model szczegółowy ma zapewnić zarówno organizacyjne jak i implementacyjne środowisko. O ile struktura organizacyjna jest prezentowana na rys. 4.2, o tyle na rysunku tym nie jest widoczna warstwa implementacyjna (poza strukturą agentów). Stąd też zaproponowano do implementacji struktury organizacyjnej trójwarstwową budowę w ramach, której widoczne elementy struktury organizacyjnej stanowią warstwę prezentacji. Pozostałe dwie warstwy zasobów i przetwarzania są niezbędne dla wprowadzenia danych i ich przetworzenia. Dane te dotyczą struktury organizacyjnej zespołu dostawcy.



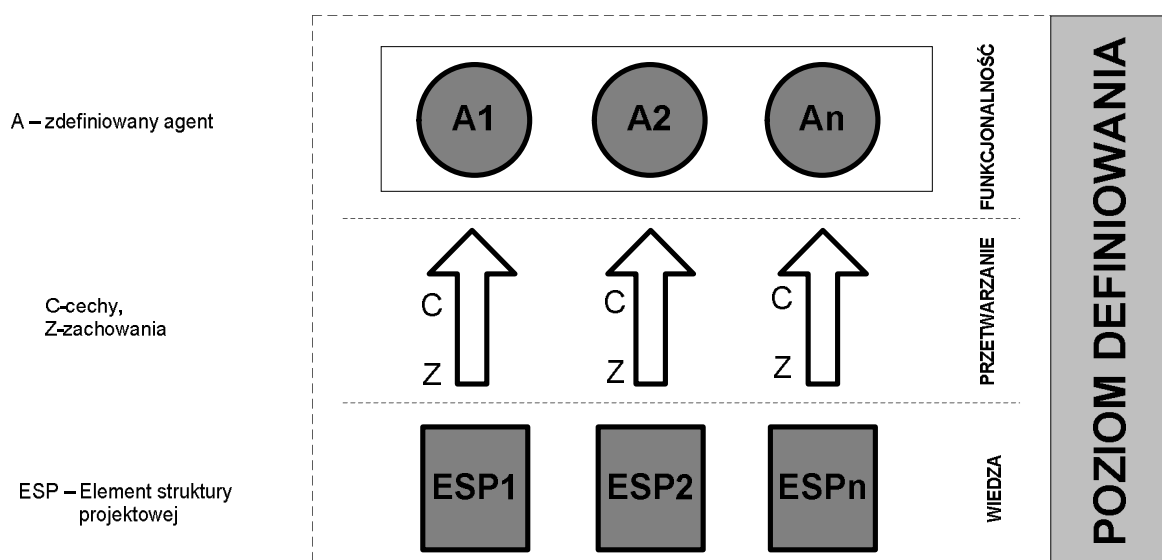
Rys. 4.2. Model w postaci szczegółowej

Źródło: opracowanie własne

Jak wynika z rysunku 4.2, model szczegółowy **a2M** należy rozpatrywać na trzech poziomach. Każdy z tych poziomów wraz z uwzględnieniem warstw został omówiony w dalszej części. Jednak na rysunku zaznaczono sekwencję osiągania określonych poziomów modelu. Zarządzanie projektem informatycznym z wykorzystaniem modelu szczegółowego nie może znajdować się na poziomie wyższym bez osiągnięcia poziomu wcześniejszego. Tak więc osiągnięcie poziomu zarządzania jest możliwe tylko po zdefiniowaniu wszystkich agentów na poziomie definiowania. Natomiast osiągnięcia poziomu adaptacji dobrych praktyk w zależności od zmiennych decyzyjnych nowego trójkąta ograniczeń można dokonać tylko wtedy, gdy agenci zostaną ujęte w odpowiedniej strukturze za pomocą wzajemnych relacji (co realizowane jest na poziomie zarządzania).

Konstrukcja modelu szczegółowego — poziom definiowania (etap I)

Poziom definiowania stanowi najniższy poziom modelu szczegółowego **a2M**. Na tym poziomie następuje wyodrębnienie wszystkich elementów struktury projektowej, które mogą zostać wykorzystane do realizacji projektu informatycznego. Na poziomie definiowania powinno również nastąpić transformowanie tych elementów do postaci agentów. Dlatego też poziom definiowania jest osiągnięty wtedy, gdy wszystkie elementy struktury mają swoją reprezentację w postaci agentów, czyli opisywane są przez zestaw cech i zachowań. Podczas definiowania tych elementów i ich transformacji do postaci agentów należy dokonać kategoryzacji agentów zgodnej ze specyfikacją, tj. agenty wiedzy, agenty funkcjonalne oraz agenty zarządzające. Rozpatrywanie agentów z punktu widzenia kategorii porządkuje proces ich definiowania.



Rys. 4.3 Poziom definiowania modelu szczegółowego a2M

Źródło: opracowanie własne

Jak wynika z rysunku 4.3, każdemu obiektowi w modelu zostaje przyporządkowany agent. Aby zakończyć definiowanie agentów na poziomie definiowania modelu szczegółowego **a2M**, niezbędne jest zdefiniowanie warstw tego poziomu (zgodnie z założeniem architektury trójwarstwowej). W tabeli 4.3 zdefiniowano warstwy oraz przedstawiono oczekiwane rezultaty procesu definiowania w ramach każdej z warstw.

Tab. 4.3 Warstwy implementacyjne i oczekiwane produkty dla poziomu definiowania modelu szczegółowego a2M

WARSTWA POZIOMU	REZULTAT
FUNKCJONALNOŚĆ	Zdefiniowano podstawowe funkcjonalności agentów — agent posiada cechy ENCJI, którą reprezentuje (jeśli człowiek — kompetencje, jeśli metoda — dobre praktyki).
PRZETWARZANIE	Następuje transformacja elementów struktury do postaci agentów. Każdy członek projektu ma reprezentację w postaci AGENTA. Każdy ekspert traktowany jest jako AGENT. Każda dobra praktyka traktowana jest jako AGENT za pośrednictwem AGENTA DOBRYCH PRAKTYK.
WIEDZA	Wyodrębnione elementy struktury projektu informatycznego: <ul style="list-style-type: none"> - członkowie projektów, - dobre praktyki, - eksperci i ich wiedza, - klient i/lub jego przedstawiciel, - kierownik i asystenci.

Źródło: opracowanie własne

Zgodnie z tabelą 4.3 na najniższym poziomie (warstwa wiedzy) występują elementy zasilania wiedzą. W tym przypadku jest to wiedza o poszczególnych elementach struktury projektowej. Wyodrębnienie tych elementów wynika z doświadczeń wyniesionych z uczestnictwa w projektach oraz z analizy literatury poświęconej zarządzaniu projektami [18,39]. Wyodrębnienie tych elementów struktur jest kluczowe z punktu widzenia późniejszej adaptacji agentów — im więcej elementów zostanie wyłonionych, tym mniejsze będzie ryzyko pominięcia elementu niezbędnego dla realizacji zadań projektowych. Wiedza zgromadzona o elementach struktury projektowej jest wykorzystywana następnie podczas procesów dopasowywania agentów przy tworzeniu relacji między nimi. W ramach tej warstwy następuje określenie poszczególnych elementów struktury projektowej, takich jak klient, kierownik projektu, asystenci kierownika, wykonawcy zadań, modele ocenowe, eksperci.

W ramach warstwy przetwarzania (zgodnie z podejściem trójwarstwowym) powinno nastąpić przetworzenie wiedzy zgromadzonej w zasobach na potrzeby struktury projektowej. Warstwa przetwarzania dla poziomu definiowania modelu szczegółowego **a2M** pozwala na przeprowadzenie transformacji wyodrębnionych elementów struktury projektowej do postaci agentów. Elementy struktury projektowej postrzegane są jako pojedyncze, niezależne byty (typowe atrybuty agentów). Przyjmuje się więc w ramach procesów przetwarzania, że każdy ekspert, członek zespołu czy nawet metoda zarządzania projektami przybiera postać agenta opisywanego zestawem cech oraz zachowań. Zgodnie z takim podejściem możliwa jest pełna definicja agentów w najwyższej warstwie (funkcjonalności) tego poziomu.

Ostatnia warstwa w modelu trójwarstwowym ma za zadanie prezentować efekty uzyskane w warstwie przetwarzania. Po uwzględnieniu wszystkich elementów struktury projektowej jako agenty można przypisać im określone cechy i zachowania. W ten sposób powstają funkcjonalności modelu na najniższym poziomie dla warstwy definiowania. Każdy element struktury projektowej ma reprezentację w postaci agenta oraz każdemu agentowi przyporządkowuje się zestaw cech i zachowań, które określają jego zakres zadań w ramach projektu. W tabeli poniżej zestawiono przykłady cech i zachowań wybranych agentów:

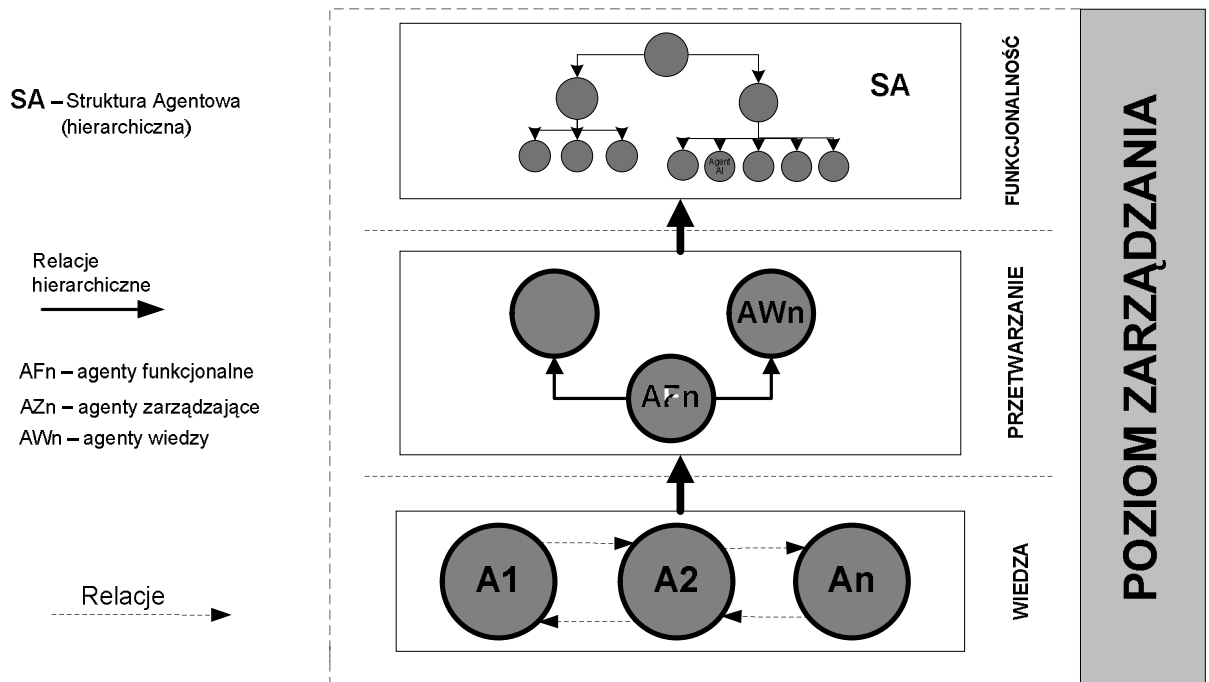
Tab. 4.4 Agenty definiowane w warstwie definiowania modelu szczegółowego a2M

KATEGORIA AGENTA	AGENT	PRZYKŁADOWE CECHY	PRZYKŁADOWE ZACHOWANIA
Agenty funkcjonalne	AGENT ZADANIOWY	-doświadczenie w projektach (w latach) -znajomość C#	- szacuje czas implementacji fragmentu systemu - komunikuje AGENTOWI MENEDŻEROWI stan postępów
Agenty zarządzające	AGENT MENEDŻER	-liczba kierowanych projektów -znajomość dobrych praktyk zarządzania	- uruchamia AGENTA KOORDYNATORA - przesyła zlecenie wykonania zadania
Agenty wiedzy	AGENT EKSPERT	- lata doświadczeń w projektach - co jest główną domeną eksperta - jaka jest jego dostępność	- komunikuje się bezpośrednio z menedżerem - wykrywa nieprawidłowości w strumieniach informacyjnych

Źródło: opracowanie własne

Konstrukcja modelu szczegółowego — poziom zarządzania (etap II)

W celu przeprowadzania adaptacji dobrych praktyk do wartości zmiennych nowego trójkąta ograniczeń (co jest osiągnięte na najwyższym poziomie modelu) niezbędne okazało się budowanie relacji pomiędzy agentami. Jeżeli zatem wyodrębniono elementy struktury projektu i dokonano ich transformacji do postaci agentów poprzez określenie ich cech i zachowań, należy określić relacje pomiędzy agentami. Stworzenie relacji jest niezbędne zarówno z punktu widzenia przepływów informacji pomiędzy agentami w projekcie, ale również porządkuje przepływy komunikatów. Dzięki przedstawieniu relacji między agentami, mechanizmy komunikacji i oddziaływania pomiędzy agentami są widoczne dla kierującego projektem.



Rys. 4.4 Poziom zarządzania modelu szczegółowego a2M

Źródło: opracowanie własne

Zwrócono także uwagę, że każdy element projektu można przedstawić, jako niezależną encję z określonym zestawem informacji (wiedzy) oraz interakcji pomiędzy poszczególnymi elementami

struktury projektowej (zgodnie z założeniami definiowania klasycznych systemów agentowych). Zauważono także, że jeżeli każdy element struktury projektu ma reprezentację w postaci agenta, to te agenty będą mogły podejmować akcje i przyjmować stany w określonych sytuacjach projektowych w zależności od ich cech i zachowań (członkowie, klient, ekspert). Zdefiniowano także te relacje oraz stany, jakie mogą przyjmować określone agenty. Dlatego drugi poziom adaptacyjnego agentowego modelu do zarządzania projektami w swoich głównych założeniach ma za zadanie zdefiniować strukturę — sieć powiązań pomiędzy agentami biorącymi udział w projekcie.

Zauważono także, że pomiędzy agentami występują dwa typy relacji:

- zależności hierarchiczne (typowe dla struktur zarządzania — nadrzędność/podrzędność, odzwierciedlające zależności w strukturze organizacyjnej),
- zależności interakcyjne (komunikacyjne — związane z przepływem informacji oraz przekazywaniem sobie zadań).

Poziom zarządzania, podobnie jak poziom definiowania modelu, może być implementowany z zastosowaniem podejścia trójwarstwowego. W tabeli 4.5 zestawiono produkty oczekiwane dla każdej warstwy poziomu zarządzania modelu szczegółowego **a2M**.

Tab. 4.5 Warstwy implementacyjne i oczekiwane produkty dla poziomu zarządzania modelu szczegółowego a2M

WARSTWA MODELU	REZULTAT/ZAWARTOŚĆ
FUNKCJONALNOŚĆ	Dobieranie/uruchamianie agentów w zależności od wejściowych parametrów.
PRZETWARZANIE	Wyodrębnienie agentów nadzorujących wszystkie procesy/komunikaty i przepływ informacji między agentami funkcyjnymi (AGENT MENERDZER, AGENT KOORDNATOR, AGENT WNIOSKUJĄCY). Stworzenie struktury agentów zarządzających przepływami informacji w systemie. Umiejętność tworzenia i koordynowania relacji pomiędzy agentami niższego szczebla.
WIEDZA	Zdefiniowane agenty z uwzględnieniem przynależności do kategorii (funkcjonalne, zarządzające, agenty wiedzy) oraz relacje (interakcje) między nimi

Źródło: opracowanie własne

W najniższej warstwie (wiedzy) poziomu zarządzania modelu szczegółowego **a2M** wykorzystana jest wiedza pozyskana na poziomie definiowania. Cechy i zachowania agentów zdefiniowane na pierwszym poziomie zostają wykorzystane do utworzenia relacji między agentami. W celu stworzenia poprawnej organizacji agentów następuje uporządkowanie wiedzy o wszystkich agentach. W ten sposób tworzony jest zbiór agentów z uwzględnieniem możliwości umieszczenia ich w odpowiedniej hierarchii. Wówczas na podstawie zgromadzonej wiedzy o cechach agentów można tworzyć relacje między nimi oraz określać przepływy komunikatów.

W warstwie przetwarzania mają miejsce procesy o charakterze organizacyjnym. Wyodrębnione zostają agenty wchodzące w skład Agentowej Struktury Zarządzania. Wyodrębnia się w ten sposób grupę agentów nadzorujących funkcjonowanie pozostałych agentów. Na tej podstawie przyporządkowuje się agentom zarządzającym zachowania związane z koordynacją przepływu zleceń (komunikatów) pomiędzy wszystkimi agentami wykorzystywanymi do realizacji projektu.

W następnym kroku uwzględnia się hierarchię agentów funkcyjnych poprzez wskazanie ich podległości względem agentów zarządzających. Podobnie uwzględniono w hierarchii agenty wiedzy. Należy zwrócić uwagę, że hierarchia nie dotyczy wyłącznie podległości agentów funkcjonalnych względem zarządzających. Agenty funkcjonalne tworzą także hierarchię funkcjonalną. Takie agenty jak agent dojrzałości zespołu dostawcy (który dokonuje analizy poziomu dojrzałości tego zespołu) mają podległe inne agenty — reprezentujące poszczególnych członków (agenty zadaniowe). Takie

podejście wynika z założenia, że w modelu każdy podmiot uczestniczący w projekcie powinien mieć reprezentację w postaci agenta. Rolą agenta dojrzałości zespołu nie jest kierowanie pracami (to leży w gestii agenta menedżera), ale monitorowanie bieżącego stanu zespołu. Należy zaznaczyć, że wraz z wyodrębnieniem zależności hierarchicznych następuje uporządkowanie przekazywania komunikatów między agentami. Każdy bowiem agent przyjmuje zlecenia i może je odbierać od agentów nadrzędnych względem niego.

Warto zaznaczyć, że zgodnie z założeniami podejścia adaptacyjnego struktura hierarchiczna może być modyfikowana w zależności od projektu. Prezentowana na rysunku 4.4. struktura jest strukturą opracowaną dla typów agentów reprezentujących najczęściej spotykane w projekcie elementy. W przypadku mniejszych projektów może nastąpić połączenie funkcji agenta koordynatora i wnioskującego, może również nastąpić wyłączenie innego, jeżeli menedżer uzna, że sam jest w stanie analizować wyniki działania innych agentów (np. analizy dojrzałości zespołu).

W warstwie funkcjonalności poziomu zarządzania modelem szczegółowego **a2M** zdefiniowano agenty (opisywane przez zestaw cech i zachowań), jak również relacje pomiędzy wszystkimi agentami w systemie. W celu osiągnięcia poprawnej funkcjonalności modelu niezbędne jest jeszcze uwzględnienie mechanizmu przesyłania zleceń między agentami. Mechanizm ten jest konsekwencją zależności hierarchicznych określanych przez zestawy komunikatów, na które mogą reagować poszczególne agenty. W opracowanym modelu przyjęto zasadę, że agent realizuje zadanie po odebraniu komunikatu od agenta wyższego szczebla w strukturze hierarchicznej (przykładowe komunikaty były prezentowane wcześniej). Dopuszcza się jednak możliwość przesyłania komunikatów między wszystkimi agentami, przy czym nie powinny być to komunikaty niosące w sobie zlecenie, a jedynie przekazujące określone informacje.

Opis podstawowych relacji pomiędzy agentami

Wcześniej wykazano zasadność tworzenia relacji hierarchicznej pomiędzy agentami ze względu na konieczność pokazania zależności pomiędzy poszczególnymi składnikami projektu. Relacja hierarchiczna jest odzwierciedleniem klasycznej struktury organizacyjnej, zatem jej zadaniem jest wprowadzenie poziomów nadrzędności i podrzędności zależnych od pełnionych funkcji (zakresu odpowiedzialności) oraz stopnia decyzyjności. Drugą ważną relacją, która jest efektem prac badawczych w II etapie konstruowania modelu, jest definiowanie interakcji między agentami.

Zauważono, że oprócz klasycznej podległości między agentami zachodzą różnego rodzaju interakcje. Niejednokrotnie od stopnia świadomości tych interakcji zależy powodzenie projektu. Przykładem na znaczenie tych interakcji może być sytuacja, w której lider zespołu ma znakomity kontakt z pozostałymi członkami. Łączą ich relacje otwarte, koleżeńskie, a tym samym przepływy informacji są poprawne. Może wtedy reagować na bieżąco na wszelkie zakłócenia (niepożądane zmiany) i w sposób kontrolowany kierować ewentualnymi konfliktami. Obserwacje wykazały, że w zespołach, w których interakcje między uczestnikami są jasne (zostało to opisane w części weryfikacyjnej) i partnerskie, prace projektowe realizowane są wydajniej.

Jeżeli tak jest, to znaczy że od poziomu tych relacji (interakcji) może zależeć istotnie powodzenie projektu i powinny one zostać zdefiniowane w trakcie tworzenia struktury projektowej. Nie tylko zatem relacje hierarchiczne, ale przede wszystkim te związane z interakcjami pomiędzy ludźmi powinny być przedmiotem analiz prowadzonych przez menedżera. Zarówno jeden jak i drugi typ relacji może zostać zamodelowany w systemach agentowych.

Zgodnie z powyższym przykładem wyodrębnienie interakcji staje się więc kluczowe z punktu widzenia sukcesu przedsięwzięcia. Jeżeli menedżer projektu będzie sobie zdawał sprawę, jakie interakcje zachodzą pomiędzy poszczególnymi uczestnikami projektu, tym łatwiej będzie podejmować decyzje np. dotyczące systemu motywowania czy ogólnej organizacji pracy.

Przyjmuje się zatem, że jeden z agentów agentowego systemu zarządzania powinien nadzorować stan tych relacji. Na tym etapie badań uznano, że za stan relacji pomiędzy agentami powinien

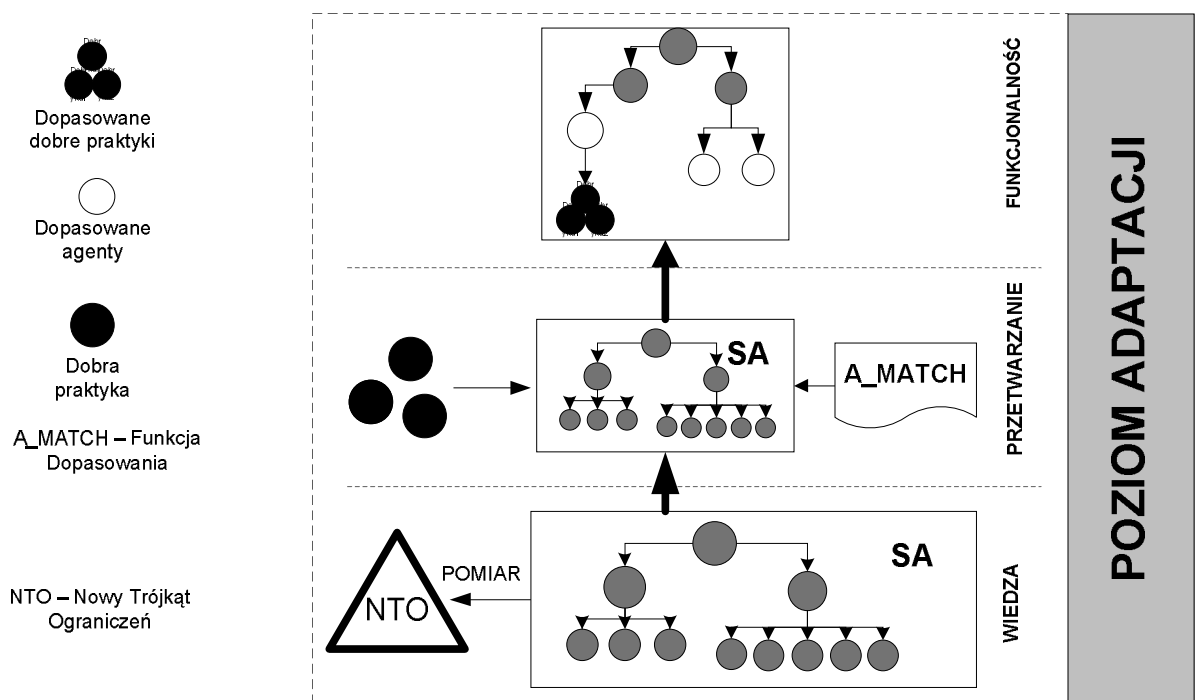
odpowiadać agent nadrzędny, czyli AGENT MENEDŻER. Autor nie był w stanie w opracowywanym modelu szczegółowym odwzorować relacji nieformalnych — koleżeństwa czy partnerstwa w projekcie. Przyjęto więc, że model szczegółowy **a2M** bazuje wyłącznie na relacjach hierarchicznych.

Konstrukcja modelu szczegółowego — poziom adaptacji (etap III)

Na poziomie adaptacji modelu szczegółowego **a2M** przeprowadza się proces dopasowania dobrych praktyk zarządzania projektami informatycznymi w zależności od wartości zmiennych decyzyjnych określonych jako nowy trójkąt ograniczeń projektowych.

Na poziomie adaptacji komponowana jest na potrzeby projektu właściwa struktura projektowa — efekt działania adaptacji odpowiednich dobrych praktyk. Na poziomie tym następuje wybór tylko tych dobrych praktyk, które są istotne z punktu widzenia nowego trójkąta ograniczeń. Odpowiednie agenty (menedżer, koordynator, agent wnioskujący) posiadają wiedzę o wartości zmiennych w nowym trójkącie ograniczeń. Dzięki takiemu podejściu znana jest entropia projektu, jak również poziomy dojrzałości zespołu dostawcy i klienta. W ramach nowych ograniczeń agenty gotowe są do podejmowania decyzji dotyczących zasobów (członkowie zespołu) i elementów wiedzy (dobre praktyki), które należy wykorzystać do realizacji projektu.

Można więc uznać, że na poziomie adaptacji modelu szczegółowego **a2M** następuje wykorzystanie odpowiednich mechanizmów przetwarzania (analizy) pozwalających na pełną adaptację tylko tych agentów (składników projektu), które są niezbędne do realizacji projektu.



Rys. 4.5 Poziom adaptacji modelu a2M

Źródło: opracowanie własne

Dla osiągnięcia przez zespół dostawcy poziomu dopasowania (adaptacji) należy zrealizować są dwa podstawowe procesy:

- Analizowanie — badanie stanu dojrzałości zespołu, dojrzałości klienta i poziomu entropii, czyli tworzenie NOWEGO TRÓJKĄTA OGRANICZEŃ.

- Budowanie struktury projektowej dopasowanej do tych nowych OGRANICZEŃ (następuje dopasowanie tylko tych agentów, które są istotne z punktu widzenia realizacji zadań w projekcie).

Tab. 4.6 Etap III Warstwy implementacyjne i oczekiwane produkty dla poziomu adaptacji modelu szczegółowego a2M

WARSTWA MODELU	REZULTAT/ZAWARTOŚĆ
FUNKCJONALNOŚĆ	Na tym poziomie pojawia się kompletna struktura projektu, czyli dopasowane dobre praktyki i niezbędne agenty do właściwej realizacji projektu, czyli elementy decyzji menedżerów.
PRZETWARZANIE	Algorytmy pozwalające na dopasowanie agentów i dobrych praktyk do ograniczeń (funkcja „agent-match”).
WIEDZA	Struktura hierarchiczna agentów oraz nowy trójkąt ograniczeń: - dojrzałość zespołu, - dojrzałość klienta, - entropia.

Źródło: opracowanie własne

W tabeli 4.5 przedstawiono warstwy implementacyjne i oczekiwane produkty dla poziomu adaptacji modelu szczegółowego a2M. Podstawą dla realizacji procesu adaptacji jest analiza nowego trójkąta ograniczeń. Stąd też w warstwie danych poziomu adaptacji należy dokonać pomiaru wartości zmiennych decyzyjnych — dojrzałości zespołu dostawcy, dojrzałości klienta oraz entropii projektu. Na podstawie cech agentów zdefiniowanych wcześniej agenty funkcyjne dokonują pomiaru tych trzech zmiennych decyzyjnych. Określenie nowego trójkąta ograniczeń jest podstawą przeprowadzenia działań adaptacyjnych. Agenty funkcyjne pozyskują wiedzę o procesach klienta, zespołu oraz cechach projektu w oparciu o dedykowane kwestionariusze. Na podstawie pozyskanych odpowiedzi przeprowadzane jest wnioskowanie (przez agenta wnioskującego lub agenta menedżera) dotyczące poziomu dojrzałości klienta, zespołu dostawcy oraz entropii. W zależności od tych wartości dopasowuje się sposób prowadzenia projektu (wybór dobrych praktyk zarządzania projektami) oraz adaptuje tylko te agenty, które są potrzebne do realizacji projektu.

W warstwie przetwarzania dla nowego trójkąta ograniczeń zostają adaptowane agenty zarządzające, które mogą podejmować podstawowe decyzje o sposobie prowadzenia prac projektowych. Takie podejście może mieć miejsce jedynie z wykorzystaniem odpowiedniego wnioskowania (procesu sprawdzenia, jakie dobre praktyki pasują do wartości zmiennych decyzyjnych). W tym celu agent wnioskujący powinien posłużyć się odpowiednim modelem ocenowym pozwalającym stwierdzić, które dobre praktyki pasują do znanych już ograniczeń w nowym trójkącie ograniczeń. Takie wnioskowanie można przeprowadzić przez zastosowanie w tej warstwie FUNKCJI ADAPTACJI — A_MATCH (od Agent match — dopasowanie). Funkcja adaptacji staje się zatem zbiorem reguł, faktów oraz mechanizmów umożliwiających właściwy dobór agentów. Przykład takiej funkcji dopasowania został przedstawiony w rozdziale II.

W warstwie funkcjonalności na podstawie trójkąta ograniczeń i działania funkcji dopasowania A_MATCH utworzona zostaje struktura projektowa właściwa dla danego projektu. Właściwą strukturę należy rozumieć jako zestaw poszczególnych składników projektu, czyli agentów. Na poziomie warstwy funkcjonalności ustalony zostaje skład zespołu projektowego, znane są cechy wszystkich uczestników zespołu projektowego, a tym samym możliwe jest przypisanie pozycji w zespole (LIDER, WYKONAWCA), jak i późniejsze przypisanie ról na podstawie zbadanych kompetencji (ANALITYK, TESTER). W warstwie tej uwzględnione są tylko te agenty (te osoby), których uczestnictwo w projekcie jest niezbędne z punktu widzenia celu projektu.

4.4 Podsumowanie

W niniejszym rozdziale zaprezentowano główne funkcjonalności adaptacyjnego agentowego modelu zarządzania projektami informatycznymi. Funkcjonalności te, zdefiniowane jako poszczególne agenty modelu, mają na celu sprostać problemowi dopasowania metod i/lub dobrych praktyk zarządzania projektami do nowych ograniczeń projektowych zdefiniowanych w rozdziale II. W niniejszym rozdziale zaprezentowano również cechy opracowanego adaptacyjnego agentowego modelu zarządzania projektem informatycznym z podziałem tych cech na poszczególne poziomy modelu. Tak skonstruowany model ma na celu zapewnić efektywną analizę stanu poprzez umożliwienie pomiaru wartości zmiennych decyzyjnych (dojrzałości klienta, dojrzałości zespołu, dojrzałości organizacji) oraz właściwe wnioskowanie w oparciu o wartości tych zmiennych, czyli dopasowywanie odpowiednich metod i/lub dobrych praktyk zarządzania projektem. Podczas omawiania modeli: ogólnego i szczegółowego **a2M** zauważono również, że analiza stanu związana z dojrzałością zespołu daje menedżerom dodatkowe informacje na temat członków zespołu realizującego projekt informatyczny. Zwrócono także uwagę na to, że zarówno model ogólny jak i szczegółowy **a2M** zapewnia możliwość kontroli zespołu, dopasowania właściwego systemu motywacyjnego czy nawet dopasowywania zadań do predyspozycji członków zespołu, przez co dopasowanie zespołu do projektu jest bardziej efektywne. Kolejnym ważnym spostrzeżeniem płynącym z wykorzystania modeli ogólnego i szczegółowego **a2M** jest jego organizacyjno-implémentacyjny charakter. Dzięki takiemu podejściu mierzalne zagregowane zmienne decyzyjne modelu (np. pomiaru dojrzałości klienta czy dojrzałości zespołu) mogą być określone (mieć przyporządkowane wartości). Stąd też na poziomach i warstwach modeli szczegółowego i ogólnego **a2M** opisano procedury postępowania (np. jak mierzyć dojrzałość zespołu). Jednocześnie procedury te mogą zostać zaimplementowane w dowolnym środowisku agentowym i stanowić system wsparcia decyzyjnego.

W celu sprawdzenia, jak zaprezentowane w niniejszym rozdziale modele szczegółowy i ogólny **a2M** zachowują się w rzeczywistym środowisku projektowym, poddano model weryfikacji (badanie zasadności replikatywnej i predykcyjnej) na przykładzie 8 projektów. Weryfikacja ta została omówiona w rozdziale V.

Rozdział V

Badania zasadności replikatywnej modelu

Zgodnie z przyjętymi założeniami (rozdział III) procesowi weryfikacji został poddany model szczegółowy **a2M** (opisany w rozdziale IV), którego konstrukcja zawiera trzy poziomy — poziom definiowania, poziom zarządzania oraz poziom adaptacji. Każdy z poziomów tego modelu zawiera trzy warstwy — warstwę zasobów wiedzy, warstwę przetwarzania oraz warstwę funkcjonalności. Przyjęto w badaniu zasadności replikatywnej, że proces badania dotyczyć będzie identyfikacji poszczególnych warstw i poziomów w zrealizowanych projektach informatycznych (autor pracy brał udział w tych projektach). Celem tych badań jest wykazanie także istnienia procesów adaptacji struktur organizacyjno-implementationary potwierdzających adaptacyjny charakter modelu szczegółowego **a2M**. Głównym jednak celem tego rozdziału jest prezentacja procesu weryfikacji hipotezy badawczej postawionej na początku tej pracy — wykazania, że opracowany model szczegółowy **a2M** usprawnia procesy zarządzania projektami (tworzy adaptacyjne organizacyjno-implementationary struktury). Zaprezentowane w kolejnym rozdziale badanie zasadności predykcyjnej ma na celu wykazanie, że na podstawie istniejących w projektach adaptacyjnych organizacyjno-implementationary struktur (rozdział V) można tworzyć scenariusze prognostyczne realizacji projektów.

5.1 Procesy badania zasadności replikatywnej modelu

Badania przeprowadzono w oparciu o przygotowane wstępnie plany badań. Badania obejmowały analizę zakończonych projektów informatycznych przeprowadzoną na podstawie dokumentacji projektowej, rozmów z członkami zespołów projektowych oraz ze specjalistami z firm informatycznych uczestniczących wspólnie z autorem tej pracy w zakończonych projektach. Celem tych analiz było wykazanie, na ile możliwa jest identyfikacja struktur modelu szczegółowego **a2M** i na ile te struktury mają adaptacyjny organizacyjno-implementationary charakter. W każdym badaniu określono: cel badania projektu, założenia dla przeprowadzenia badań oraz środowisko badań. Wynikiem badań była identyfikacja warstw i poziomów modelu szczegółowego **a2M**.

Środowiskiem badań zasadności replikatywnej były:

- rzeczywiste projekty informatyczne dotyczące wytwarzania oprogramowania, w których uczestniczył autor pracy pomiędzy 2006 a 2011 rokiem,
- zespoły projektowe złożone ze studentów specjalności Zarządzanie Technologiami Informatycznymi na Politechnice Gdańskiej; w ramach przedmiotów dydaktycznych realizowanych ze studentami powoływano zespoły projektowe i analizowano pracę tych zespołów w ramach zleconych im zadań,
- projekty informatyczne wykonywane na potrzeby Uniwersyteckiego Centrum Kompetencyjnego Technologii Rational powstałego na Politechnice Gdańskiej przy współpracy z firmą IBM; projekty realizowane w ramach Centrum Kompetencyjnego miały zazwyczaj charakter badawczo-rozwojowy; nie wiązały się więc z wytworzeniem oprogramowania, ale jego rozwojem i integracją,
- dedykowane środowisko przygotowane na potrzeby jednego z eksperymentów; podczas konferencji poświęconej projektom informatycznym (ITAKM 2011, Jurata) zaproszono przedstawicieli firm informatycznych oraz ekspertów dziedzinowych, a następnie utworzono z nich zespoły i analizowano realizację zadań; eksperyment przebiegał w specjalnie przygotowanym pomieszczeniu dla pracy grupowej, a zadania dotyczyły planowania projektów informatycznych.

W tabeli 5.1 zebrano ogólne dane dotyczące środowiska badań. Zidentyfikowano tytuł badania (nazwę projektu) oraz jego cel. Przyjęty plan posłużył do przeprowadzenia dalszych badań oraz eksperymentów. W trakcie prowadzonych badań niektóre plany ulegały modyfikacjom, dlatego przy omawianiu poszczególnych badań zostały one poprzedzone stosowną prezentacją środowiska badań. W ramach zaplanowanych badań zidentyfikowano poziomy i warstwy modelu szczegółowego oraz wykazywano adaptacyjny organizacyjno-implementacyjny charakter modelu i jego przydatność dla procesów zarządzania projektem.

Tabela 5.1 Plan badania zasadności replikatywnej – ogólny opis środowiska badań

POZIOM MODELU	WARSTWA	PLANOWANE BADANIE	CEL
definiowalny	WIEDZA	PROJEKT: PSE — wdrożenie pilotażowe	Definiowanie struktury projektowej przy dużej jej złożoności.
	PRZETWARZANIE	PROJEKT: „kreator innowacyjności” — wytwarzanie systemu	Wyodrębnienie agentów i określenie ich cech (atrybutów).
	FUNKCJONALNOŚĆ	PROJEKT: „kreator innowacyjności” — wdrożenie	Funkcjonalności poszczególnych agentów.
zarządzalny	WIEDZA	PROJEKT: testy kompetencji	Określenie parametrów dla relacji między agentami.
	PRZETWARZANIE	PROJEKT: PSE, MNiSW	Zdefiniowanie struktury opartej na relacjach i zależnościach pomiędzy agentami, badanie przepływu informacji i zadań.
	FUNKCJONALNOŚĆ	PROJEKT: MNiSW	Sprawdzenie współpracy agentów przy dużej dojrzałości klienta i dostawcy.
optymalizacji	WIEDZA	PROJEKT: projekt Program Operacyjny 5.1 (przy współpracy z HP)	Określenie możliwości definiowania trójkąta ograniczeń: dojrzałość zespołu, klienta, entropia.
	PRZETWARZANIE	PROJEKT: projekty dla Programu Operacyjnego i Kapitał Ludzki (IBM, Bank BPH)	Algorytmy pozwalające na dopasowanie agentów i dobrych praktyk do ograniczeń — weryfikacja instrumentów pomiarowych i zasadności pomiaru – funkcja A_MATCH (algorytm dopasowania).
	FUNKCJONALNOŚĆ	PROJEKT: projekty dla Programu Operacyjnego i Kapitał Ludzki (IBM, HP, Bank BPH)	Dopasowanie zespołów i dobrych praktyk. Obserwacja postępów realizacji projektów.

Źródło: opracowanie własne

5.1.1 Badanie 1-R. Zasadności definiowania struktur projektowych

Badanie zostało przeprowadzone w ramach projektu pilotażowego wdrożenia systemu Rational Team Concert (RTC) w firmie PSE INFO. W realizację projektu zaangażowany był zespół powołany w Zakładzie Zarządzania Technologiami Informatycznymi Politechniki Gdańskiej na zlecenie dostawcy oprogramowania RTC — firmy IBM. Projekt realizowany był od stycznia 2009 r. do końca lipca 2009 r. Założenia do badania 1-R przedstawiono w tabeli 5.2

Tabela 5.2 Założenia do badania 1-R

		Kod badania 1-R	
Cel	Celem badania jest weryfikacja zasadności definiowania struktur projektowych przez kierowników projektów.		
Środowisko badania	Projekt wdrożenia oprogramowania do zarządzania projektami dla klienta z krajowego sektora energetycznego.		
Poziomy model a2M do identyfikacji	Do potwierdzenia – poziom definiowania inne poziomy?		Procesy adaptacji?
Warstwa modelu a2M do identyfikacji	Do potwierdzenia – warstwa wiedzy inne warstwy?		Procesy adaptacji?
Początek realizacji	Marzec 2009	Zakończono	
Termin zakończenia	Wrzesień 2009	Zakończono	

Źródło: opracowanie własne

Opis przebiegu badania

Przeprowadzone badanie miało na celu wykazanie, jak złożona może być struktura projektowa poprzez wykazanie istnienia możliwych elementów tej struktury. Oznacza to zatem, że weryfikując warstwę wiedzy na poziomie definiowalnym modelu szczegółowego **a2M** należy sprawdzić możliwość wyodrębnienia podstawowych elementów struktury, które następnie należy przekształcić do postaci agentów opisywanych cechami oraz zachowaniami.

Klient omawianego projektu, w ramach którego przeprowadzano eksperyment, oczekiwał rozwiązania usprawniającego przepływu pracy (tzw. *workflow*) podczas realizacji przez niego różnych projektów (również informatycznych) na własne potrzeby. Oczekiwania klienta ukierunkowane były przede wszystkim na zastosowanie takiej technologii, dzięki której podniesie się komfort pracy zespołowej. Oczekiwano również usprawnienia komunikacji w zespole oraz usprawnienia przepływu dokumentów.

W związku z problemami organizacyjnymi klient omawianego projektu zwrócił się z prośbą o wdrożenie produktu do dostawcy oprogramowania (firmy IBM). W celu usprawnienia procesów w organizacji klienta zaproponowano wdrożenie oprogramowania Rational Team Concert oraz zmodyfikowanie tego narzędzia do potrzeb klienta. Jednocześnie dostawca oprogramowania zwrócił się do jednostki naukowo-badawczej, jaką jest Zakład Zarządzania Technologiami Informatycznymi, z propozycją przeprowadzenia pilotażowego wdrożenia oprogramowania RTC. W Zakładzie prowadzono prace badawcze nad tymi technologiami i dysponowano specjalistami. Stąd też jednostka naukowo-badawcza reprezentowała dostawcę oprogramowania.

Na podstawie podanego opisu stwierdzono obecność podstawowych elementów struktury projektowej, które występują w rzeczywistych projektach:

- organizacja klienta, która ma problemem z wdrożeniem systemu informatycznego,
- dostawca oprogramowania, która udostępni swój produkt,
- jednostka naukowo-badawcza, która może wesprzeć realizację projektu informatycznego swoimi zasobami (pracownicy i studenci).

Kolejnym ustaleniem w ramach omawianego projektu było powołanie organu mającego nadzorować przebieg projektu. W skład tego organu wchodziły trzy osoby — po jednej z każdej organizacji zainteresowanej realizacją omawianego projektu. Przy czym ustalono, że prace związane

z wdrożeniem zostaną powierzone młodemu zespołowi złożonemu ze studentów, którzy zrealizują zadania w ramach praktyk ukierunkowanych na zdobycie doświadczenia projektowego. Jednocześnie uznano, że należy wyłonić osobę zarządzającą projektem, która będzie organizować pracę zespołowi studentów. Z uwagi na rozwojowy charakter projektu oraz konieczność integracji nowo wdrażanego systemu z istniejącymi już w organizacji klienta, powołano 3 zespoły projektowe.

Badanie miało na celu wyłonienie wszystkich elementów struktury projektowej, które zostaną wyodrębnione podczas ustaleń pomiędzy klientem a dostawcą. Jak wynika z podanego opisu, projekt realizowany był w oparciu o strukturę złożoną z organu zarządzającego projektem (analogia do agentowego systemu zarządzania).

Należy dodać, że w trakcie realizacji projektu struktura projektowa została poddana modyfikacjom w wyniku kolejnych ustaleń między dostawcą a klientem. Powołano kierownika podległego organowi nadzorującemu i powierzono mu kierowanie pracami trzech kilkusobowych zespołów w ramach tego projektu. Stwierdzono istnienie złożonej struktury projektowej składającej się z podmiotów powiązanych relacjami hierarchicznymi oraz potwierdzono, że kierownik projektu uwzględniał potrzebę takich struktur w zarządzanym przez siebie projekcie informatycznym.

Po zdefiniowaniu podstawowych ról projektowych w ramach zespołów kierownik wyznaczył również liderów poszczególnych grup, których zadaniem było monitorowanie postępów i kierowanie pracami powierzonych im zespołów. Oprócz organu nadzoru, kierownika i zespołów pojawili się także liderzy prowadzący pomniejszych grupy projektowe. Okazało się również, że przedstawiciel klienta wchodzący w skład organu nadzorującego prace nie jest w stanie uczestniczyć w pracach projektowych i wydelegował do tego celu dwóch pracowników działu informatycznego. Stąd też pojawił się w projekcie zarówno przedstawiciel klienta, jak i przedstawiciel użytkownika końcowego.

Uznano również, że w celu lepszej organizacji zadań związanych z wdrożeniem nowego oprogramowania w firmie klienta, dostawca (producent) oprogramowania zapewni wsparcie techniczne, powołując ze swojej strony dwóch mentorów, którym powierzono wspieranie zespołów projektowych podczas wdrażania wspomnianego oprogramowania. W ten sposób struktura projektowa powiększyła się o kolejne osoby, które bezpośrednio nie wchodziły w skład zespołu, ale stały się podmiotami zainteresowanymi przebiegiem projektu. Mentorzy powołani przez firmę IBM stanowili więc ekspertów dziedzinowych w rozumieniu modelu agentowego. Zgodnie z założeniem, że struktury projektów są coraz bardziej rozbudowane, a kierownicy powinni je dokładnie analizować przed dobraniem dobrych praktyk zarządzania projektami informatycznymi, przystąpiono do definiowania tej struktury. Kierownik projektu przygotował odpowiednie dokumenty pokazujące rolę każdego uczestnika w realizowanym projekcie.

Okazało się także, że klient nie do końca jest w stanie określić sposób funkcjonowania swojej organizacji za pomocą oferowanych na potrzeby zarządzania technologii. Stąd też jeden z trzech zespołów projektowych przygotował warianty wykorzystania technologii, które zaprezentował klientowi w celu wybrania najlepszego. Przygotowanie wariantów spowodowało jednak to, że pojawiło się zapotrzebowanie na odpowiednie modele ocenowe tych technologii, które posłużyły do wartościowania przedstawianych klientowi rozwiązań. Jednocześnie podczas prezentacji wariantów rozwiązań dla klienta poproszono o wsparcie pracowników Zakładu w celu weryfikacji proponowanych przez zespoły wariantów. W ten sposób struktura omawianego w eksperymencie projektu rozszerzyła się o kolejne nowe osoby. Zauważono także, że zwykła struktura projektowa składająca się z klienta, zespołu i kierownika nie jest wcale taka oczywista i może składać się z wielu innych powiązanych elementów.

Wnioski

Tak przeprowadzone badanie wykazało znaczenie struktury projektowej dla planowania i realizacji późniejszych zadań w projekcie. Pozwoliło na identyfikację poziomu definiowania modelu szczegółowego **a2M** oraz warstwy wiedzy. Wskazało także na tworzenie struktur hierarchicznych

typowych dla poziomu zarządzania. Wskazało także na mechanizmy tworzenia poziomu zarządzania modelem szczegółowego **a2M**, który w analizowanym projekcie ma charakter iteracyjny. Nie stwierdzono istnienia w realizowanym projekcie poziomu adaptacji. Nie stwierdzono także istnienia procesów adaptacyjnych. W tabeli 5.3 zebrano wyniki badania 2-R

Tabela 5.3 Wnioski z badania 1-R

		Kod badania: 1-R
Realizacja celu badania	Cel badania osiągnięty	
Zidentyfikowane poziomy modelu szczegółowego a2M	Definiowania Zarządzania	Nie istnieją procesy adaptacji
Zidentyfikowane warstwy modelu szczegółowego a2M	Wiedzy Przetwarzania	Nie istnieją procesy adaptacji

Źródło: opracowanie własne

Zgodnie w powyższych wnioskami zastanawiano się, jak przeprowadzone badanie wpływa na strukturę modelu szczegółowego **a2M**. Niewątpliwie potwierdzono, że w najniższej warstwie modelu wspierającego zarządzanie projektami należy definiować wszystkie składniki, co do których kierownik projektu może podejmować późniejsze decyzje. Pozostaje jednak pytanie, jak kierownik może tę strukturę definiować oraz jakie zastosować działania, aby zapewnić właściwą współpracę w ramach tak złożonych struktur. Przeprowadzone badanie potwierdziło słuszność definiowania struktury projektowej, ale wykazało, że niektóre elementy tej struktury (jak mentorzy — eksperci) pojawiają się już po rozpoczęciu projektu.

Przeprowadzone badanie wykazało, że należy nie tylko analizować i definiować struktury w projektach informatycznych, lecz jeszcze należy uwzględniać ich dynamikę. Struktura bowiem może się zmieniać, ulegać modyfikacjom w czasie i zmieniać swój poziom złożoności. Badanie wykazało także jak złożona może być taka struktura projektowa i jakie interakcje mogą zachodzić pomiędzy tymi elementami. Mogą być one odwzorowane poprzez relacje hierarchiczne (jak opisana podległość kierownika projektu wobec organu nadzoru czy podległość liderów względem kierownika), ale również jako relacje oparte na przepływie wiedzy, jak relacja pomiędzy mentorami a członkami zespołów. Stąd też zasadne wydaje się, aby model szczegółowy **a2M** uwzględniał również relacje, co założono na poziomie zarządzanym modelem i co stanowić będzie podstawę weryfikacji w kolejnych badaniach.

Ważnym wnioskiem płynącym z przeprowadzonego eksperymentu jest również możliwość identyfikacji elementów struktury projektowej, które są w stanie dostarczać merytoryczną wiedzę osobom uczestniczącym. Ten wniosek jest istotny z dwóch powodów. Po pierwsze stanowi wskazówkę dla kierownika projektu, aby uwzględnić możliwość korzystania z wiedzy i doświadczenia mentorów (zapotrzebowanie na AGENTA EKSPERTA). Po drugie obecność takich osób w projekcie świadczy o tym, że struktury projektowe mają charakter dynamiczny, czyli mogą zmieniać się w czasie. Potwierdza to potrzebę włączenia mentorów w omawianym badaniu.

Wnioskiem, który także wynika z przeprowadzonego badania, jest fakt, że w projekcie informatycznym może nastąpić zapotrzebowanie nie tylko na osoby dostarczające wiedzę, ale również na inne źródła wiedzy oraz technologie informatyczne wspomagające realizację zadań w projekcie.

Takim źródłem wiedzy wykorzystanym w badaniu były np. modele ocenowe pozwalające na wartościowanie wariantów rozwiązań przedstawianych klientowi. Takim źródłem wiedzy może też być zestaw dobrych praktyk zarządzania projektami zawarty w formalnych metodach zarządzania projektami.

5.1.2 Badanie 2-R. Ocena możliwości identyfikacji elementów struktury projektowej jako agentów projektu

Celem drugiego badania było sprawdzenie możliwości implementacji elementów struktury projektowej na poziom agentów. Aby to osiągnąć, konieczne stało traktowanie uczestników jako niezależnych jednostek reprezentowanych za pomocą cech i zachowań. Do przeprowadzenia badania wybrano ocenę jednego etapu projektu programu „Kreator innowacyjności — wsparcie innowacyjnej przedsiębiorczości akademickiej”. Dostawcą (wytwórcą) oprogramowania była firma ZETO Olsztyn. Koordynatorem projektu z ramienia Politechniki Gdańskiej — Zakład Zarządzania Technologiami Informatycznymi. Projekt finansowany był ze środków Ministerstwa Nauki i Szkolnictwa Wyższego. Projekt realizowano na przestrzeni jednego roku, od stycznia do listopada 2010 r. Do eksperymentu wybrano etap implementacji, w tym projekcie przeprowadzony na przełomie marca i kwietnia 2010 r. Założenia do badania 2-R przedstawiono w tabeli 5.4.

Tabela 5.4 Założenia do badania 2-R

		Kod badania	2-R
Cel	Celem eksperymentu jest wykazanie możliwości definiowania pojedynczych elementów struktury projektowej, jako niezależne agenty podejmujące współpracę w projekcie.		
Środowisko badania	Środowiskiem weryfikacji jest projekt informatyczny polegający na wytworzeniu platformy wspomagającej integrację uczelni, biznesu oraz studentów w celu komercjalizacji badań naukowych. Głównym produktem projektu była aplikacja bazodanowa umożliwiająca zamieszczanie ofert zapotrzebowania na określone technologie przez pracodawców oraz ofert naukowców chcących komercjalizować swoje badania, zgodnie z założeniami programu „Innowacyjna Gospodarka”.		
Poziomy model a2M do identyfikacji	Do potwierdzenia – poziom definiowania Inne poziomy?	Procesy adaptacji?	
Warstwa modelu a2M do identyfikacji	Do potwierdzenia – warstwa przetwarzania Do potwierdzenia – warstwa funkcjonalności Inne warstwy?	Procesy adaptacji?	
Początek realizacji	Marzec 2010	Ukończono	
Termin zakończenia	Kwiecień 2010	Ukończono	

Źródło: opracowanie własne

Opis przebiegu eksperymentu

Drugie badanie prowadzące do weryfikacji modelu szczegółowego **a2M** wspomagającego kierowników projektów w zarządzaniu projektami informatycznymi miało na celu wykazanie możliwości prezentowania elementów struktur projektów informatycznych (czyli członków zespołów, ekspertów, ale również zasobów wiedzy jak modele ocenowe) w postaci agentów. W omawianym projekcie, który stanowił środowisko dla badania 2-R, zidentyfikowano dużą dynamikę struktury projektowej. W projekcie uwzględniono bowiem możliwość włączenia na niektórych etapach np.

ekspertów (zwanych w projekcie specjalistami). Stwierdzono również, że z racji rozproszonego miejsca działania przedstawicieli koordynatora (Gdańsk) oraz dostawcy (Olsztyn) jest to sprzyjające środowisko do oceny transformacji elementów struktury projektu do postaci agentów. W związku z tym zdecydowano się sprawdzić, na ile koncepcja wykorzystania agentów w zarządzaniu projektem jest możliwa w odniesieniu do rzeczywistego projektu. Postanowiono więc, aby ocenić zarządzanie tym projektem i traktować wszystkich jego uczestników jako agentów. Jest to zgodne z przyjętą definicją, w myśl której agentem określa się każdy byt (fragment rzeczywistości), który potrafi podejmować działanie (aktywność) zgodnie z przypisanymi mu atrybutami w imieniu zlecającego, określanego mianem użytkownika. Przeniesienie koncepcji agentów na płaszczyznę zarządzania oznacza, że możliwe jest potraktowanie elementów struktury projektowej jako takie niezależne byty współpracujące w określonych momentach (czyli reagujące na określone sygnały, zwane również w podejściu agentowym komunikatami albo bodźcami).

Zgodnie więc z tą koncepcją, planując strukturę projektową omawianego eksperymentu, uznano, aby potraktować wszystkich uczestników jako agenty, przydzielając każdemu zakres uprawnień, w ramach których powinien się poruszać (np. lider zespołu realizującego projekt miał za zadanie komunikować się wyłącznie z kierownikiem projektu i wykonywać zadania przez niego zlecone tylko w określonych odstępach czasu). W podobny sposób potraktowano pozostałych uczestników projektu. Okazało się bowiem, że na pewnym etapie niezbędny jest ekspert dziedzinowy, który będzie w stanie ocenić, na ile rozwiązanie oferowane przez dostawcę jest zgodne z oczekiwaniami klienta. Ekspert miał być osobą niezależną, która dokona oceny w odpowiednim (wskazanym przez kierownika projektu) momencie. Powołanie eksperta mieści się w koncepcji podejścia agentowego, który został włączony do struktury projektu informatycznego przez kierownika, ale nie podejmował żadnych działań do momentu nie otrzymania właściwego zlecenia (komunikatu). Dopiero po otrzymaniu polecenia od kierownika projektu ekspert podejmował działanie.

Podobne działania realizowano w ramach całego zespołu wytwórczego. W trakcie etapu implementacyjnego omawianego projektu uznano, że główny programista będzie reagował na określone komunikaty (zlecenia) przesłane przez kierownika projektu. Sygnałem (komunikatem) do implementacji określonego modułu było więc zlecenie od kierownika projektu kierowane do głównego programisty, który pełnił funkcję lidera zespołu wytwórczego. W ten sposób komunikat pomiędzy dwoma uczestnikami projektu stanowił sygnał do podejmowania działań przez zespół wytwórczy (AGENCY ZADANIOWE). Takie podejście również wpasowuje się w koncepcję agentowego zarządzania projektami i wykazuje, że podejście agentowe ma swoje odbicie w rzeczywistych projektach informatycznych.

Kolejnym aspektem uwzględnionym w omawianym casie badawczym jest fakt, że dostawca wytwarzanego oprogramowania pochodził z innego ośrodka niż klient. Dodatkowo także zespół wytwórczy pracował w warunkach rozproszonych (programiści przebywali w różnych miejscach podczas pracy). Oznacza to, że uczestnicy projektu podejmowali działania (aktywność) tylko w sytuacji, kiedy otrzymywali wyraźny sygnał (komunikat) do działania. O ile kierownik projektu zlecał zadanie liderowi, o tyle później to lider „uruchamiał” poszczególnych wykonawców poprzez odpowiednie zlecenia (czyli komunikaty w myśl podejścia agentowego). Takie podejście okazało się skuteczne, gdyż prace były realizowane terminowo. Zminimalizowany kontakt bezpośredni nie okazał się w tym wypadku przeszkodą.

Innym ważnym zjawiskiem zaobserwowanym w prowadzonym badaniu był fakt, że struktura projektowa, podobnie jak w przypadku eksperymentu 1-R, również była dość złożona. Oprócz kierownika i zespołu wytwórczego występował jeszcze dodatkowo lider zespołu wytwórczego. Ponadto bardzo precyzyjnie były zdefiniowane role klienta. Dodatkowo zdefiniowano rolę eksperta (w projekcie nazywanego specjalistą) oraz rolę wsparcia technicznego. Każda z tych ról była odzwierciedleniem określonego agenta i tak ustalono zasady, aby każda z ról reagowała na określone komunikaty przekazywane przez określone role (agenty). Dzięki temu w strukturze omawianego

projektu sprawdzono doświadczalnie, czy podejście agentowe może być odwzorowane w rzeczywistym projekcie.

Jedyny problem, jaki zaobserwowano w projekcie, dotyczył obszaru przygotowania dokumentacji projektowej. Okazało się bowiem, że nie przypisano odpowiedzialności za przygotowanie dokumentów prezentujących stworzony produkt. Na tej podstawie powstał spór pomiędzy liderem zespołu a kierownikiem projektu, który z nich jest odpowiedzialny za przygotowanie dokumentów.

Wnioski

Przeprowadzone badanie pozwoliło na identyfikację elementów struktury projektowej jako agentów projektu. Umożliwiło identyfikację poziomów definiowania i zarządzania modelem szczegółowego **a2M** oraz warstw wiedzy, przetwarzania i funkcjonalności. Wskazało także na mechanizmy tworzenia struktur agentowych zarówno na poziomie definiowania, jak i zarządzania modelem szczegółowego **a2M**. Nie stwierdzono istnienia w realizowanym projekcie poziomu adaptacji. Nie stwierdzono także istnienia procesów adaptacyjnych. Wnioski z badania 1-R przedstawiono w tabeli 5.5.

Tabela 5.5 Wnioski z badania 2-R

		Kod badania: 2-R
Realizacja celu badania	Cel badania osiągnięty	
Zidentyfikowane poziomy model szczegółowego a2M	Definiowania Zarządzania	Nie istnieją procesy adaptacji
Zidentyfikowane warstwy modelu szczegółowego a2M	Wiedzy Przetwarzania Funkcjonalności	Nie istnieją procesy adaptacji

Źródło: opracowanie własne

Na podstawie przedstawionego eksperymentu zaobserwowano także, że podejście agentowe nie tylko jest możliwe do zastosowania w rzeczywistym projekcie, ale że w określonych sytuacjach (jak rozproszona struktura projektowa, różne miejsca przebywania) jest niemal konieczne. Zauważono również, że odzwierciedlenie w projekcie (na podstawie przeprowadzonego badania) zasad typowych dla systemów agentowych opartych na komunikatach niosących ze sobą zlecenie wykonania określonego zadania jest nie tylko możliwe do wykonania, ale sprawdza się bardzo porządkując realizację zadań.

Kolejnym wnioskiem płynącym z przeprowadzonego eksperymentu jest fakt, że role projektowe mogą być tak skonstruowane, aby odzwierciedlać pewne zachowania typowe dla agentów. Okazało się bowiem, że zarówno kierownik projektu jako agent, jak również wszyscy wykonawcy mogą reagować na określone komunikaty w sposób poprawny i wykonywać powierzone zadania. Pozwoliło to zatem potwierdzić założenia stawiane dla analizowanej warstwy modelu szczegółowego **a2M**, że możliwe jest przełożenie rzeczywistych struktur projektowych na struktury agentowe.

Ważnym wnioskiem płynącym z przeprowadzonego eksperymentu jest również to, że podczas definiowania agentów najważniejszym zadaniem jest przypisanie im odpowiednich uprawnień, czyli cech oraz możliwości działania, czyli zachowań. Kompetencje te dotyczą podległości agentów względem siebie (np. lider zespołu podlega kierownikowi projektu). Kompetencje te mogą jednak dotyczyć także zakresu wykonywanych zadań. Ważnym zatem wnioskiem płynącym

z przeprowadzonych badań jest to, aby stosując agentowy model zarządzania projektem informatycznym umiejętnie odwzorować role projektowe do postaci agentów.

Kolejnym wnioskiem wypływającym z przeprowadzonego eksperymentu jest sprawdzenie, że przypisanie cech agentów może dotyczyć zarówno poziomu zarządzania (agent kierujący projektem, agent reprezentujący klienta), jak i poziomu wytwarzania (gdzie mianem agentów określono także członków zespołu wytwórczego).

Wyciągnięte wnioski z przeprowadzonego badania stanowią również punkt wyjścia do modyfikacji modelu szczegółowego **a2M**. Zaobserwowano, że dla poprawnego funkcjonowania wszystkich członków zespołu projektu zgodnie z podejściem agentowym należy bardzo precyzyjnie określić ich możliwe zachowania i wzajemne powiązania. Należy zatem w modelu szczegółowym **a2M** uwzględnić także sposób komunikowania się agentów między sobą, jak również relacje między agentami. Budowanie relacji pomiędzy agentami w projekcie informatycznym jest możliwe tylko w przypadku agentów niezbędnych dla realizacji projektu informatycznego. Należy więc uwzględnić w warstwie zarządzania modelu pełne definiowanie agentów poprzez przypisanie im cech oraz zachowań (czyli możliwych działań w ramach projektu).

5.1.3 Badanie 3-R. Weryfikacja warstwy funkcjonalności modelu na poziomie definiowania

Trzecie z badań zasadności replikatywnej modelu szczegółowego **a2M** dotyczy weryfikacji warstwy funkcjonalności modelu szczegółowego **a2M** na poziomie definiowania. W warstwie tej powstaje pełna definicja elementów struktury projektowej do postaci agentów, każdy agent posiada przypisany zestaw cech i zachowań. Środowiskiem weryfikacji był etap projektu związanego z implementacją systemu opartego na sztucznej inteligencji na potrzeby grantu „Badanie technologii informatycznych z wykorzystaniem systemów inteligentnych”. Projekt realizowany był na przestrzeni trzech lat, od stycznia 2008 do grudnia 2010 r. Dostawcą oprogramowania była firma NETHOS sp. z o.o., klientem — Zakład Zarządzania Technologiami Informatycznymi Politechniki Gdańskiej pełniący funkcję koordynatora przyznanego przez Ministerstwo Nauki i Szkolnictwa Wyższego grantu. W tabeli 5.6 przedstawiono założenia do przeprowadzania badania.

Tabela 5.6 Założenia do badania 3-R

		Kod badania	3-R
Cel	Wykazanie zasadności definiowania cech i zachowań agentów reprezentujących poszczególne elementy struktury projektowej.		
Środowisko badania	Projekt informatyczny polegający na wytworzeniu decyzyjnego systemu wsparcia dla kierowników projektów oparty o regułowe bazy wiedzy.		
Poziomy modelu a2M do identyfikacji	do potwierdzenia – poziom definiowania inne poziomy?	Procesy adaptacji?	
Warstwa modelu a2M do identyfikacji	do potwierdzenia – warstwa funkcjonalności inne warstwy?	Procesy adaptacji?	
Początek realizacji	Lipiec 2010	ukończono	
Termin zakończenia	Wrzesień 2010	ukończono	

Źródło: opracowanie własne

Opis przebiegu eksperymentu

Celem badania było wykazanie zasadności pełnego definiowania wszystkich agentów wchodzących w skład struktury projektowej. Na podstawie poprzednich badań stwierdzono, że podejście agentowe jest możliwe do zastosowania w rzeczywistych projektach. W niniejszym badaniu postanowiono wykazać, że dla poprawnego funkcjonowania agentów konieczne jest wyraźne określenie ich atrybutów, rozumianych jako zestaw cech i zachowań.

Agent programowy, zgodnie z definicją, może posiadać swoje cechy decydujące o jego przydatności z punktu widzenia celu, w jakim agenta stworzono.[90,91] Cechy agenta przeniesione do projektu związane są z jego rolą, jaką pełni w strukturze projektu. Zachowania natomiast oznaczają, jakie akcje może podejmować agent w zależności od otrzymanego zlecenia (komunikatu). Rodzi się więc pytanie, czy traktując elementy struktury projektu informatycznego jako agenty, mogą one również być reprezentowane przez określone zestawy cech i zachowań. Przyjęto bowiem w modelu szczegółowym **a2M**, że poziom definiowania tego modelu umożliwia zdefiniowanie struktury projektowej opartej na agentach. Każdy zatem element tej struktury, zarówno związany z ludźmi jak i technologiami czy innymi źródłami wiedzy (jak modele ocenowe czy dobre praktyki zarządzania projektami), posiada swoją reprezentację w postaci agenta. W tym badaniu postanowiono wykazać, że takie podejście jest zasadne, czyli że każdy agent tej struktury może zostać określony właściwymi atrybutami — cechami i zachowaniami oraz że w oparciu o te atrybuty agenty mogą realizować zadania w ramach projektu.

Badanie, którego celem jest weryfikacja warstwy funkcjonalnej na poziomie definiowania w modelu szczegółowym **a2M**, dotyczyło projektu prowadzonego według lekkiej metody zarządzania projektami informatycznymi SCRUM

Celem projektu, jak zaznaczono na wstępie, było wytworzenie decyzyjnego systemu wsparcia dla kierowników. Przyjęto więc w realizacji projektu role typowe dla metody SCRUM, przypisując je do odpowiednich osób. Uznano jednak, że ze względu na rozproszenie uczestników podczas realizacji należy zastosować podejście agentowe. Zastosowanie podejścia agentowego oznacza, że zespół nie będzie spotykał się na codziennych spotkaniach, ale role zostaną uzależnione (zdefiniowane) jak agenty. Stąd też zamiast klasycznych ról dla metody SCRUM użyto określeń związanych z agentami. Powołano na tej podstawie następujące role: agent klienta, agent menedżer oraz agenty zadaniowe. Bardzo szybko zorientowano się podczas realizacji tego projektu, że w celu usprawnienia przepływu informacji w projekcie konieczne jest zdefiniowanie kanałów komunikacji. Agenty pracujące w rozproszeniu muszą wymieniać ze sobą informacje w sposób sformalizowany. Wybrano więc główne kanały przesyłania informacji (SMS, e-mail, rozmowa telefoniczna). Jednocześnie uznano, że w celu zapewnienia właściwej organizacji pracy konieczne jest sprecyzowanie hierarchii ważności komunikatów (priorytetyzacja) oraz które agenty mogą zlecać wykonanie określonych zadań innym. Na tej podstawie udało się określić następujące cechy oraz możliwe zachowania poszczególnych agentów. W tabeli 5.7 przedstawiono typy agentów wyodrębnione w systemie agentowym na poziomie definiowania warstwy funkcjonalnej

Tabela 5.7 Typy agentów wyodrębnione w systemie agentowym na poziomie definiowania warstwy funkcjonalnej

TYP AGENTA	CECHY	ZACHOWANIA
Agent MENEDŻER	Zdolności kierownicze	Zleca klientowi przygotowanie wymagań. Zleca agentom zadaniowym wykonanie zadań.
Agent KLIENTA	Umiejętność precyzowania wymagań	Odbiera komunikaty od agenta kierownika . Wykonuje zlecenia od kierownika.
Agent ZADANIOWY	Znajomość języków programowania	Odbiera komunikaty od kierownika. Wykonuje zlecenia w oparciu o komunikaty kierownika. Raportuje stan kierownikowi.

Źródło: opracowanie własne

Jak wynika z tabeli 5.7, stosując podejście agentowe w zarządzaniu projektem, konieczne jest określenie cech poszczególnych agentów oraz zdefiniowanie zachowań poszczególnych agentów. W ten sposób możliwe jest zapewnienie właściwej organizacji prac projektowych oraz komunikacji pomiędzy poszczególnymi agentami. Każdy agent identyfikuje zależności względem innego agenta i podejmuje określone akcje (zachowania) po otrzymaniu zlecenia (komunikatu).

Zaobserwowano jednak w trakcie trwania eksperymentu pewne problemy. Chcąc bowiem postępować z zasadami zawartymi w metodzie SCRUM zauważono brak agenta odpowiedzialnego za poprawny przebieg metody. Zgodnie bowiem z dobrymi praktykami zawartymi w metodzie SCRUM, za poprawność zarządzania projektem według tej metody odpowiada osoba, której powierza się rolę tzw. Mistrza Młyna (ang. *Scrum Master*). Zauważono, że żadnemu z uczestników nie przypisano cech związanych z kierowaniem pracami według metody SCRUM. Zauważono więc zapotrzebowanie na kolejnego agenta. Z racji ograniczonych zasobów ludzkich nie było to możliwe i odpowiedzialność typową dla Mistrza Młyna powierzono ostatecznie kierownikowi.

Wnioski

Przeprowadzone badanie pozwoliło na wykazanie możliwości definiowania cech i zachowań agentów reprezentujących poszczególne elementy struktury projektowej. Umożliwiło identyfikację poziomów definiowania modelu szczegółowego a2M oraz warstw funkcjonalności. Wskazało możliwości tworzenia struktur agentowych. Podczas tworzenia struktur agentowych wykorzystano procesy adaptacji tych struktur w oparciu o wiedzę o członkach zespołu projektowego i przetwarzanie tej wiedzy. Wnioski z badania 3-R przedstawiono w tabeli 5.8.

Tabela 5.8 Wnioski z badania 3-R

		Kod badania: 3-R	
Realizacja celu badania	Cel badania osiągnięty		
Zidentyfikowane poziomy modelu szczegółowego a2M	Definiowania Zarządzania	Nie istnieją procesy adaptacji	
Zidentyfikowane warstwy modelu szczegółowego a2M	Wiedzy Przetwarzania Funkcjonalności	Istnieją procesy adaptacji struktur agentowych	

Źródło: opracowanie własne

Podstawowym wnioskiem płynącym z przeprowadzonego eksperymentu jest to, że zakładane w modelu definiowanie agentów jako odpowiedników określonych ról zespołowych zostało potwierdzone. Można zatem wykorzystywać agenty jako formę organizacji zespołu oraz zarządzania takim zespołem. Jest to bardzo cenny wniosek płynący dla menedżerów projektów, którym zaleca się bardzo dokładne definiowanie własności takich agentów (czyli wszystkich elementów struktury projektowej).

Druga wyraźna konkluzja to potwierdzenie założenia, że każdy agent funkcjonuje poprawnie w strukturze agentowej tylko wtedy, gdy jego własności są precyzyjnie opisane. Te agenty, którym określono w eksperymencie cechy i zachowania, reagowały poprawnie na zlecone im zadania (tzn. wykonywały je terminowo i nie zaobserwowano żadnych problemów z przepływem informacji). Stąd też uznano za słuszne, aby stosując adaptacyjny agentowy model zarządzania projektami, definiować

agenty w sposób możliwie dokładny, opierając się na przypisaniu im cech i zachowań, czyli odpowiedzialności oraz możliwego pola działania. Takie podejście okazuje się słuszne także z punktu widzenia kontroli przepływów informacyjnych w projekcie informatycznym.

Kolejnym wnioskiem płynącym z eksperymentu jest efekt pewnych spostrzeżeń odnośnie dobrych praktyk. Jak zaznaczono w opisie przebiegu eksperymentu, w pewnym momencie zaczęto zastanawiać się nad możliwością traktowania dobrych praktyk jako agenty. Do tej pory odnoszono bowiem agenty wyłącznie do odpowiedników osób realizujących projekt. Zauważono jednak, że każda właściwie dobra praktyka ma określone cechy (np. dotyczy zespołu czy planowania) oraz określone zachowania (np. stosowana do projektów o małej entropii). Stąd też zauważono, że dobrą praktykę można także przedstawić jako agenta, który wykorzystywany jest w projekcie tylko w określonym momencie, w zależności od potrzeb. Zdecydowano również, aby wzorem metody SCRUM zalecającej obecność Mistrza Młyna, w projektach informatycznych zarządzanych zgodnie z proponowanym modelem szczegółowego **a2M**, powoływać agenta metody, czyli osobę, która zajmie się przygotowaniem dobrych praktyk pod potrzeby danego projektu informatycznego. W małych projektach (gdzie uczestników jest niewielu, a zakres obowiązków mały) uznaje się, że taką rolę może pełnić np. kierownik projektu. To spostrzeżenie odnośnie traktowania dobrych praktyk jako agentów okazało się również istotne z punktu widzenia dalszych prac. Skoro bowiem opracowywany model ma być modelem adaptacyjnym, czyli dopasowującym dobre praktyki zarządzania projektami w zależności od struktury projektowej, to stosowanie dobrych praktyk jako agenty jest niezwykle istotne z punktu widzenia późniejszej adaptacji.

Ostatnim bardzo istotnym wnioskiem płynącym z przeprowadzonego eksperymentu jest to, że po raz kolejny (mimo niewielkiego projektu) bardzo wyraźnie zaznaczyły się zależności hierarchiczne pomiędzy agentami. Takie zależności zaczęto roboczo w eksperymencie nazywać „relacjami”, a po zakończeniu projektu uznano za jedną z najważniejszych przyczyn zakończenia projektu sukcesem (w terminie, przy założonym budżecie). Zaczęto się więc zastanawiać, czy nie powinno się w projektach zarządzanych za pomocą modelu szczegółowego **a2M** zacząć zwracać uwagi na znaczenie relacji pomiędzy wszystkimi agentami realizowanego projektu. Zwrócono bowiem uwagę na relacje hierarchiczne, które porządkowały przepływ komunikatów, ale mówiono również o wzajemnych interakcjach dotyczących zaufania, koleżeństwa i podobnych zachowań organizacyjnych.

Przeprowadzony eksperyment oraz wnioski z niego płynące wskazują na kolejne możliwości modyfikacji modelu szczegółowego **a2M**. Zauważono bowiem, że dobre praktyki należy uwzględniać w modelu jako niezależne agenty. Zauważono również, że pomiędzy agentami powinny być zdefiniowane zależności nazywane relacjami oraz że stosując podejście agentowe uwidaczniają się zależności hierarchiczne pomiędzy różnymi agentami. Stąd też w kolejnych eksperymentach badawczych postanowiono przyjrzeć się bardziej relacjom między agentami.

5.1.4 Badanie 4-R. Weryfikacja warstwy zarządzania modelem (możliwości współpracy agentów)

Celem kolejnego eksperymentu badającego zasadność predykcyjną modelu szczegółowego **a2M** było zweryfikowanie, jak agenty współpracują ze sobą przy realizacji projektu. Postanowiono zweryfikować sposób funkcjonowania agentów przez ocenę realizacji projektów informatycznych przez grupy studentów w ramach przedmiotu Zarządzanie Przedsięwzięciami Informatycznymi. Eksperyment trwał przez 5 miesięcy (jeden semestr). Zespoły utworzono ze studentów specjalności Zarządzanie Technologiemi Informatycznymi, następnie uzgodniono tematy projektów do wykonywania (m.in. aplikacja bazodanowa, serwis WWW, sklep internetowy), a następnie obserwowano i monitorowano współpracę wśród poszczególnych uczestników, traktowanych jako agenty. W tabeli 5.9 przedstawiono założenia do przeprowadzania badania.

Tabela 5.9 Założenia do badania 4-R

		Kod badania 4-R
Cel	Celem eksperymentu jest weryfikowanie wcześniejszych założeń zawartych we wnioskach z eksperymentu 3-R. Zaobserwowano bowiem, że pomiędzy agentami tworzącymi zależności hierarchiczne przepływ komunikatów był bardzo wydajny. W niniejszym eksperymencie należy zatem sprawdzić zachowanie się agentów w trakcie współpracy między nimi. Pozwoli to na usystematyzowanie wiedzy na temat relacji między agentami.	
Środowisko badania	Na środowisko eksperymentu wybrano małe zespoły projektowe o niedużym doświadczeniu w zarządzaniu projektami. Część obserwowanych zespołów należała do koła naukowego, część realizowała swoje prace w ramach zajęć dydaktycznych z przedmiotu zarządzanie projektami informatycznymi. W każdym z przypadków należało wykonać określone prace zespołowe.	
Poziomy model a2M do identyfikacji	do potwierdzenia – poziom zarządzania inne poziomy?	Procesy adaptacji?
Warstwa modelu a2M do identyfikacji	do potwierdzenia – warstwa wiedzy inne warstwy?	Procesy adaptacji?
Początek realizacji	październik 2010	ukończono
Termin zakończenia	luty 2011	ukończono

Źródło: opracowanie własne

Opis przebiegu badania

Niniejszy eksperyment miał za zadanie sprawdzenie, jak zachowują się poszczególne agenty mające ze sobą współpracować oraz czy relacje występujące między agentami mają wpływ na przebieg pracy. Stąd też nadrzędnym celem tego badania było usystematyzowanie wiedzy o agentach oraz uzyskanie potwierdzenia, że agenty, które definiuje się jako niezależne jednostki (posiadające przypisane cechy i zachowania), mogą współdziałać w określonych sytuacjach projektowych oraz że jakość działania agentów zależy od rodzaju relacji między nimi.

Do tego celu wykorzystano zespoły złożone ze studentów ostatniego roku studiów magisterskich, którzy studiowali na profilu związanym z zarządzaniem technologiami. W skład tych studentów wchodził zarówno inżynierowie zarządzania, jak i inżynierowie informatyki. Pozwoliło to na tworzenie zespołów składających się z osób z wiedzą zarówno zarządczą, jak i wytwórczą.

Struktura zespołów, podobnie jak w kilku wcześniejszych przypadkach, stanowiła strukturę rozproszoną. Studenci uczestniczący w eksperymencie spotykali się doraźnie tylko w ramach zajęć (ewentualnie spotkań koła naukowego), ale większość prac prowadzili poza zajęciami stosunkowo niezależnie (czyli każdy we własnym domu). Zadaniem studentów było wytworzenie określonego produktu (proste oprogramowanie bazodanowe przydatne do zarządzania firmą, np. fakturowanie, rozliczenia finansowe z dostawcami itp.).

Sposób konstruowania zespołów został opracowany przez autora niniejszej rozprawy i oparty na teście Belblina [7]. Postanowiono w pierwszej kolejności określić predyspozycje do pracy zespołowej poszczególnych członków, a następnie utworzyć na tej podstawie 4 zespoły. Dwa z tych zespołów składały się z osób o bardzo wyraźnych cechach (predyspozycjach) pozyskanych z testu Belblina

(wyraźny lider, wyraźny skrupulatny wykonawca, wyraźny innowator itp.). Dwa pozostałe utworzono z osób o cechach mniej wyraźnych, czyli osób, u których nie zaznaczały się widoczne cechy liderские ani cechy związane z innymi pozycjami w zespole.

Taki podział zespołów miał na celu zweryfikowanie zachowań poszczególnych agentów, jak i całych zespołów w zależności od tego, czy cechy takiego agenta są jasno zdefiniowane. Dodatkowo polecono zespołom ustalenie ram współpracy i przygotowanie dokumentu przedstawiającego te ramy. Okazało się podczas tego eksperymentu, że w zespołach posiadających wyraźnych liderów i wyraźne cechy, zdefiniowanie takich ram współpracy odbywało się bardzo szybko i ramy były definiowane w sposób bardzo precyzyjny. Ramy współpracy miały być dla studentów określeniem zachowań, jakie może przyjmować każdy z agentów. Takie podejście pozwoliło jednocześnie na weryfikację działania agentów w zespołach, w których zachowania te były dobrze zdefiniowane i w których zachowania były zdefiniowane w sposób mniej precyzyjny.

W trakcie trwania badania poddano ocenie poziom tworzonej dokumentacji zawierającej doświadczenia płynące ze współpracy. W ten sposób postanowiono określić jakość relacji pomiędzy poszczególnymi agentami. Studenci mieli więc za zadanie nie tylko wykonać (zrealizować) określony projekt informatyczny, ale jednocześnie przygotować sprzężenie zwrotne z realizacji tego projektu. Liderzy poszczególnych zespołów spotkali się więc z autorem badania po zakończeniu projektu. Polecono liderom, aby ocenili współpracę w swoim zespole w skali od 1 do 10. Liderzy zespołów z wyraźnymi cechami agentów najczęściej oceniali tę współpracę bardzo wysoko. Liderzy pozostałych zespołów zwracali uwagę na problemy z wzajemną komunikacją, które wywoływały opóźnienia w terminowej realizacji zadań.

Ważną obserwacją podczas tego badania jest również fakt, że prace prowadzone przez zespoły złożone z agentów o wyraźnych cechach zrealizowały swoje projekty w krótszym czasie. Jakość prezentowanych produktów była co prawda zbliżona (studenci dysponowali podobnym poziomem wiedzy), może jedynie z lekką (nie wyraźną) przewagą tych projektów, które zrealizowały zespoły złożone z agentów o wyraźnych cechach. Natomiast czas realizacji i stopień zadowolenia ze współpracy był o wiele większy w zespołach, w których funkcjonowały wyraźnie zdefiniowane agenty.

Na końcu poproszono również o ocenę lidera przez poszczególnych uczestników (agenty). Zaobserwowano, że zespoły z wyraźnie zdefiniowanymi agentami lepiej oceniały pracę liderów, a nawet pozwalały sobie na szereg pochwał i konstruktywnych uwag. Pozwoliło to domniemywać, że zespoły złożone z wyraźnie zdefiniowanych agentów dojrzewały w trakcie realizacji prac i były w stanie wyciągać wnioski z ewentualnych błędów zaobserwowanych podczas realizacji projektów.

Wnioski

Przeprowadzone badanie pozwoliło na zweryfikowanie możliwości współpracy agentów przy realizacji projektu. Umożliwiło identyfikację poziomu zarządzania modelem szczegółowego **a2M** oraz warstw wiedzy i przetwarzania. Wskazało możliwości współpracy agentów. Podczas analizy możliwości współpracy agentów analizowano procesy adaptacji tych struktur w oparciu o wiedzę i przetwarzanie tej wiedzy o członkach zespołu projektowego. Wnioski z badania 4-R przedstawiono w tabeli 5.10.

Tabela 5.10 Wnioski z badania 4-R

		Kod badania: 4-R
Realizacja badania	celu	Cel badania osiągnięty

Zidentyfikowane poziomy modelu szczegółowego a2M	Definiowania Zarządzania	Istnieją procesy adaptacji struktur agentowych
Zidentyfikowane warstwy modelu szczegółowego a2M	Wiedzy Przetwarzania	Istnieją procesy adaptacji struktur agentowych

Źródło: opracowanie własne

Podstawowy wniosek, jaki płynie z przeprowadzonego badania, potwierdził założenie, że stosując podejście agentowe kluczowe jest wyraźne zdefiniowanie poszczególnych agentów, najlepiej za pomocą cech i zachowań. Te agenty, które miały dobrze zdefiniowane cechy i wyraźnego lidera, bardzo szybko potrafiły określić swoje zachowania, a następnie pracować według ustalonych zachowań. Okazało się również, że jakość pracy takich agentów została oceniona wyżej niż w przypadku zespołów, gdzie cechy agentów nie były zbyt wyraźne.

Kolejnym wnioskiem płynącym z przeprowadzonego eksperymentu jest wskazanie wszystkich uczestniczących na istotną rolę lidera. Pokazuje to, że zarządzając zespołem opartym na agentach konieczne jest zdefiniowanie agenta nadrzędnego, który będzie monitorował zachowania pozostałych agentów. Obecność wyraźnego lidera (przywódcy) pozwala zatem na stwierdzenie, że kluczowe dla przebiegu projektu zarządzanego zgodnie z modelem szczegółowego **a2M** jest wyodrębnienie agenta menedżera, który będzie w stanie zarządzać wszystkimi pozostałymi agentami.

Związany z powyższym wniosek jest również taki, że agent menedżer powinien (tak jak w omawianym eksperymencie) uzgodnić z pozostałymi agentami zakresy odpowiedzialności, czyli możliwe ramy zachowań poszczególnych agentów.

Na uwagę zasługuje również fakt, że takie wnioski są bardzo ważne dla kierowników projektów informatycznych. Jeżeli bowiem decydują się na stosowanie zaleceń modelu szczegółowego **a2M**, to muszą pamiętać o tym, że ciężar odpowiedzialności za poprawność relacji spoczywa na nich.

Warto jednocześnie zwrócić uwagę właśnie na ważność relacji wypływającą z przeprowadzonego eksperymentu. W zespołach, w których te relacje były bardzo dobrze określone, komfort pracy był o wiele lepszy. Większe było również tempo prac. Pozwala to zatem przyjąć wniosek, że stosując podejście agentowe prace projektowe realizowane są szybciej. Podejście to nie daje być może zysku jakościowego produktu, ale przy pełnej definicji agentów może prowadzić do skrócenia czasu realizacji projektu (efekt działania dobrze zdefiniowanych relacji), co w myśl zasady „czas to pieniądz” może zapewniać oszczędności w projektach informatycznych.

Reasumując powyższe rozważania, należy uznać, że koncepcja wykorzystania agentów i relacji między nimi proponowana w modelu szczegółowego **a2M** może mieć duże znaczenie z punktu widzenia organizacji pracy w projekcie oraz z punktu widzenia motywowania zespołu. Wcześniejsze eksperymenty wykazywały, że stosowanie agentów znacznie ułatwia kontrolowanie. Niniejszy eksperyment pokazał zasadność modelu z punktu widzenia organizowania i motywowania. Na tej podstawie można zatem uznać model szczegółowy **a2M** za model usprawniający zarządzanie projektami informatycznymi. Takie wnioski pozwoliły na tworzenie planu dalszego doskonalenia modelu, zwłaszcza w obszarze zależności pomiędzy agentami, czyli relacji między nimi.

W założeniach budowy w modelu szczegółowym **a2M** przyjęto, że kluczowym elementem tego poziomu jest tworzenie relacji pomiędzy agentami modelu. Relacje te mają dwójaki charakter — pokazują zależności hierarchiczne między agentami z jednej strony, z drugiej zaś dotyczą aspektów wpływających na jakość współpracy. Przeprowadzony eksperyment wyraźnie pokazał, że relacje tworzone między agentami uczestniczącymi w pracach z dobrze zdefiniowanym agentem liderem przekładały się na relacje odniesione do jakości współpracy. Stąd też wydaje się słuszne, aby

usprawniając model szczegółowy **a2M**, bardzo duży nacisk położyc na określenie (definiowanie) tych właśnie relacji.

5.1.5 Badanie 5-R. Weryfikacja warstwy zarządzania modelu (definiowania relacji)

Badanie miało na celu określenie możliwości definiowania relacji pomiędzy agentami w projekcie informatycznym. Za środowisko weryfikacji przyjęto trwający trzy lata projekt związany z badaniem technologii informatycznych finansowany przez Ministerstwo Nauki i Szkolnictwa Wyższego. Na podstawie wyników tego projektu przygotowano badanie pozwalające na określenie relacji zachodzących pomiędzy agentami w projekcie informatycznym. Projekt nosił nazwę „Badanie technologii informatycznych z wykorzystaniem systemów inteligentnych”. Warto nadmienić, że również w ramach tego projektu wykonywano prototypy systemów agentowych (omawiane w rozdziale III). Koordynatorem projektu był Zakład Zarządzania Technologiami Informatycznymi. Część implementacyjna realizowana była przez firmę NETHOS wymienioną w eksperymencie 3-R oraz inne osoby wykonujące prace programistyczne na umowę-zlecenie. W tabeli 5.11 przedstawiono założenia do przeprowadzania badania.

Tabela 5.11 Założenia do badania 5-R

		Kod badania	5-R
Cel	Celem badania jest zweryfikowanie istotności relacji pomiędzy agentami modelu szczegółowego a2M . Założono, że poprawność realizacji projektów zależy od uprzednio zdefiniowanej struktury projektowej opartej na agentach. Stąd też w eksperymencie należy zweryfikować, jak zachowują się poszczególne agenty uczestniczące w projekcie oraz czy możliwe jest stworzenie struktury hierarchicznej między agentami w celu usprawnienia przepływu informacji i realizacji zadań.		
Środowisko badania	Jako środowisko przeprowadzenia badania przyjęto jeden z projektów realizowanych przez Zakład Zarządzania Technologiami Informatycznymi na Politechnice Gdańskiej. Projekt dotyczył wytworzenia systemu zarządzania projektami informatycznymi i realizowany był na potrzeby MNiSW. W projekcie uczestniczyli członkowie ZZTI oraz osoby z firm zewnętrznych, a także eksperci z innych uczelni.		
Poziomy model a2M do identyfikacji	do potwierdzenia – poziom zarządzania inne poziomy?		Procesy adaptacji?
Warstwa modelu a2M do identyfikacji	do potwierdzenia – warstwa przetwarzania inne warstwy?		Procesy adaptacji?
Początek realizacji	Styczeń 2009	Ukończono	
Termin zakończenia	Grudzień 2010	Ukończono	

Źródło: opracowanie własne

Opis przebiegu eksperymentu

Badanie zostało przeprowadzone po zakończeniu projektu informatycznego realizowanego na potrzeby Ministerstwa Nauki i Szkolnictwa Wyższego. Wybrano ten projekt jako środowisko weryfikacji przydatności warstwy przetwarzania w poziomie zarządzanym modelu szczegółowym **a2M** ze względu na złożoną strukturę projektu. Złożoność struktury była typowa dla założeń przyjętych w poziomie definiowania modelu szczegółowego **a2M**. Zakładano bowiem, że w projekcie

mogą uczestniczyć, oprócz lidera i zespołu realizującego zadania, eksperci z zewnętrznych ośrodków. Ponadto umowa z Ministerstwem Nauki i Szkolnictwa Wyższego wyraźnie precyzowała odpowiedzialność za przebieg prowadzonych prac, powierzając je jednej osobie.

Takie warunki początkowe odpowiadały środowisku agentowemu. Dodatkowym czynnikiem sprzyjającym zastosowaniu agentów był fakt, że większość członków zespołu pracowała stosunkowo niezależnie, to znaczy że nie spotykali się w jednym miejscu, ale działali osobno, czasami w ośrodkach oddalonych od siebie.

W trakcie analizy dokumentacji postanowiono ustalić zależności hierarchiczne poszczególnych uczestników projektu. W ten sposób weryfikacji poddano możliwość oraz zasadność tworzenia relacji pomiędzy agentami w modelu szczegółowym **a2M**. Na tej podstawie stworzono strukturę projektową opisaną poniżej.

Nadrzędnym organem, któremu podlegała realizacja prac, stanowiącym jednocześnie klienta, było Ministerstwo Nauki i Szkolnictwa Wyższego. Beneficjentami rozwiązania (użytkownikami końcowymi) byli menedżerowie organizacji informatycznych potrzebujący oceny technologii informatycznych. Z uwagi na ograniczone kontakty z klientem i beneficjentami (tylko podczas branżowych konferencji) uznano, że w celu usprawnienia realizacji projektu najważniejsze jest ustalenie relacji pomiędzy osobami odpowiedzialnymi za wytworzenie produktu we wspomnianym projekcie. Podczas ustaleń odnośnie zależności pomiędzy agentami zaangażowanymi w projekt powołano podstawowe agenty, jak Agent Menedżer i Agent Koordynator.

Agent Menedżer, czyli osoba która miała za zadanie zarządzać wszystkimi aspektami projektu, związanymi między innymi z modelowaniem (było to konieczne z punktu widzenia założeń projektowych zaakceptowanych przez MNiSW). Agentowi temu powierzono również prowadzenie negocjacji z zamawiającym (klientem), jak również szereg wspierających działań strategicznych prowadzących do usuwania wszelkich przeszkód (barier) pojawiających się w projekcie. Ustalono również, że pozostałe agenty będą podlegać Agentowi Menedżerowi i będą zobowiązane do podjęcia działań na zlecenie Agent Menedżera. Agent Menedżer stanowił odpowiednik kierownika projektu.

Okazało się także, że osiągnięcie ostatecznego produktu projektu możliwe jest tylko w sytuacji, kiedy będzie właściwa koordynacja działań na poziomie operacyjnym. Z uwagi jednak na szereg działań strategicznych, Agent Menedżer (kierownik projektu) nie był w stanie nadzorować wszystkich działań na poziomie operacyjnym. Zaistniała w związku z tym konieczność powołania innego agenta, którego zadaniem będzie koordynowanie prac poszczególnych innych rozproszonych agentów. Na tej podstawie powołano Agent Koordynatora, który odbierał komunikaty od Agent Menedżera oraz miał uprawnienia do zlecania innym agentom określonych zadań.

Na najniższym poziomie tej struktury znaleźli się wykonawcy (w tym podmiot wytwarzający oprogramowanie, specjaliści od modelowania czyli AGENTY ZADANIOWE).

Tak stworzona struktura agentowa pozwoliła w kolejnym kroku tego badania sprawdzić zachowania poszczególnych agentów. Okazało się, że przepływy informacji na wyższym szczeblu (pomiędzy Agentem Menedżerem i Agentem Koordynatorem) przebiegały bez żadnych przeszkód. Przepływy informacji pomiędzy Agentem Koordynatorem oraz pozostałymi agentami operacyjnymi (funkcjonalnymi) były również poprawne. Uznano zatem, że przyjęta struktura hierarchiczna jest słuszna, a podejście agentowe pozwala lepiej zorganizować prace poszczególnych członków projektu. Skoro zatem układ hierarchiczny funkcjonował sprawnie, postanowiono poddać analizie poprawność tych przepływów. Zauważono bowiem, że nie jest to spowodowane wyłącznie zależnościami hierarchicznymi.

Analiza zachowań poszczególnych agentów wykazywała poprawne reagowanie na zlecenia (komunikaty) płynące od agenta menedżera oraz agenta koordynującego. Zaobserwowano także, że oprócz tej poprawności, jakość komunikacji była zróżnicowana w zależności od agentów. Okazało się także, że pomiędzy niektórymi agentami tworzone są relacje oparte na koleżeństwie lub wspólnych zainteresowaniach, co znacznie poprawiało komunikaty. Uznano zatem, że struktura hierarchiczna

stworzona między agentami porządkuje przepływy komunikacyjne. Wydaje się także, że przy tworzeniu zależności pomiędzy agentami należy również uwzględnić możliwość wystąpienia tych relacji nieformalnych. Okazało się to jednak trudne do opisanego podczas prowadzonego badania.

W sytuacji, kiedy znane były zachowania poszczególnych agentów ułożonych w relacjach hierarchicznych, zastanawiano się w trakcie badania, w jaki sposób umieszczać w strukturze pozostałe agenty, takie jak Agent Metody czy Agent Ekspert. Zauważono problem w doborze eksperta dziedzinowego. Natomiast w doborze dobrych praktyk zarządzania projektami należy praktyki te wstępnie rozpoznać. Zauważono więc brak dedykowanego agenta, który potrafiłby komunikować się z ekspertami w określonych sytuacjach i zlecać ekspertom wykonanie określonych działań (np. przygotowanie jakiejś analizy). Stąd też powołano dodatkowego agenta, którego określono mianem Agent Przeszukującego, który zdobywał informacje o dobrych praktykach zarządzania projektami, które można wykorzystać (dobierać) do projektów oraz który komunikował się z ekspertami rozproszonymi po całym kraju. Bardzo szybko okazało się, że powołując nowego agenta (Agent Przeszukujący) powstała kolejna relacja hierarchiczna w strukturze projektowej. Agent ten podlegał Agentowi Menedżerowi, ale nadzorował pracę ekspertów oraz zajmował się dobrymi praktykami zarządzania projektami (które również zgodnie z wnioskami z poprzedniego eksperymentu traktuje się jako agenty).

Analiza końcowej części dokumentacji projektu wskazuje, że zgromadzono bardzo znaczne zasoby wiedzy dostarczane przez poszczególnych agentów (modele, sprawozdania, raporty). Zaczęto się zastanawiać, jak uporządkować tę wiedzę z punktu widzenia celu projektu. Wskazano, że rozsądnym rozwiązaniem z punktu widzenia istniejącej struktury agentowej będzie posiadanie odpowiedniego agenta, który potrafi przetwarzać wiedzę i generować odpowiednie wnioski. Uznano także, że w sytuacji, kiedy zasoby wiedzy w projekcie zaczynają być coraz większe, należy powierzyć znaczny zakres odpowiedzialności za tę wiedzę odpowiedniemu agentowi. Może to być agent wiedzy lub agent wnioskujący. Nie był on brany pod uwagę w realizowanym badaniu, ale pod jego koniec zwrócono uwagę na zapotrzebowanie na agenta odpowiedzialnego za zarządzanie wiedzą w projekcie.

Wnioski

Przeprowadzone badanie pozwoliło na zweryfikowanie relacji pomiędzy agentami przy realizacji projektu. Umożliwiło identyfikację i sprecyzowanie relacji na poziomie zarządzania modelem szczegółowego a2M oraz warstw wiedzy, przetwarzania i funkcjonalności. Wskazało mechanizmy współpracy agentów. Podczas analizy relacji agentów oceniano procesy adaptacji tych relacji w oparciu o wiedzę i przetwarzanie tej wiedzy o relacjach zespołu projektowego. Wnioski z badania 5-R przedstawiono w tabeli 5.11.

Tabela 5.11 Wnioski z badania 5-R

		Kod badania: 5-R
Realizacja celu badania	Cel badania osiągnięty	
Zidentyfikowane poziomy modelu szczegółowego a2M	definiowania zarządzania	Istnieją procesy adaptacji struktur agentowych
Zidentyfikowane warstwy modelu szczegółowego a2M	wiedzy przetwarzania funkcjonalności	Istnieją procesy adaptacji struktur agentowych

Źródło: opracowanie własne

Przeprowadzone badanie potwierdziło potrzebę zastosowania podejścia agentowego do weryfikacji relacji pomiędzy wykonawcami omawianego projektu. Utworzenie struktury opartej na relacjach hierarchicznych znacznie ułatwia zarządzanie wszystkimi podmiotami zaangażowanymi w realizację projektu.

Badanie potwierdziło przyjęte założenie, że po zdefiniowaniu agentów należy określić odpowiednie relacje. Obecność relacji hierarchicznych pozwala na usprawnienie przepływu informacji w projekcie. Zauważono również, że taki układ relacji wspomaga zarządzanie zespołem projektowym, każdy bowiem uczestnik ma przydzielone swoje cechy oraz zachowania i poruszając się w ramach tych atrybutów typowych dla agentów realizuje swoje prace.

Innym ważnym wnioskiem wynikającym z badania jest stwierdzenie, że stosowanie podejścia agentowego i tworzenie struktury między agentami odpowiada specyfice dzisiejszych projektów. To badanie pokazało, że możliwe jest aktywowanie poszczególnych agentów w zależności od entropii projektu (jak np. eksperta, który na co dzień nie współpracuje z zespołem).

Nie udało się niestety na podstawie przeprowadzonego badania opisać relacji nieformalnych, jakie powstają lub mogą wystąpić pomiędzy agentami systemu. Uchwycenie pewnych wzajemnych sympatii uczestników projektu czy też wspólnych zainteresowań jest problemem złożonym. Ważnym wnioskiem płynącym z przeprowadzonego badania jest stwierdzenie obecności takich relacji oraz ich przełożenie na sposób realizacji projektu informatycznego. Jeżeli relacje uda się zaobserwować, możliwe jest stworzenie agentowego systemu zarządzania, który znacznie usprawnia zarządzanie zespołem i podmiotami powiązаныmi z projektem (klient, ekspert itp.).

Przeprowadzone badanie pozwala na wyciągnięcie wniosków celem modyfikacji modelu szczegółowego **a2M**. Okazało się, że zdefiniowanie relacji hierarchicznych między agentami jest poprawne z punktu widzenia usprawnienia przepływów informacji między agentami. Jest to cenne usprawnienie związane z zarządzaniem projektem informatycznym. Natomiast zastanawiający był fakt, że wydajność agentów na poziomie operacyjnym (zespoły projektowe) zależała od czynników związanych z dojrzałością zespołu. Uznano zatem za konieczność uwzględnienie w modelu szczegółowym **a2M** poziomów dojrzałości zespołu dostawcy, klienta oraz entropii projektu.

5.1.6 Badanie 6-R. Weryfikacja warstwy zarządzania modelu (zarządzanie relacjami)

Badanie miało na celu wykazanie, że zdefiniowanie relacji między agentami oraz zapewnienie dokładnego przepływu informacji i zleceń między nimi prowadzi do usprawnienia realizacji zadań w projekcie informatycznym. Możliwe jest to dzięki ustaleniu relacji hierarchicznych oraz przypisaniu agentom cech i zachowań, w ramach których funkcjonują one w projekcie. Do badań wybrano projekt wdrożenia serwisu WWW na potrzeby Krajowej Konferencji Inżynierii Oprogramowania organizowanej w Gdańsku przez Polskie Towarzystwo Informatyczne, Oddział Pomorski w 2010 r. Projekt wdrożenia serwisu WWW rozpoczęto w listopadzie 2009. W tabeli 5.12 przedstawiono założenia do przeprowadzania badania.

Tabela 5.12 Założenia do badania 6-R

Kod badania 6-R	
Cel	Sprawdzenie, czy zdefiniowana struktura hierarchiczna agentów wpływa na realizację zadań projektowych.
Środowisko badania	Projekt wdrożenia nowego serwisu WWW na potrzeby jednej z naukowych konferencji. Dostawcą oprogramowania była firma informatyczna z bardzo doświadczonym zespołem. Klientem był doświadczony w projektach informatycznych przedstawiciel jednostki naukowo-badawczej.

Poziomy modelu a2M do identyfikacji	do potwierdzenia – poziom zarządzania inne poziomy?		Procesy adaptacji?
Warstwa modelu a2M do identyfikacji	do potwierdzenia – warstwa funkcjonalności inne warstwy?		Procesy adaptacji?
Początek realizacji	Listopad 2009	Ukończono	
Termin zakończenia	Luty 2010	Ukończono	

Źródło: opracowanie własne

Opis przebiegu eksperymentu

Uprzednie badania wykazały konieczność definiowania agentów oraz łączenia ich relacjami hierarchicznymi w ramach struktury. Takie podejście poprawia przepływ komunikacji między agentami zaangażowanymi w realizację projektu. Podejście to możliwe było do weryfikacji w ramach projektu realizowanego na potrzeby konferencji organizowanej przez Polskie Towarzystwo Informatyczne, Oddział Pomorski. Uznano, że na potrzeby konferencji organizowanej należy wdrożyć odpowiedni serwis WWW, który pozwoli zarządzać rejestracjami uczestników oraz komunikacją z nimi.

Do realizacji tego projektu powołano zespół złożony zarówno z przedstawiciela klienta (pracownik jednostki naukowo-badawczej reprezentujący jednocześnie Polskie Towarzystwo Informatyczne), jak również dwóch programistów mających wykonać produkt tego projektu w postaci wspomnianego serwisu WWW.

Kierownik omawianego projektu w pierwszej kolejności zdefiniował podstawowe agenty. Występował agent koordynator (w osobie prezesa Polskiego Towarzystwa Informatycznego), jak również agent menedżer (w osobie kierownika projektu). Obydwa agenty zarządzające (koordynator i kierownik) miały uprawnienia do zlecenia zadań dwom agentom funkcjonalnym (wykonawcy serwisu WWW — programiści). Uczestnicy projektu (identyfikowani jako agenty z przypisanymi uprawnieniami oraz zakresem obowiązków, czyli cechami oraz zachowaniami w myśl podejścia agentowego) spotkali się na początku projektu, a następnie pracowali w środowisku rozproszonym. Dlatego też na pierwszym (i jedynym) roboczym spotkaniu ustalono cechy i zachowania poszczególnych członków zespołu oraz opracowano kanały przepływu informacji. Było to zgodne z założeniami modelu szczegółowego **a2M**. Zdefiniowanie kanałów komunikacji w tym projekcie odzwierciedlało bowiem konieczność zdefiniowania komunikatów stanowiących zlecenia. Należy zaznaczyć, że spośród osób zaangażowanych w projekt (4 osoby, 3 miesiące pracy) wyłoniono strukturę — koordynator projektu oraz agenty mu podległe. Określono także przepływ komunikatów (zlecenie komunikatów) i przystąpiono do realizacji prac.

W trakcie eksperymentu realizowano zadania po uprzednim zdefiniowaniu zarówno struktury hierarchicznej między agentami, jak również po zdefiniowaniu możliwych komunikatów. Podczas przebiegu eksperymentu zauważono, że poszczególne agenty bardzo precyzyjnie zlecały sobie wzajemnie wykonanie określonych zadań. Wszystkie wymagania odnośnie produktu tego projektu (serwisu WWW) były przekazywane terminowo (najczęściej za pomocą opisów tekstowych, ale również w postaci notacji UML). Wymagania zlecone do implementacji przez kierownika były zrozumiałe i implementowane w przewidzianym terminie. Ostatecznie zbudowany serwis WWW spełniał wszystkie założenia i został wykonany w przewidzianych ramach czasowych i zakładanym budżecie. Warto nadmienić, że projekt nie był obarczony ryzykiem, jak również nie nosił znamion dużej niepewności (jego entropię można określić jako małą). Należy dodać, że uczestnicy posiadali duże doświadczenie w realizacji projektów.

Warto zaznaczyć także, że podczas realizacji projektu okazało się, że występuje zapotrzebowanie na eksperta dziedzinowego. Na pewnym etapie prac konieczne było opracowanie elementów graficznych serwisu. Stąd też wystąpiła potrzeba poszukiwania eksperta i włączenia go jako agenta do struktury projektu na okres dwóch tygodni, po czym agent ekspert przestał być członkiem zespołu.

Wnioski

Przeprowadzone badanie pozwoliło na weryfikację procesów zarządzania relacjami pomiędzy agentami przy realizacji projektu. Umożliwiło identyfikację relacji i procesów zarządzania nimi na poziomie zarządzania modelem szczegółowego a2M oraz warstw wiedzy, przetwarzania i funkcjonalności. Podczas analizy relacji agentów oceniano procesy adaptacji tych relacji w oparciu o wiedzę i przetwarzanie wiedzy o relacjach zespołu projektowego. Wnioski z badania 6-R przedstawiono w tabeli 5.13.

Tabela 5.13 Wnioski z badania 6-R

		Kod badania:	6-R
Realizacja celu badania	Cel badania osiągnięty		
Zidentyfikowane poziomy modelu szczegółowego a2M	definiowania zarządzania	Istnieją procesy adaptacji struktur agentowych	
Zidentyfikowane warstwy modelu szczegółowego a2M	wiedzy przetwarzania funkcjonalności	Istnieją procesy adaptacji struktur agentowych	

Źródło: opracowanie własne

Przeprowadzone badanie umożliwiło wykazanie znaczenia relacji między agentami oraz komunikatów między nimi. Obserwacje poczynione podczas realizacji eksperymentu 6-R pozwoliły również na zwrócenie uwagi, że oprócz właściwych relacji między agentami należy uwzględnić poziomy dojrzałości zespołów dostawcy oraz klienta, które wykorzystują te relacje. Innym wnioskiem płynącym z przeprowadzonego eksperymentu jest potwierdzenie roli eksperta w projekcie. Włączenie go do zespołu stanowi typowe zachowanie agentowe, czyli reagowanie na bodźce, działanie tylko w określonym momencie jako reakcja na komunikat o braku specjalisty od grafiki komputerowej.

Przeprowadzone badanie wykazało znaczny wpływ na przebieg realizacji zadań poziomów dojrzałości podczas definiowania zależności między agentami. Jednak nie zawsze osoby współpracujące ze sobą będą prezentować wysoki poziom dojrzałości (klienta czy członka zespołu dostawcy). W tej sytuacji konieczne jest zdefiniowanie pełnego trójkąta ograniczeń, z uwzględnieniem również entropii projektowej.

5.1.7 Badanie 7-R. Weryfikacja warstwy adaptacji modelu (dobór dobrych praktyk)

Celem badania było określenie możliwości dobierania dobrych praktyk zarządzania projektami przy znanym poziomie dojrzałości zespołu dostawcy, dojrzałości klienta oraz entropii projektowej. Na środowisko przeprowadzenia badania wybrano projekt dotyczący integracji narzędzi Rational wspomagających cykl wytwarzania oprogramowania. Klientem projektu była firma IBM. Dostawcą w tym projekcie było Uniwersyteckie Centrum Kompetencyjnego Technologii Rational powołane na Politechnice Gdańskiej, a wykonawcami członkowie koła naukowego podległego temu Centrum. Klient (firma IBM) zlecił sprawdzenie możliwości integracji technologii Rational. W tabeli 5.14 przedstawiono założenia do przeprowadzania badania.

Tabela 5.14 Założenia do badania 7-R

		Kod badania	7-R
Cel	Celem eksperymentu jest sprawdzenie możliwości realizacji projektu informatycznego, dla którego pomierzony jest poziom dojrzałości dostawcy oraz klienta, a także znany jest poziom entropii z uwzględnieniem dobrych praktyk zarządzania. Eksperyment powinien wykazać, na ile zmienne decyzyjne pozwalają na dobieranie dobrych praktyk zarządzania projektami w zależności od wartości tych zmiennych oraz czy dobranie dobrych praktyk pochodzących z różnych metod zarządzania projektami jest możliwe (czy możliwa jest adaptacja dobrych praktyk).		
Środowisko badania	Środowiskiem eksperymentu był projekt informatyczny polegający na integracji narzędzi w cyklu wytwarzania oprogramowania. Zadaniem osób uczestniczących było sprawdzenie możliwości integracji narzędzi jednego z większych producentów oprogramowania oraz stworzenie modelu integracji na poziomie danych lub na poziomie plików (model API).		
Poziomy model a2M do identyfikacji	do potwierdzenia – poziom adaptacji	inne poziomy?	Procesy adaptacji?
Warstwa modelu a2M do identyfikacji	do potwierdzenia – warstwa przetwarzania	inne warstwy?	Procesy adaptacji?
Początek realizacji:	Kwiecień 2011	wykonano	
Termin zakończenia:	Wrzesień 2011	Zakończono pierwszy etap-analiza możliwości integracyjnych	

Źródło: opracowanie własne

Opis przebiegu eksperymentu

Przeprowadzone badania integracji narzędzi wspomagających procesy organizacji (w tym przypadku proces wytwarzania oprogramowania) miały wykazać przydatność modelu szczegółowego a2M dla warunków dużego projektu o znanych wartościach zmiennych decyzyjnych (entropii projektu, dojrzałości klienta oraz dojrzałości zespołu dostawcy) przy jednoczesnym dopasowaniu dobrych praktyk zarządzania projektami. Taka potrzeba wynikała z wniosków poprzedniego eksperymentu, 7-R, gdzie zauważono konieczność dobierania dobrych praktyk w zależności od tych zmiennych. Okazało się także, że projekt integracji narzędzi, który stanowił środowisko dla eksperymentu, pozwala na uwzględnienie także wszystkich wcześniejszych założeń dla modelu szczegółowego a2M (dotyczących poziomu definiowania oraz zarządzania). Dlatego też podczas opisu tego badania zwrócono uwagę na szczegóły dotyczące także niższych poziomów modelu.

Po dokonaniu wstępnych ustaleń z przedstawicielem klienta określono zakres projektu. Wiadomym było, że należy opracować model pozwalający integrować narzędzia w cyklu wytwarzania oprogramowania. Niestety, od początku trwania projektu nie było wiadomo, na jakim poziomie tej integracji należy dokonać. Czy tylko w oparciu o przesyłanie plików między narzędziami za pomocą funkcji API, czy też w niższej warstwie (na poziomie danych w formacie XML). Z uwagi na fakt, że rodzina narzędzi Rational, które miały zostać poddane analizie, była stosunkowo duża, uznano także, że prace będą realizować pracownicy Zakładu Zarządzania Technologiami Informatycznymi oraz studenci kierunków informatycznych i menedżerskich Politechniki Gdańskiej w ramach kilkusobowych grup (zespołów). Dla każdego z tych zespołów określono poziom dojrzałości oraz wyłoniono lidera. Był to pierwszy krok do zastosowania podejścia agentowego (definiowanie agentów, ich cech oraz zachowań). Liderom przypisano takie zachowania jak komunikacja

z członkami zespołu czy raportowanie wyników do kierownictwa projektu. W celu odzwierciedlenia agentowej struktury zarządzania, którą założono w modelu szczegółowym **a2M**, powołano również taką strukturę w ramach tego projektu. Kierownikiem projektu (agent menedżer) był Profesor kierujący Centrum Kompetencyjnym. Funkcje agenta koordynatora oraz agenta wnioskującego powierzono dwóm asystentom Profesora. Zadaniem agenta koordynatora było nadzorowanie prac poszczególnych zespołów poprzez komunikację z liderami. Rolą agenta wnioskującego w tym projekcie było dobieranie dobrych praktyk zarządzania projektem w zależności od zmiennych decyzyjnych nowego trójkąta ograniczeń.

Podczas badania realizowanego w ramach projektu integracji narzędzi Rational zastosowano kompleksowo model szczegółowy **a2M**. Zastosowanie podejścia agentowego było konieczne również ze względu na rozproszone środowisko realizacji projektu (klient przebywał w innym mieście, kontaktowano się z nim za pośrednictwem wideokonferencji, zespoły złożone z uczestników pracujących w różnych rejonach Gdańska). Kolejnym typowym zjawiskiem dla podejścia agentowego było wyłonienie mentora, który służył pomocą w określonych momentach projektu. Miał być zatem „uruchamiany” w zależności od sytuacji, na zasadzie typowego agenta (stan i akcja jako odpowiedniki cech i zachowań uczestnika projektu). Zadaniem mentora było reagowanie na zlecenia koordynatora (np. dostarczenie przewodników po narzędziach informatycznych analizowanych przez zespoły, zarządzanie licencjami oprogramowania itp.).

Tak zdefiniowana struktura projektu, a tym samym środowisko przeprowadzenia eksperymentu, pozwoliła na realizację kolejnego etapu projektu: pomiaru dojrzałości poszczególnych zmiennych decyzyjnych. Klient, z uwagi na duże doświadczenie, okazał się być klientem zarówno dopasowanym, jak i odpowiednim. Potrafił bowiem sprecyzować wszystkie swoje oczekiwania, a nawet zawrzeć sugestie co do tego, jak wykonać określone zadania związane z integracją narzędzi będących przedmiotem analiz w omawianym projekcie. Zespoły, z racji tego, że złożone były głównie ze studentów (zarówno kierunków menedżerskich, jak i informatycznych), wykazywały się dużą wiedzą merytoryczną i dość dobrymi cechami członków związanymi z pozycjami w zespole (ponad połowa osób miała wyraźnie określone pozycje zespołowe na podstawie testu Belblina, takie jak lider czy skrupulatny wykonawca). Jednak brakowało członkom zespołu doświadczenia związanego z realizacją projektów. Udało się jednak ocenić poziom dojrzałości zespołów, który przedstawiał się na poziomie małym bądź średnim (w przypadku jednego zespołu). Wreszcie dokonano również analizy entropii projektu, która w omawianym case’ie badawczym została uznana za wysoką. Wynikało to z tego, że o ile poszczególne zadania związane z analizą technologii informatycznych były jasne, o tyle dokładna wizja projektu i forma ostatecznego produktu była do końca niejasna. Próbowano oprzeć finalny produkt na zasadach szyny ESB (ang. *Enterprise Service Bus*) albo na modelu XML, ale od początku kierownik projektu wraz z klientem nie byli przekonani, który z wariantów jest słuszny. W związku z taką entropią postanowiono zastosować dobrą praktykę zarządzania projektem, która pozwoli na uruchomienie prac projektowych przy wysokim poziomie entropii. Zdecydowano się więc zastosować pracę w sprintach (zaczepniętą z metody SCRUM). Pierwszy sprint dotyczył analizy technologii i możliwości ich wzajemnego powiązania (które technologie warto integrować z innymi). Zaproponowano, aby podejście do modelowania integracji technologii opracować w kolejnych sprintach. W ten sposób dobrano dobrą praktykę zarządzania projektem do nowego trójkąta ograniczeń. Przy czym okazało się konieczne uwzględnienie także poziomów dojrzałości zespołów. Gdyby realizować projekt zgodnie z metodą SCRUM, zespoły powinny same zaplanować realizację zadań. Jednakże po uwzględnieniu poziomów dojrzałości zespołów (trzy zespoły o małym poziomie dojrzałości — zarządzany i jeden o średnim — ilościowo zarządzany według skali przyjętej dla pomiaru dojrzałości zespołu) uznano, że zespoły nie są w stanie samoorganizować się ze względu na brak wystarczającego doświadczenia w zarządzaniu projektami. O ile bowiem poziom kompetencji zespołowych wśród uczestników był duży, o tyle poziom doświadczeń projektowych mały. Stąd też uznano za konieczne dopasowanie odpowiednich dobrych praktyk zarządzania projektami

w zależności od poziomów dojrzałości zespołu. W tym celu pozyskano dobre praktyki zarządzania projektami według PMBoK, zalecając zespołom ocenę zadań z perspektywy wejścia, przetwarzania i wyjścia. Postawiono poddać modyfikacji tę dobrą praktykę o zalecenia metody RUP dotyczące definiowania zadań razem z oczekiwanymi produktami pracy. Takie podejście pozwoliło zespołom na realizację zadań w oparciu o przygotowany wcześniej plan działania. Zespoły definiowały razem z agentem koordynatorem zadania, określały, jakich produktów oczekuje się po ich zakończeniu, i dzięki temu zadania powierzone zespołom zostały wykonywane terminowo. Po kilku tygodniach okazało się, że zachodzi konieczność gromadzenia wyników badań na potrzeby późniejszych publikacji tych wyników. Stąd też przygotowano odpowiedni dokument (zgodnie z zaleceniami podejść ciężkich) zawierający zapis doświadczeń wszystkich uczestników (agentów) realizujących prace.

Przy kolejnej weryfikacji postępów w ramach zespołów postanowiono zaadaptować dobrą praktykę pochodzącą z metody XP polegającą na pracy parami. Każdemu uczestnikowi z wiedzą zarządczą, ale o mniejszym doświadczeniu w aspektach technicznych, przydzielono uczestnika o dużej wiedzy technicznej (ale mniejszym doświadczeniu w zarządzaniu). Dzięki takiemu podejściu uzyskano efekt wspólnego pozyskiwania wiedzy i możliwa stała się kontynuacja realizacji zadań. Projekt zakończył się w wyznaczonym harmonogramie oraz zakresie (w projekcie nie przewidziano budżetu, prace były wykonywane w ramach bezpłatnych praktyk).

Wnioski

Przeprowadzone badanie pozwoliło na zweryfikowanie relacji pomiędzy agentami przy realizacji projektu. Umożliwiło identyfikację i sprecyzowanie relacji na poziomie zarządzania modelem szczegółowego a2M oraz warstw wiedzy, przetwarzania i funkcjonalności. Wskazało mechanizmy współpracy agentów. Podczas analizy relacji agentów oceniano procesy adaptacji tych relacji w oparciu o wiedzę i przetwarzanie tej wiedzy o relacjach zespołu projektowego. Wnioski z badania 7-R przedstawiono w tabeli 5.15.

Tabela 5.15 Wnioski z badania 7-R

		Kod badania: 7-R	
Realizacja badania	celu	Cel badania osiągnięty	
Zidentyfikowane poziomy model szczegółowego a2M		definiowania zarządzania adaptacji	Istnieją procesy adaptacji struktur agentowych
Zidentyfikowane warstwy modelu szczegółowego a2M		wiedzy przetwarzania funkcjonalności	Istnieją procesy adaptacji struktur agentowych

Źródło: opracowanie własne

Na podstawie przeprowadzonych badań stwierdzono, że możliwe jest adaptowanie dobrych praktyk zarządzania projektami informatycznymi w zależności od zmiennych decyzyjnych nowego trójkąta ograniczeń. Zaobserwowano jednocześnie, że adaptowanie dobrych praktyk przynosi spodziewane korzyści tylko przy znanym poziomie dojrzałości klienta, zespołu dostawcy oraz entropii projektowej. Uznano również na podstawie badania, że adaptowanie dobrych praktyk z różnych metod (jak założono w modelu szczegółowego a2M) jest nie tylko możliwe, ale wręcz konieczne dla poprawnego przebiegu projektu informatycznego.

W sytuacji, kiedy poziomy dojrzałości są zróżnicowane (dojrzały klient, niedojrzały zespół dostawcy), należy dopasowywać dobre praktyki z różnych metod (zarówno lekkich jak i ciężkich). Dzięki takiemu podejściu zastosowanemu w eksperymencie udało się zrealizować omawiany projekt. Warto wspomnieć, że trwają przygotowania do kontynuacji tego projektu.

Należy zaznaczyć także, że podczas rozmów z niektórymi uczestnikami zwrócono uwagę, że jakość pracy poprawiła się znacznie dopiero po zastosowaniu dobrej praktyki związanej z pracą w parach. Jest to bardzo ważna wskazówka dla kierowników projektów, aby w zespołach o mniejszym doświadczeniu próbować integrować uczestników poprzez właśnie takie działanie. Dzięki temu uczestnicy mają więcej pewności co do tego, co robią (jest druga osoba, współodpowiedzialna), a jednocześnie wzajemnie się uczą. To pokazało również, że stosowanie założeń modelu szczegółowego **a2M** czy adaptowanie dobrych praktyk zarządzania projektami informatycznymi w odniesieniu do wartości zmiennych decyzyjnych (dojrzałość klienta, zespołu, entropia) wpływa na jakość pracy uczestników projektu. Tym samym buduje relacje między nimi i poprawia poziom tych relacji.

Warte podkreślenia jest również to, że dobre praktyki były adaptowane wraz z rozwojem projektu. Kiedy bowiem okazywało się, że zespół nie radzi sobie z zadaniami, stosowano pracę w parach, która pozwoliła na wznowienie prac. Jest to kolejna cenna wskazówka dla kierowników projektu, żeby dobre praktyki adaptować nie tylko na początku projektu, ale również w trakcie jego realizacji. Aby jednak było to możliwe, konieczne wydaje się stosowanie podejścia agentowego. Tylko bowiem na podstawie komunikatów przesyłanych pomiędzy agentami (uczestnikami) możliwe jest zauważenie potrzeby podjęcia nowych decyzji odnośnie adaptacji kolejnych dobrych praktyk.

Takie podejście też wydaje się wносить nowe spojrzenie na zarządzanie projektami informatycznymi. O ile bowiem zakładano do tej pory, że kierownik projektu wybiera metodę zarządzania projektami i dostosowuje prowadzenie prac do jej zasad, o tyle w podejściu agentowym mamy działanie w drugą stronę. To w zależności od rozwoju sytuacji dopasowuje się dobre praktyki. Przy stosowaniu metod zarządzania projektami niemożliwe jest zmienianie zasad projektu (uczestnicy dostosowują się do z góry narzuconych reguł), o tyle w podejściu agentowym to zasady realizacji dostosowuje się do warunków. Następuje więc adaptacja dobrych praktyk do ograniczeń tego projektu.

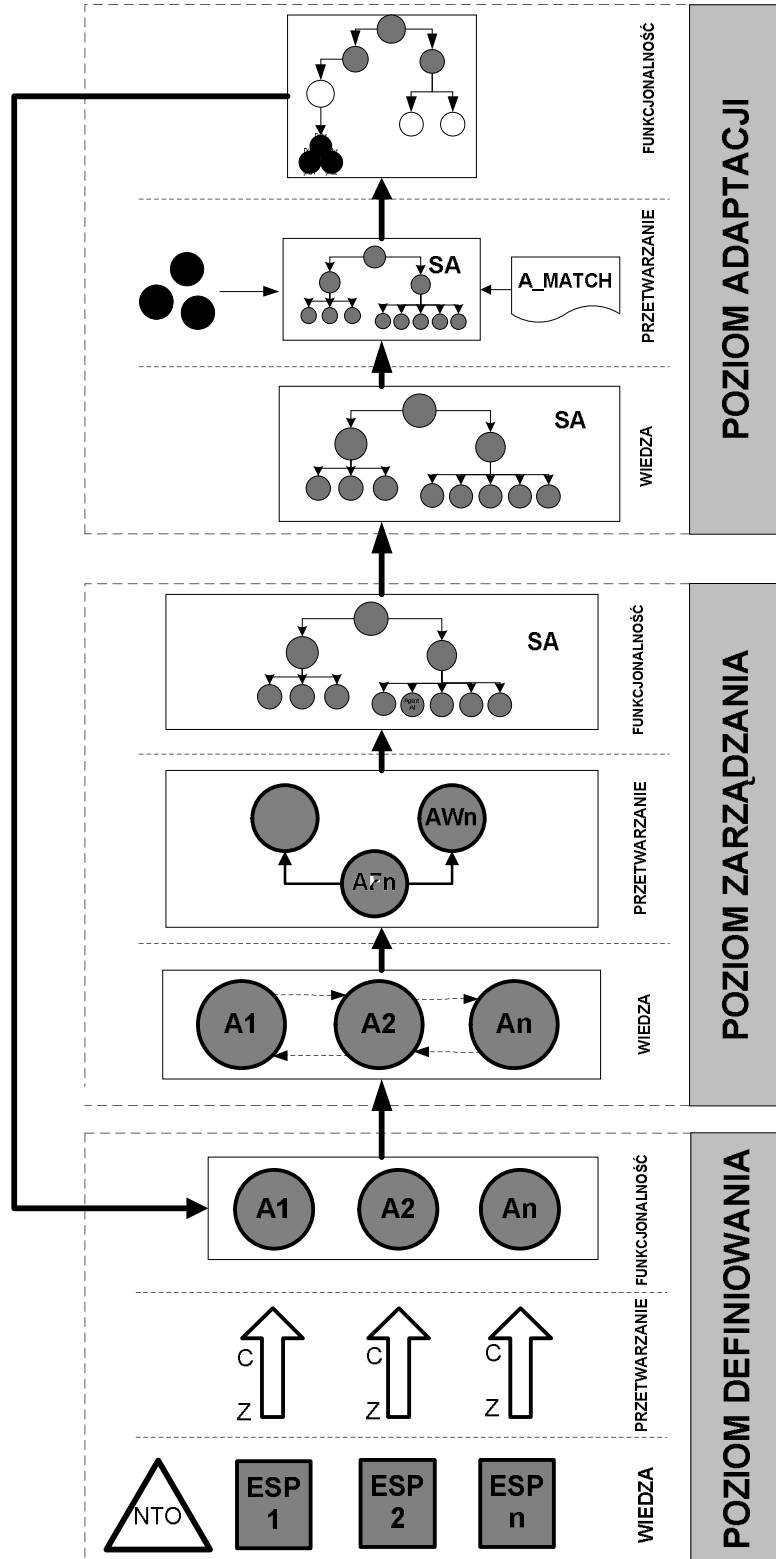
Przeprowadzone badanie pozwoliło również na wskazanie pola do modyfikacji modelu szczegółowego **a2M**. Zastanawiano się (bazując na doświadczeniach z tego projektu), w jaki sposób kierownik projektu powinien dobierać dobre praktyki według jakich zasad. W omawianym projekcie wykorzystano doświadczenia i wiedzę agenta koordynatora. Rozważano opracowanie formalnego rozwiązania pozwalającego na adaptowanie dobrych praktyk. Uznano za konieczne sprawdzenie także możliwości sformalizowania adaptacji dobrych praktyk. Formalizacja adaptowania dobrych praktyk powinna zostać oparta na odpowiedniej funkcji (takiej jak proponowana w rozdziale II funkcja **A_MATCH**).

5.2 Podsumowanie — konstrukcja modelu po procesach badania zasadności replikatywnej

Na podstawie wniosków płynących z przeprowadzonych badań postanowiono zmodyfikować konstrukcję opracowanego modelu.

Po pierwsze uznano, że trójkąt ograniczeń projektowych powinien być określany nie tylko na najwyższym poziomie, ale właściwie na każdym poziomie modelu szczegółowego **a2M**. Pozwoli to na lepsze dopasowywanie (adaptację) agentów. Może bowiem okazać się, że jeden z agentów zarządzających (np. agent koordynator) powoływany do sprawowania funkcji nadzoru nad agentami funkcjonalnymi (co założono na drugim poziomie modelu) posiada cechy osobowościowe, które nie sprzyjają powierzeniu mu funkcji koordynatora. Gdyby istniała możliwość pomiaru dojrzałości

pojedynczych agentów już na poziomie definiowania, można by uniknąć przypisania niewłaściwych funkcji niewłaściwym agentom. Dokonywanie pomiaru zmiennych decyzyjnych także na poziomie zarządzania modelu szczegółowego **a2M** powinno pozwolić na modyfikację przyjętej struktury (hierarchii) między agentami. Ze względu np. na cechy uczestników projektów związane z pełnieniem funkcji w zespole struktura może zostać zmodyfikowana (relacje hierarchiczne zmienione), a tym samym usprawniony będzie przepływ komunikatów (zleceń).



Rys. 5.1 Konstrukcja modelu po badaniu zasadności replikatywnej

Źródło: opracowanie własne

Jak wynika z rysunku 5.1 nowy trójkąt ograniczeń został uwzględniony już na poziomie definiowania. Dzięki temu możliwe jest dokonywanie pomiaru zmiennych decyzyjnych już w najniższej warstwie przez co wszystkie procesy zachodzące w pozostałych warstwach poszczególnych poziomów odbywają się z uwzględnieniem nowego trójkąta ograniczeń.

Kolejnym wnioskiem wypływającym z przeprowadzonych badań jest wskazanie, że proces adaptacji powinien być rozszerzony i dotyczyć nie tylko dobrych praktyk. Procesy adaptacji powinny dotyczyć wszystkich elementów struktury projektowej. Struktura zatem powinna ulegać zmianie wraz ze zmianami w projekcie (uwzględniono to na rysunku w postaci pętli sprzężenia zwrotnego).

Trzecim istotnym wnioskiem jest nieodzowna rola koordynatora, który pojawiał się niemal w każdym projekcie jako osoba uzupełniająca (wspomagająca) kierownika projektu. Rodzi się więc pytanie, czy taki koordynator nie powinien pełnić funkcji równorzędnej kierownikowi projektu (agentowi menedżerowi). Oznacza to, że w projekcie należy definiować przynajmniej dwa agenty nadrzędne, które kontrolują realizację zadań przez pozostałe agenty. Nie jest to zgodne z typową dla zarządzania hierarchiczną strukturą organizacyjną, jednak z punktu widzenia przesyłania komunikatów w systemie agentowym dawałoby określone korzyści.

Na podstawie przeprowadzonych badań (1-R do 7-R) zasadności replikatywnej modelu oraz podanych powyżej sugestii dotyczących zmian w modelu na tym etapie badań dokonano trzech najważniejszych (integrujących podane zmiany) modyfikacji. Modyfikacje te stają się niezbędne przed dalszym wykorzystywaniem modelu szczegółowego **a2M**. Zmiany te obejmują:

1. Wprowadzenie nowego trójkąta ograniczeń na każdy poziom modelu szczegółowego **a2M**. Dobór członków zespołu (poziom definiowania), określanie relacji zarządczych (poziom zarządzania) oraz dobór dobrych praktyk (poziom adaptacji) powinny być analizowane w oparciu o opracowany w tej pracy nowy trójkąt ograniczeń.
2. Procesy zarządcze zarówno na płaszczyźnie zarządzania jak i adaptacji powinny mieć charakter dynamiczny (iteracyjny), a nie statyczny (jednorazowy). Dobór członków zespołu/agentów modelu **a2M** ma zwykle charakter przyrostowy. Takie podejście potwierdziły przeprowadzone eksperymenty.
3. W obecnej wersji modelu szczegółowego **a2M** nie zdefiniowano procesów adaptacji. Stwierdza się tylko, że istnieją i identyfikuje się je na każdej z warstw i na każdym z poziomów. Na rys. 5.1 przedstawiono zmodyfikowaną wersję modelu **a2M** z naniesionymi procesami adaptacyjnymi o charakterze wstępnym.

Inne sugerowane zmiany jak: formalizacja doświadczeń, wprowadzenie mechanizmu pozyskiwania dobrych praktyk będą brane pod uwagę w kolejnych wersjach modelu szczegółowego **a2M**.

Przeprowadzone badanie zasadności replikatywnej (siedem badań dotyczących różnych co do skali i znaczenia projektów informatycznych) wykazało, że opracowany model szczegółowy **a2M** odpowiada rzeczywistości projektowej, porządkuje procesy zarządcze i usprawnia ich przebieg.

Pierwsze trzy badania omówione w rozdziale V pozwoliły na wykazanie konieczności pełnego definiowania wszystkich elementów struktury projektowej. Wykazano, że w rzeczywistych projektach informatycznych kierownicy projektów zaczynają wykonywać swoje zadania począwszy od wyodrębnienia niezbędnych elementów właśnie struktury projektowej.

Trzy kolejne badania miały na celu wskazanie konieczności tworzenia struktur projektowych w projektach informatycznych. Wykazano, że łączenie agentów w struktury hierarchiczne pozwala na lepszą organizację przepływów komunikatów między nimi. Uznano również, że relacje między agentami mają największe znaczenie dla późniejszej realizacji zadań. Podczas przeprowadzania badań potwierdzono konieczność tworzenia relacji hierarchicznych. Zwrócono jednak uwagę również na relacje nieformalne, jakie zawiązują się między uczestnikami projektu. Tych relacji nie udało się poddać procesom modelowania i przedstawić w modelu szczegółowego **a2M**.

Ostatnie z prowadzonych badań poświęcone zostało wykazaniu słuszności mierzenia zmiennych decyzyjnych nowego trójkąta ograniczeń oraz możliwości dobierania dobrych praktyk w zależności od wartości tych zmiennych. Przeprowadzone badania wykazały wpływ zmiennych nowego trójkąta ograniczeń na realizację prac projektowych. Okazało się, że dobierając dobre praktyki w zależności od nowego trójkąta ograniczeń, w obserwowanych projektach realizowano zadania terminowo, a wykonawcy zadań nie zgłaszali żadnych uwag do organizacji pracy.

Rozdział VI

Badania zasadności predykcyjnej modelu

Dokonana w poprzednim rozdziale weryfikacja zasadności replikatywnej modelu pokazała na ile opracowany model szczegółowy **a2M** odzwierciedla realia projektowe i na ile usprawnia procesy zarządzania. W ramach badania zasadności replikatywnej przeprowadzono siedem badań zakończonych projektów informatycznych, weryfikując w oparciu o dane tych projektów przydatność opracowanego modelu szczegółowego **a2M**.

Obecnie przystępujemy do badania zasadności predykcyjnej weryfikując opracowany model szczegółowy **a2M** w środowisku bieżących projektów informatycznych badając wpływ modelu na procesy zarządzania projektami. Badanie zasadności predykcyjnej modelu prezentowane w tym rozdziale kończy procesy weryfikacji hipotezy badawczej.

Rozdział ten zawiera pięć przeprowadzonych w oparciu o realizowane projekty eksperymentów badawczych. Dwa z nich są odniesione do konstrukcji zespołów projektowych natomiast trzy do implementacji modelu szczegółowego **a2M** w technologiach informatycznych. Odzwierciedlony zostanie w ten sposób adaptacyjny informacyjno- implementacyjny charakter modelu szczegółowego **a2M**. W przypadku konstrukcji zespołów projektowych model szczegółowy **a2M** został zastosowany w środowisku dwu projektów informatycznych realizowanych we współpracy z dużymi firmami informatycznymi. Model szczegółowy **a2M** został potraktowany jako szablon projektu i w oparciu o jego strukturę powstała struktura zespołu projektowego.

W drugim przypadku implementacji modelu szczegółowego **a2M** uznano za celowe zbadanie czy opracowany model może stanowić kompletną architekturę niezbędną do implementacji w technologiach informatycznych wspomagających prace menedżera projektu informatycznego.

6.1 Procesy badania zasadności predykcyjnej modelu

Podczas badania zasadności predykcyjnej modelu postanowiono skoncentrować się na trzech podstawowych procesach. Pierwszy proces dotyczy oceny przydatności modelu do budowy zespołu projektowego i realizowany jest w oparciu o dwa eksperymenty (eksperymenty 1-P i 2-P). Drugi dotyczy możliwości implementacji modelu szczegółowego **a2M** w wybranym środowisku wspomagającym zarządzanie projektami informatycznymi (eksperymenty 3-P i 4-P). W trakcie eksperymentu podjęto decyzję, aby do implementacji założeń modelu wykorzystać narzędzie z rodziny Rational, jakim jest IBM Rational Team Concert. Narzędzie to jest przedmiotem badań i rozwoju w Uniwersyteckim Centrum Kompetencyjnym Technologii Rational powstałym na Politechnice Gdańskiej przy współpracy z firmą IBM.

Dostępność narzędzia oraz dotychczasowe prace, jak również obecność wielu specjalistów znających dobrze to narzędzie pozwoliły na wybór właśnie środowiska Rational. Uznano jednocześnie, że do implementacji założeń modelu szczegółowego **a2M** zostanie powołana jedna osoba (specjalista RTC – mgr inż. Paweł Madej), która po otrzymaniu specyfikacji odnośnie modelu szczegółowego **a2M** przeprowadzi procesy implementacji.

Jednocześnie założono, że badając zasadność predykcyjną modelu należy poddać implementacji wszystkie poziomy i warstwy modelu. Nie weryfikowano każdej z warstw modelu, jak było to w przypadku badań zasadności replikatywnej (rozdział piąty). Takie podejście jest konsekwencją faktu, że implementując wybrany poziom modelu szczegółowego **a2M** z wykorzystaniem RTC konieczne jest implementowanie trzech warstw (danych, przetwarzania i funkcjonalności) jednocześnie.

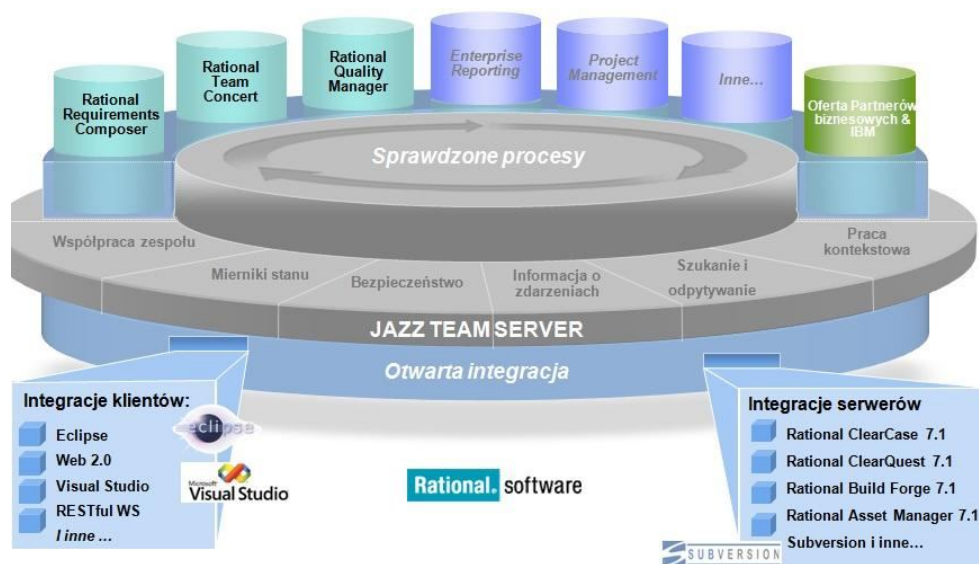
Trzecim procesem badania zasadności predykcijnej jest zdolność przystosowania modelu do oczekiwań kierownika, czyli zapewnienie możliwości stosowania modelu szczegółowego **a2M** jako alternatywy dla formalnych metod zarządzania projektami informatycznymi (eksperyment 5-P). W ramach eksperymentu 5-P, poddano implementacji model szczegółowy **a2M** w całości, wszystkie jego poziomy. Dodano także procedurę postępowania dla kierownika projektu (zestaw kroków niezbędnych do wykonania przy wykorzystywaniu modelu szczegółowego **a2M**). Na zakończenie eksperymentu 5-P przeprowadzono sesję kierownika projektu z systemem podczas której sprawdzi on na ile korzystanie z pełnej organizacyjno-implmentacyjnej wersji modelu szczegółowego **a2M** jest możliwe.

6.1.1 Środowisko badań i eksperymenty badawcze

Do przeprowadzenia badań predykcijnych modelu wybrano dwa środowiska zarządzania projektami. Dla eksperymentów 1-P i 2-P środowiskiem badań będzie 9 zespołów o różnej liczebności członków. Każdy z zespołów przygotowuje plan realizacji projektu informatycznego będący podstawą do późniejszej realizacji takiego projektu.

W przypadku eksperymentów 3-P, 4-P i 5-P środowiskiem projektu będzie narzędzie do zarządzania projektami informatycznymi - IBM Rational Team Concert (RTC). Jest to narzędzie, powstałe z inicjatywy firmy IBM, wspomagające pracę zespołową pozwalające na integrację zadań na przestrzeni całego cyklu życia oprogramowania.

Narzędzie to powstało w oparciu o platformę Jazz, służącą do wspomagania wytwarzania oprogramowania. Platforma Jazz wspiera nie tylko twórców oprogramowania, ale również kierowników projektów tworząc warunki organizacji pracy i gromadzenia wiedzy.



Rys. 6.1 Platforma Jazz służąca do wspomagania wytwarzania oprogramowania

Źródło: [103]

Architektura RTC to architektura dwuwarstwowa składająca z następujących składowych:
 - klienta (umożliwiająca korzystanie zarówno z poziomu klienta grubego jak i cienkiego),
 -serwera (serwerem tej aplikacji jest TOMCAT lub WebSphere).

RTC posiada szereg funkcjonalności pozwalających na usprawnienie procesu zarządzania projektami informatycznymi. Jest to przede wszystkim możliwość dostosowania procesów do potrzeb realizowanego projektu. W trakcie realizacji projektu w (ramach eksperymentów tej pracy) RTC wspomaga pracę zespołu poprzez opcję automatycznego tworzenia zadań i możliwość śledzenia postępu prac zgodnie z regułami eksperymentów. Narzędzie to pozwala również na analizę obciążenia poszczególnych członków zespołu zadaniami oraz sprawniejsze i skuteczniejsze zarządzanie dostępnymi zasobami. W RTC istnieje możliwość definiowania zespołu projektowego, jego członków oraz przypisania zadań do każdego z nich. Istotną zaletą tej technologii jest wspomaganie komunikacji pomiędzy członkami zespołu w ramach realizacji prac projektowych. Takie podejście jest możliwe poprzez wybór jednego z możliwych komunikatorów (Jabber lub IBM Lotus Sametime).

Po prezentacji środowiska kolejno zostaną zaprezentowane eksperymenty badawcze. Eksperymenty te będą przedstawione w podobny sposób jak badania w rozdziale piątym. W każdym eksperymencie określono: cel eksperymentu, założenia dla jego przeprowadzenia oraz środowisko badań. Wynikiem eksperymentów będzie identyfikacja warstw i poziomów modelu szczegółowego a2M oraz możliwości ich implementacji w technologiach do zarządzania projektami.

6.1.2 Eksperyment 1-P. Wsparcie procesów definiowania agentów (poziom definiowania)

Na wstępie przedstawiono środowisko eksperymentu. Do eksperymentu zaproszono przedstawicieli różnych dziedzin w celu przygotowania planu realizacji projektów informatycznych. Postanowiono utworzyć 9 zespołów o różnej liczebności członków. Każdy z zespołów miał przygotować plan projektu informatycznego będący podstawą do późniejszej realizacji takiego projektu. Tematy projektów wraz z opisem członków zespołów zostały przedstawione w załączniku. Eksperyment przeprowadzono 11 kwietnia 2011 roku w ramach konferencji ITAKM 2011 w Juracie. W tabeli 6.1 przedstawiono założenia do przeprowadzania badania.

Tabela 6.1 Założenia do badania 1-P

Kod eksperymentu 1-P		
Cel	Celem eksperymentu jest sprawdzenie zasadności pomiaru poziomów dojrzałości zespołów, entropii projektu i dojrzałości klienta. Celem eksperymentu jest również ocena, w jakim stopniu nowy trójkąt ograniczeń wpływa na pracę zespołu projektowego złożonego z agentów	
Środowisko badania	Za środowisko badania przyjęto 9 zespołów projektowych pracujących równolegle. Powołano zespoły interdyscyplinarne przygotowujące plan realizacji określonych projektów.	
Poziomy modelu a2M do identyfikacji	do potwierdzenia – poziom adaptacji inne poziomy?	Procesy adaptacji?
Warstwa modelu a2M do identyfikacji	do potwierdzenia – warstwa wiedzy inne warstwy?	Procesy adaptacji?
Początek realizacji:	Kwiecień 2011	ukończono
Termin zakończenia:	Kwiecień 201	ukończono

Źródło: opracowanie własne

Opis przebiegu eksperymentu

W ramach eksperymentu narzucono wszystkim zespołom dobre praktyki zarządzania projektami. Nie stosowano konkretnej metody zarządzania projektem (np. RUP czy SCRUM) ale zaczerpnięto jedynie wybrane dobre praktyki. Następnie zaznaczono, że dobre praktyki odnoszą się do wszystkich zespołów w podobny sposób. To znaczy, że zespołom przekazano zrealizowanie zadań koncepcyjnych (związanych z planowaniem projektów) w określonych krótkich ramach czasowych (sprintach). Następnie weryfikowano ich pracę (w oparciu o przygotowane przez zespoły dokumenty) a po weryfikacji prac realizowano kolejne sprinty.

Ekspertyment przebiegał według następujących zasad:

- SPRINT #1 godz. 9.00-11.00
 - Runda 1 – zdefiniowanie celu ogólnego projektu (30 min)
 - Runda 2 – zdefiniowanie celów szczegółowych (30 min)
 - Runda 3 – sprawdzenie barier prawnych (30 min)
 - Runda 4 – ogólna koncepcja procesu realizacji projektu - zadań (30 min)

- SPRINT #2 11.30-13.30
 - Runda 5 – określenie zadań i przydział odpowiedzialności (30 min)
 - Runda 6 – określenie wstępnego harmonogramu prac (30 min)
 - Runda 7 – oszacowanie budżetu (30 min)
 - Runda 8 – weryfikacja prawna (30 min)

Jak wynika z przedstawionego planu eksperymentu, zastosowano dla wszystkich badanych projektów dobre praktyki odnośnie tworzenia celu i wizji projektu oraz podziału pracy i przepływów tej pracy. Weryfikacja pracy poszczególnych zespołów następowała po zakończeniu każdej rundy w ramach sprintów w oparciu o dane zawarte w arkuszu obserwacyjnym.

Należy podkreślić, że zespoły były zróżnicowane doświadczeniem projektowym, to znaczy, że składały się zarówno z osób pracujących jako kierownicy projektów informatycznych, jak również z osób reprezentujących podmioty reprezentujące potencjalnych użytkowników końcowych lub wspomagających późniejsze zastosowanie produktów projektu (np. osoby z jednostek administracji państwowej). W eksperymencie zespoły przygotowywały plany projektów informatycznych, które miały być współfinansowane ze źródeł zewnętrznych (fundusze strukturalne UE). Istniało również zapotrzebowanie na specjalistyczną wiedzę prawną. Stąd też do eksperymentu zaproszono przedstawicieli o dużej wiedzy z zakresu prawa UE, aby mogły wspierać zespoły przy planowaniu określonych etapów projektu.

Należy wspomnieć, że wszyscy uczestnicy eksperymentu reprezentowali zarówno różne instytucje jak również różne regiony kraju. Okazało się, że po zaplanowaniu projektów, realizacja tych projektów odbywać się będzie w trybie rozproszonym. Taka sytuacja dodatkowo pozwalała traktować wszystkich uczestników, jako agenty, które będą realizować zadania tylko w określonych sytuacjach. Takie podejście spowodowało konieczność powołania odpowiedniego agenta koordynatora (autor niniejszej pracy), który nie tylko kierował pracą zespołów podczas eksperymentu. Odpowiadał także za monitorowanie późniejszej realizacji projektów zaplanowanych podczas eksperymentu.

Zgodnie z celem eksperymentu, należało ocenić dojrzałości tych zespołów oraz entropię proponowanych do realizacji projektów. Dojrzałość klienta uznano w tym wypadku za stałą, zwłaszcza, że projekty miały być współfinansowane ze źródeł zewnętrznych, co oznacza precyzyjne zdefiniowanie wymagań w stosunku do projektu. Skoro dojrzałość klienta jest znana zwrócono szczególną uwagę na dojrzałości zespołów oraz oszacowano entropię projektu. Jedynym pomiarem koniecznym do przeprowadzenia był pomiar dojrzałości zespołów. Do tego celu użyto kwestionariusza przygotowanego na potrzeby niniejszej rozprawy (prezentowany w załączniku).

Po pomiarze dojrzałości poszczególnych zespołów, przystąpiono do realizacji prac zgodnie z przyjętym planem eksperymentu. Wszystkie zespoły rozpoczęły pracę w tym samym czasie. Rolą

koordynatora projektów było sprawdzenie, czy zakładane produkty pośrednie (jak cel, wizja projektu itp.) zostaną zrealizowane zgodnie z harmonogramem. Zgodnie z przewidywaniami, zespoły o niższym poziomie dojrzałości albo zespoły, w których entropia proponowanego do realizacji projektu była duża, wykonywały zadania w dłuższym horyzoncie czasowym niż zespoły o wyższej dojrzałości przygotowujących projekty o niskiej entropii. Wyniki przeprowadzonych ocen zestawiono w tabeli 6.2

Tabela 6.2 Wyniki ocen dojrzałości zespołów projektowych

NR ZESPOŁU TEMAT PROJEKTU	POZIOM DOJRZAŁOŚCI	POZIOM ENTROPII	WYNIKI OBSERWACJI PODZAS EKSPERYMENTU
#1. Budowa centrów kompetencyjnych firm informatycznych dla wsparcia procesów dydaktyki i biznesu (uczelnie i biznes),	3 (definiowalny)	ŚREDNI	W pierwszym sprincie zespół realizował zadania w terminie, udało się zrealizować cel ogólny oraz cele szczegółowe.
#2. Wspólne przedsięwzięcie IBM i PG - "wsparcie cyklu wytwarzania oprogramowania poprzez integrację narzędzi informatycznych IBM"	3 (definiowalny)	ŚREDNI	Zadania w pierwszym sprincie zrealizowano bez większych problemów. Określono cele projektu i wstępny podział zadań. W drugim sprincie zaobserwowano problemy z określeniem harmonogramu i budżetu
#3. Rozbudowa Centrów Kompetencyjnych w oparciu o program Smarter Planet i zaproszenie do współpracy i realizacji wspólnego projektu partnerów z Litwy i Obwodu Kaliningradzkiego	3 (definiowalny)	ŚREDNI	Zespół wykonywał zadania terminowo w pierwszym sprincie. W drugim napotkał na problemy z określeniem ram czasowych projektu
#4. Biznesowe wsparcie Centrum Kompetencyjnego PG dla wdrażania środowiska RTC dla potrzeb partnerów zewnętrznych.	3 (definiowalny)	ŚREDNI	Zadania pierwszego sprintu wykonano terminowo. Zadania w drugim sprincie realizowano z małymi opóźnieniami.
#5. Oprogramowanie Service Manager-wsparcie procesów budowy organizacji informatycznych architekturą SOA/korporacyjną/ITIL	4 (ilościowo zarządzany)	ŚREDNI	Zespół realizował zadania w terminie, nie odnotowano żadnych problemów zarówno w pierwszym jak i drugim sprincie.
#6. Budowa Data Center — dobór i monitorowanie zapotrzebowania na zasoby „chmury”; analiza kosztów outsourcingu przetwarzania danych.	4 (ilościowo zarządzany)	DUŻY	Zauważono opóźnienia w definiowaniu celu projektu (najprawdopodobniej z powodu wysokiej entropii proponowanego do realizacji projektu). Uczestnicy zwracali uwagę na zapotrzebowanie wsparcia przez eksperta w objaśnieniu przepisów prawnych.
#7. Prognozowanie stanu (dojrzałości) organizacji informatycznej IT w oparciu o systemy oparte na wiedzy	4 (ilościowo zarządzany)	ŚREDNI	Wykonano zadania z pierwszego sprintu terminowo. Drugi sprint nie został zrealizowany podczas trwania eksperymentu z powodu wyjazdu członków zespołu

#8. MS Dynamics AX 2009 — centrum kompetencyjne Microsoft i rozwój dydaktyki; wykorzystanie usług systemu dla tworzenia rozwiązań dla sektora MSP	2 (zarządzany)	DUŻY	Prace przebiegały ze sporymi opóźnieniami. Po rozpoczęciu drugiego sprintu w ramach eksperymentu, zespół opracowywał zadania ze sprintu pierwszego.
#9. Semantyczny portal wiedzy środowiskiem wsparcia procesów integracji uczelni i biznesu	4 (ilościowo zarządzany)	NISKI	Zespół pracował bardzo wydajnie. Zarówno w pierwszym jak i drugim sprincie wykonano wszystkie przewidziane zadania.

Źródło: opracowanie własne

Wnioski

Przeprowadzony eksperyment pozwolił na ocenę możliwości wsparcia zespołów projektowych modelem szczegółowym **a2M**. Umożliwił identyfikację członków zespołu (jagentów), pozwolił na prezentację relacji między nimi. W eksperymencie badawczym poddano weryfikacji poziom definiowania, zarządzania i adaptacji oraz warstwy wiedzy i przetwarzania modelu szczegółowego **a2M**. Podczas budowy zespołu poddano weryfikacji procesy adaptacji struktur agendowych dla potrzeb budowy zespołu projektowego. Wnioski z badania 1-P przedstawiono w tabeli 6.3

Tabela 6.3 Wnioski z badania 1-P

		Kod eksperymentu: 1-P	
Realizacja badania	celu	Cel badania osiągnięty	
Zidentyfikowane poziomy modelu szczegółowego a2M		Definiowania zarządzania adaptacji	Istnieją procesy adaptacji struktur agentowych
Zidentyfikowane warstwy modelu szczegółowego a2M		wiedzy przetwarzania	Istnieją procesy adaptacji struktur agentowych

Źródło: opracowanie własne

Podstawowym wnioskiem z przeprowadzonego eksperymentu jest potwierdzenie przyjętego w modelu szczegółowego **a2M** założenia, że konieczne jest analizowanie poziomu dojrzałości klienta, dojrzałości zespołu oraz entropii przed podjęciem decyzji dotyczącej doboru dobrych praktyk. Na podstawie przeprowadzonego eksperymentu można zauważyć, że zespoły o wyższym poziomie dojrzałości realizowały zadania skuteczniej. Można również zauważyć, że liderzy zespołów o dużej dojrzałości podejmowali trafniejsze decyzje w stosunku do projektów o dużej entropii. Wnioski z obserwacji każdego z zespołów zestawiono w tabeli 6.2.

Przeprowadzony eksperyment potwierdził, że dojrzałość zespołu projektowego oraz entropia projektowa są istotne z punktu widzenia realizacji projektów. Tym samym kierownicy projektów powinni nie tylko dokonywać pomiaru tych zmiennych, ale w oparciu o pomiary podejmować określone decyzje. Konieczność pomiaru zmiennych decyzyjnych (entropii, dojrzałości organizacji i klienta) pozwoliła na zwrócenie uwagi na potrzebę regularnego dokonywania pomiarów tych zmiennych. Warto zatem w projektach informatycznych wyodrębnić dedykowanego agenta odpowiedzialnego za takie pomiary. Zgodnie z przyjętą koncepcją modelu szczegółowego **a2M** rolę agenta odpowiedzialnego za pomiary dojrzałości powinien pełnić albo kierownik projektu (agent

menedżer) albo przy projektach bardziej złożonych, dedykowany agent dojrzałości lub agent entropii (osoba o cechach analityka biznesowego)

Ważnym spostrzeżeniem będącym efektem przeprowadzonego eksperymentu, jest również to, że posiadając takiego agenta dojrzałości można dokonywać pomiarów dojrzałości częściej niż tylko na początku projektu. Ponieważ zespół wraz z rozwojem projektu zmienia swą dojrzałość takie pomiary są niezbędne. Jest to o tyle ważny wniosek, gdyż wskazuje, że struktura projektu informatycznego (wszystkie podmioty) zmienia się z czasem. Tym samym zmienia się zapotrzebowanie na określone dobre praktyki w zależności od zmian zespołu. Oznacza to zatem, że struktura projektowa w projektach informatycznych ma charakter dynamiczny. W załączniku 4 przedstawiono arkusze obserwacyjne które były wykorzystane do odnotowywania podanych uwag.

Przeprowadzony eksperyment pozwolił jednocześnie zaobserwować, że istnieje potrzeba dobierania dobrych praktyk zarządzania projektami w zależności od zmieniających się ograniczeń wynikających ze zmian dojrzałości. Nie można sugerować dobrych praktyk bez znajomości dojrzałości zespołów, trzeba je adaptować do zmieniających się uwarunkowań projektu.

6.1.3 Eksperyment 2-P. Wpływ modelu na proces realizacji rzeczywistego projektu

Drugi z eksperymentów w ramach, którego badana jest zasadność replikatywna modelu szczegółowego **a2M**. Badane jest wykorzystanie wszystkich poziomów i warstw modelu szczegółowego we wsparciu procesu zarządzania projektem informatycznym. Taki układ eksperymentu oznacza, że na podstawie wprowadzonego w pracy trójkąta ograniczeń – dojrzałości klienta, dojrzałości zespołu dostawcy oraz entropii projektu oraz po wyodrębnieniu wszystkich agentów struktury projektowej należy dobrać dobre praktyki zarządzania projektem. Do tego eksperymentu wybrano jeden z 9 projektów, które były obserwowane podczas realizacji eksperymentu 1-P. Z dziewięciu zespołów przygotowujących plan projektu, wybrano zespół o wysokim poziomie dojrzałości (poziom zespołu określono jako ilościowo zarządzany), który realizował projekt o najwyższej entropii (zespół nr 7 z eksperymentu 1-P).

Zakres prac tego zespołu objętych eksperymentem dotyczył przygotowania studium wykonalności budowy centrum przetwarzania danych oraz powołania operatora takiego centrum. Prace objęte eksperymentem trwały przez okres pięciu miesięcy, od kwietnia 2011 do września 2011. Koordynatorem projektu był Zakład Zarządzania Technologiami Informatycznymi Politechniki Gdańskiej, dostawcą rozwiązań potrzebnych dla budowy takiego centrum przetwarzania danych firma HP. Ponadto planowano aby operatorem centrum przetwarzania danych powstałego jako efekt projektu było konsorcjum złożone z kilku trójmiejskich firm kooperujących (m.in. INFOZIT, SINERSIO Polska sp. z o.o., MEDIUM SOFT Polska Sp.z o.o). Projekt oszacowano wstępnie na 10 000 000 PLN. Część budżetu miała być współfinansowana z Programu Operacyjnego Innowacyjna Gospodarka w ramach programu Polskiej Agencji Rozwoju Przedsiębiorczości. W tabeli 6.4 przedstawiono założenia do przeprowadzania badania.

Tabela 6.4 Założenia do badania 2-P

Kod eksperymentu 2-P	
Cel	Celem eksperymentu jest sprawdzenie wpływu zastosowania modelu szczegółowego a2M na proces realizacji rzeczywistego projektu.
Środowisko badania	Środowiskiem weryfikacji modelu szczegółowego a2M jest projekt informatyczny związany z tworzeniem i zarządzaniem centrum przetwarzania danych. Projekt jest realizowany przy współpracy z firmą informatyczną dostarczającą kompleksowych rozwiązań sprzętowych oraz oprogramowania.

Poziomy model a2M do identyfikacji	do potwierdzenia – poziom adaptacji inne poziomy?		Procesy adaptacji?
Warstwa modelu a2M do identyfikacji	do potwierdzenia – warstwa funkcjonalności inne warstwy?		Procesy adaptacji?
Początek realizacji:	Kwiecień 2011	wykonano	
Termin zakończenia:	Wrzesień 2011	wykonano	

Źródło: opracowanie własne

Opis przebiegu eksperymentu

Do realizacji projektu przystąpił zespół złożony z przedstawicieli organizacji naukowo-badawczej, przedstawicieli firm kooperujących w ramach planowanego projektu oraz przedstawiciele firmy zainteresowanej dostarczeniem odpowiedniej infrastruktury potrzebnej dla planowanego data center. Przed jego realizacją dokonano ponownej identyfikacji wszystkich elementów wchodzących w skład struktury projektowej. W projekcie tym oprócz zespołu złożonego z osób zarządzających, uwzględniono również szereg podmiotów zainteresowanych wynikami projektu (interesariusze), którzy zostali zdefiniowani w projekcie jako agenty. Takie podejście wynikało z uwarunkowań prawnych, w ramach których należało stworzyć konsorcjum spełniające wymogi Programu Operacyjnego Innowacyjna Gospodarka. Takie uwarunkowania spowodowały także konieczność stworzenia złożonej struktury projektowej w skład, której wchodziła organizacja koordynująca działania projektowe (Zakład Zarządzania Technologiami Informatycznymi) oraz 9 podmiotów podległych. Okazało się że do stworzenia struktury niezbędne okazało się podejście agentowe. Przypisanie cech i zachowań poszczególnych uczestników projektu było więc pierwszym krokiem realizowanym przez koordynatora projektu.

W drugim kroku zdefiniowano zapotrzebowanie na wiedzę ekspercką, zwłaszcza w zakresie prawa. Postanowiono również (w związku z wysoką entropią projektu) wyłonić kolejnego agenta zadaniowego, którego zadaniem było gromadzenie stosownej dokumentacji. Zaadaptowano w ten sposób dobrą praktykę typową dla metod ciężkich, gdzie powierza się odpowiedzialność za przygotowywanie i gromadzenie dokumentacji członkowi zespołu projektowego. Było to konieczne z uwagi na sytuację, w której dla wysokiego poziomu dojrzałości zespołu projektowego i wysokiego poziomu entropii uniemożliwiono zespołowi pracę ze względu na problemy prawne projektu. Dopiero włączenie agenta eksperta (reprezentant kancelarii adwokackiej) pozwoliło na zmniejszenie entropii projektu.

Kolejnym działaniem (zgodnym z modelem szczegółowym **a2M**) było wyodrębnienie agenta koordynującego zarówno projekt, jak też pracę pozostałych agentów. Dla przyjętej struktury projektowej (zgodnej z modelem szczegółowym **a2M**) niezbędne okazało się powołanie agenta koordynatora. Agentem koordynatorem była osoba z dużym doświadczeniem projektowym. Zadaniem agenta koordynatora było wybranie odpowiednich dobrych praktyk zarządzania projektem w zależności od zmiennych decyzyjnych (dojrzałość zespołu oraz entropii projektu). Dobierano je z wykorzystaniem algorytmów dopasowania zawartych w funkcji `A_MATCH`. W projekcie uczestniczył autor niniejszej rozprawy jako obserwator, którego zadaniem było śledzenie realizacji prac przez wszystkie agenty powołane w omawianym projekcie.

Omawiany eksperyment stanowił podsumowanie wszystkich dotychczasowych badań (1-R do 7-R) i eksperymentów (1-P do 2-P) związanych z weryfikacją modelu szczegółowego **a2M**. W procesach weryfikacji na wstępie wyodrębniono wszystkie agenty niezbędne do realizacji projektu. W następnym kroku ustalono relacje między agentami oraz zestawy cech i zachowań dla wszystkich

agentów. Dokonano również pomiarów entropii oraz dojrzałości zespołu. Zespół złożony z osób bardzo dobrze znających specyfikę projektów informatycznych był zespołem o wysokiej dojrzałości. W takim przypadku entropia projektu była mała. Dojrzałość klienta (pełna dokumentacja wymagań) była duża, ale bez większego wpływu na przebieg planowania realizacji projektu.

Tak przeprowadzone procesy weryfikacji (na poziomie definiowalnym i zarządzanym modelu) pozwoliły na realizację działań związanych z adaptacją poszczególnych dobrych praktyk do projektu. Zdecydowano się na dobre praktyki związane z podejściami lekkimi, jak np. realizację w krótkich odstępach czasu (sprinty metody SCRUM). Problem, który zaobserwowano w trakcie procesów weryfikacji dotyczył braku kontroli realizacji zadań przez agenta koordynatora. Był to efekt problemów komunikacji z agentami zadaniowymi. Ostatecznie jednak wszystkie zadania zaplanowane w ramach przygotowania studium wykonalności projektu dotyczącego budowy centrum przetwarzania danych zrealizowano w terminie oraz przy założonym budżecie.

Wnioski

Przeprowadzony eksperyment podobnie jak eksperyment 1-P pozwolił na ocenę możliwości wsparcia zespołów projektowych modelem szczegółowym **a2M**. W porównaniu z poprzednim eksperymentem pozwolił na weryfikację warstwy funkcjonalności. Wnioski z badania 2-P przedstawiono w tabeli 6.5

Tabela 6.5 Wnioski z badania 1-P

		Kod eksperymentu: 2-P	
Realizacja celu badania	Cel badania osiągnięty		
Zidentyfikowane poziomy modelu szczegółowego a2M	Definiowania zarządzania adaptacji	Istnieją procesy adaptacji struktur agentowych	
Zidentyfikowane warstwy modelu szczegółowego a2M	wiedzy przetwarzania funkcjonalności	Istnieją procesy adaptacji struktur agentowych	

Źródło: opracowanie własne

Przeprowadzony eksperyment miał na celu weryfikację kompletnego modelu szczegółowego **a2M** w środowisku rzeczywistego projektu. Otrzymane wyniki potwierdziły, że zastosowanie podejścia agentowego daje duże korzyści z punktu widzenia realizacji zadań. Uczestnicy wiedzą jakie zadania leżą w ich kompetencjach oraz które komunikaty (zlecenia) są kluczowe dla realizacji określonych zadań. Przeprowadzone procesy kontroli wykazały, że zadania były realizowane terminowo. Przeprowadzony eksperyment pokazał również, że niezbędne dla przebiegu projektu informatycznego jest adaptowanie elementów struktury projektu.

Procesy adaptacji powinny dotyczyć:

- Po pierwsze - adaptacji poszczególnych agentów w ramach zespołu. Na podstawie analizy dojrzałości zespołu projektowego kierownik podejmuje decyzję dotyczącą włączenia agenta do danego zespołu.
- Po drugie - adaptacja dotyczy także dobrych praktyk, które są niezbędne do poprawnej i terminowej realizacji prac.
- Po trzecie - procesy adaptacji dotyczą także ekspertów. Tworząc strukturę projektową oraz łącząc poszczególne agenty relacjami, należy także uwzględniać obecność eksperta.

Zastosowanie modelu szczegółowego **a2M** w eksperymencie 2-P porządkuje przepływy informacji oraz wspomaga zarządzanie relacjami tworzonymi pomiędzy uczestnikami projektu.

6.1.4 Eksperyment 3-P. Implementacja poziomu definiowania modelu szczegółowego

Eksperyment 3-P dotyczył możliwości implementacji poziomu definiowania modelu szczegółowego **a2M**. Uznano za konieczne sprawdzenie w jaki sposób można dokonać transformacji elementów struktury projektowej do postaci agentów opisywanych zestawem cech i zachowań. Eksperyment przeprowadzono w ramach funkcjonowania Uniwersyteckiego Centrum Kompetencyjnego Technologii Rational powołanego przy współpracy Zakładu Zarządzania Technologiami Informatycznymi Politechniki Gdańskiej z firmą IBM. Do implementacji przystąpiono na przełomie stycznia i lutego 2012 roku. W tabeli 6.6 przedstawiono założenia do przeprowadzania badania.

Tabela 6.6 Założenia do badania 3-P

		Kod eksperymentu	3-P
Cel	Celem eksperymentu jest sprawdzenie możliwości implementacji poziomu definiowania modelu szczegółowego a2M .		
Środowisko badania	IBM Rational Team Concert, narzędzie do kompleksowej realizacji projektów informatycznych, wspomagające procesy zarządcze i wytwórcze. Implementacja: mgr inż. Paweł Madej, specjalista RTC w UCK Gdańsk		
Poziomy modelu a2M do identyfikacji	do potwierdzenia – poziom definiowania inne poziomy?	Procesy adaptacji?	
Warstwa modelu a2M do identyfikacji	do potwierdzenia – warstwa funkcjonalności inne warstwy?	Procesy adaptacji?	
Początek realizacji:	Styczeń 2012	wykonano	
Termin zakończenia:	Luty 2012	wykonano	

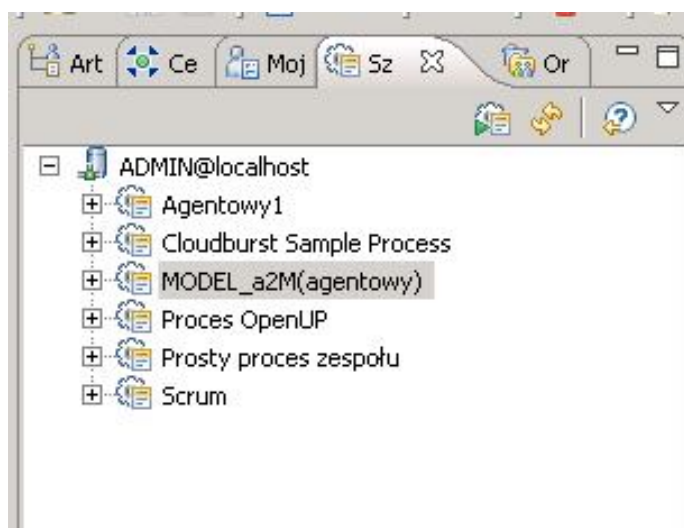
Opis eksperymentu

Przygotowaną specyfikację modelu szczegółowego **a2M** przekazano osobie odpowiedzialnej za implementację. W przekazanej dokumentacji zawarto opis poziomu definiowania modelu. Oczekiwano implementacji w narzędziu RTC poziomu definiowania modelu szczegółowego **a2M** (definiowania agentów poprzez nadanie im nazw oraz przypisanie cech i zachowań). Oczekiwano również zbudowania (w oparciu o narzędzie) interfejsu pozwalającego kierownikowi projektu dowolny dobór agentów (elementy struktury projektowej) oraz modyfikowanie ich cech i zachowań.

Ograniczeniem implementacyjnym okazało się środowisko RTC z uwagi na własny interfejs graficzny w ramach, którego można rozbudowywać tylko podstawowe jego funkcjonalności. Efekty implementacji przedstawiono na rysunkach (6.1-6.5). Każdy z etapów implementacji został szczegółowo opisany.

Należy jednocześnie nadmienić, że realizacja tego eksperymentu odbywała się z zastosowaniem podejścia agentowego (praca w rozproszonym środowisku, reagowanie na określone komunikaty osoby zlecającej wykonanie implementacji). Przebieg eksperymentu miał także charakter iteracyjny. W ramach iteracji implementowano kolejne warstwy modelu szczegółowego **a2M**.

Podstawowym procesem, który należało wykonać podczas implementacji modelu szczegółowego **a2M** w narzędziu RTC to przygotowanie obszaru roboczego, w którym możliwe jest zaimplementowanie modelu. Konieczne było przygotowanie dedykowanego szablonu procesu. Szablony procesu w RTC stanowią obszary robocze, w których kierownik projektu może definiować elementy struktury projektu. Takie podejście wydaje się jak najbardziej zasadne dla modelu szczegółowego **a2M**. Dzięki temu kierownik korzystający z narzędzia otrzymuje możliwość wyboru modelu szczegółowego **a2M** jako alternatywy względem wbudowanych w RTC szablonów procesów zgodnych z metodami zarządzania projektami (np. SCRUM). Na tej podstawie przygotowano szablon dla modelu szczegółowego **a2M** stanowiący środowisko do późniejszego definiowania agentów. Wynik implementacji szablonu modelu szczegółowego **a2M** przedstawiono na rys. 6.1



Rys. 6.1 Zdefiniowanie obszaru roboczego dla modelu szczegółowego **a2M**

Źródło: opracowanie własne

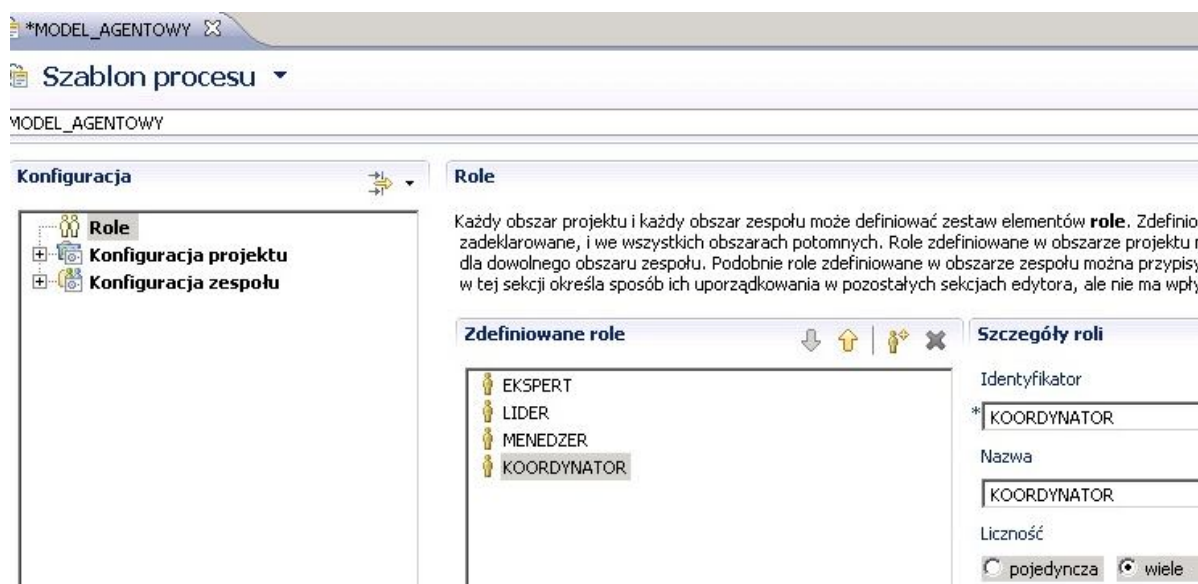
Po implementacji szablonu uzyskano środowisko do definiowania wszystkich pozostałych elementów struktury projektowej. Postanowiono dodatkowo sprawdzić możliwość implementacji zarówno członków zespołu jak również zasobów wiedzy (modele ocenowe albo dobre praktyki).

Możliwość definiowania struktury projektowej

Po zdefiniowaniu obszaru roboczego przystąpiono do pierwszego kroku związanego z badaniem zasadności predykcyjnej modelu. Postanowiono sprawdzić możliwość implementowania struktury projektowej z wykorzystaniem narzędzia RTC. Uznano za konieczne sprawdzenie możliwości definiowania takich elementów struktury jak: członkowie zespołu, eksperci, klient, dobre praktyki zarządzania projektami informatycznymi.

Okazało się, że środowisko implementacji modelu szczegółowego **a2M** jakim jest RTC posiada wbudowane funkcjonalności umożliwiające definiowanie ról projektowych. Uznano zatem, że zgodnie z założeniami dla najniższej warstwy modelu szczegółowego **a2M** konieczne jest umożliwienie kierownikowi projektu definiowanie poszczególnych elementów struktury projektowej w taki sposób, w jaki definiuje się role w metodach zarządzania projektami informatycznymi.

Implementacja tej funkcjonalności wykazała, że zdefiniowanie składników (elementów) struktury projektowej dokonuje się w RTC na zasadzie wyboru ról projektowych. Kierownik projektu, który dokonuje wyboru ról może nadawać nazwy zgodne ze swoimi założeniami oraz może określać liczebność poszczególnych elementów struktury projektowej. Na rysunku 6.2 przedstawiono możliwości definiowania elementów struktury projektu wg założeń modelu szczegółowego **a2M**.



Rys. 6.2 Definiowanie elementów struktury projektu wg założeń modelu szczegółowego a2M

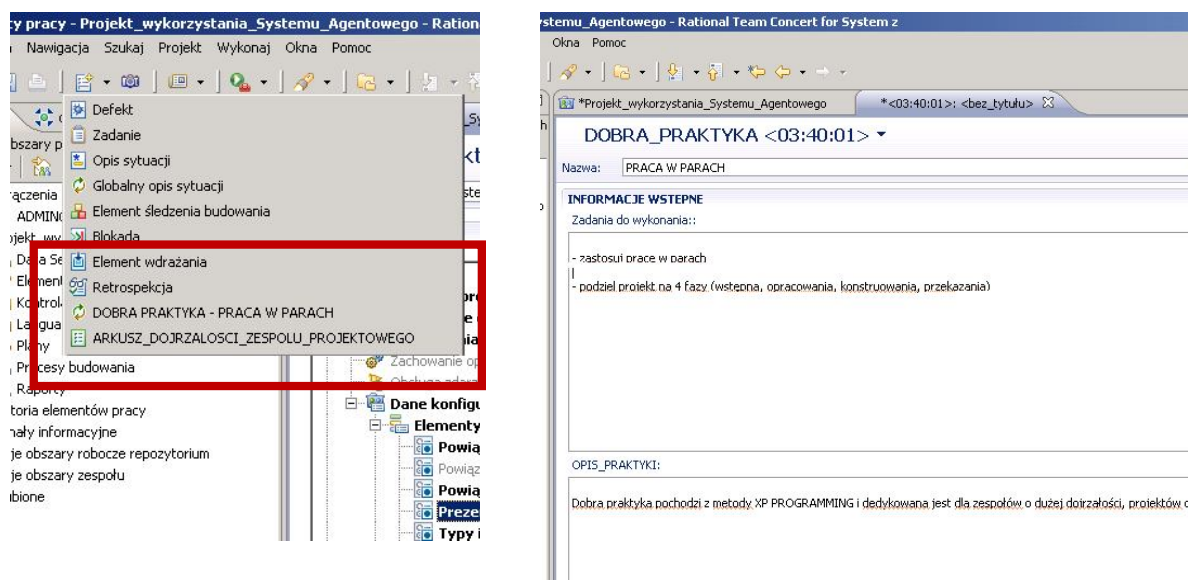
Źródło: opracowanie własne

Poddano także analizie możliwości implementacji modelu szczegółowego **a2M** poza strukturą szablonów. Efekt takiej implementacji postanowiono zaprezentować w kolejnym etapie eksperymentu, gdy sprawdzona zostanie możliwość stosowania podejścia agentowego podczas implementacji modelu w RTC. Jednocześnie podczas implementacji struktury projektu zauważono problem z możliwościami definiowania dobrych praktyk. Definiowanie dobrych praktyk nie może być realizowane w RTC na tej samej zasadzie w jaki definiuje się role projektowe.

Możliwość implementacji dobrych praktyk

Okazało się, że tworząc własny szablon do zarządzania projektami informatycznymi, narzędzie RTC nie posiada predefiniowanych dobrych praktyk zarządzania projektami. Istnieją co prawda szablony procesów dla projektów realizowanych według konkretnej metody zarządzania projektami (np. szablon do SCRUM albo RUP). Jednak w tych szablonych dobre praktyki dostępne są pośrednio, tzn. wynikają z działań prowadzonych w ramach wybranego szablonu. Np. wybierając szablon SCRUM kierownik projektu może planować sprint, może definiować rejestr produktu lub rejestr sprintu (nazwane w RTC dziennikami). Uznano jednak, że skoro dobre praktyki implementuje się na końcu (wprowadzane na poziomie adaptacji) kierownicy projektów powinni jedynie mieć świadomość konieczności stosowania dobrych praktyk i ich znaczenia dla powodzenia projektu.

Przyjęto definiowanie dobrych praktyk jako określonych jednostek pracy (ang. *work items*) traktując te dobre praktyki jako źródło wiedzy dla kierownika projektu. Takie podejście wynikało z konieczności zapewnienia przejrzystości interfejsu w szablonie modelu szczegółowego **a2M**. Zaimplementowanie dobrych praktyk w obszarze ról, powodowałoby ich nieczytelność z punktu widzenia użytkownika. Zaimplementowanie ich w narzędziu jako jednostki pracy nie zmieniło agentowego charakteru dobrych praktyk. Przypisanie ich do konkretnego agenta w narzędziu RTC pozwala danemu agentowi zapoznać się z nimi w celu poszerzenia wiedzy. Na rysunku 6.3 przedstawiono sposób implementacji dobrych praktyk w modelu szczegółowego **a2M**.



Rys. 6.3 Implementacja dobrych praktyk w modelu szczegółowego a2M

Źródło: opracowanie własne

Jak wynika z rysunku 6.3, w szablonie procesu dedykowanym dla modelu szczegółowego **a2M**, przyporządkowano kierownikowi projektu przykładowe dobre praktyki. Kierownik projektu powinien zapoznać się z nimi a następnie podejmować decyzje odnośnie wybrania tylko tych, które są niezbędne dla realizacji danego projektu.

Możliwość uwzględnienia eksperta (sposób komunikacji z ekspertem)

Realizując niniejszy eksperyment brano pod uwagę również możliwości uwzględniania ekspertów. Niewątpliwie ekspert jest jednym z podstawowych elementów struktury projektowej, ale na podstawie eksperymentów przeprowadzonych podczas badania zasadności replikatywnej, zaobserwowano duże zróżnicowanie dziedzin w ramach, których kierownicy poszukują ekspertów. Wczesniejsze analizy wyraźnie pokazały, że zapotrzebowanie na eksperta może dotyczyć obszarów niekoniecznie związanych z zarządzaniem projektami informatycznymi. Może istnieć zapotrzebowanie na ekspertów z innych dziedzin jak prawo, administracja, albo zupełnie odległej dziedziny (np. medycyna, gdy projekt realizowany jest dla branży medycznej). Pojawiło się więc pytanie w jaki sposób kierownik może określić tematy, w których ekspert może wspomagać jego projekt.

Drugim pytaniem, które pojawiło się podczas eksperymentu 3-P było to jak włączać eksperta do projektu, skoro zapotrzebowanie na eksperta może wystąpić w dowolnym momencie trwania projektu. Kierownik projektu nie może bowiem przewidzieć zapotrzebowania na wszystkich ekspertów na początku trwania projektu. Pierwszy z problemów związanych z definiowaniem eksperta w projekcie informatycznym udało się rozwiązać w narzędziu RTC korzystając z dodatkowych pól uszczegółwiających każdy z definiowanych elementów struktury projektowej. Na rysunku 6.4 przedstawiono sposób definiowania atrybutów elementu struktury projektowej (eksperta).

Rys. 6.4 Definiowanie atrybutów elementu struktury projektowej (eksperta)

Źródło: opracowanie własne

Następnie postanowiono rozwiązać kolejny problem, związany z korzystaniem z narzędzia RTC jako środowiska wspierającego zarządzanie projektem wg modelu szczegółowego **a2M**. Należało bowiem wykluczyć konieczność instalacji środowiska RTC u eksperta (który może przebywać w dowolnym miejscu na świecie) oraz wykluczyć konieczność szkolenia eksperta z obsługi RTC. Uznano więc, że uczestnictwo eksperta w projekcie informatycznym zarządzanym wg modelu szczegółowego **a2M** a wspieranym przez środowisko RTC należy wspomagać z poziomu cienkiego klienta (opcji dostępnej w RTC za pośrednictwem serwera JTS- Jazz Team Server). Rozwiązanie takie umożliwia korzystanie z określonych funkcjonalności RTC z poziomu przeglądarki internetowej.

Możliwość definiowania agentów (zastosowania podejścia agentowego)

Kolejnym etapem eksperymentu było badanie możliwości implementacji elementów struktury projektu w oparciu o agenty (podejście agentowe) w RTC. Implementacja podejścia agentowego jest kluczowa z punktu widzenia dalszych działań. Zgodnie bowiem z przyjętym podejściem agentowym, każdy uczestnik projektu oraz każda dobra praktyka powinna być definiowana jako niezależny agent, który w efekcie działań i decyzji kierownika projektu wkomponowany jest w strukturę projektu. Dlatego też na początku sprawdzono na ile definiowanie poszczególnych elementów struktury projektowej pozwala na zachowanie odrębności, autonomii i niezależności działania poszczególnych agentów. Obecnie przystąpiono do sprawdzenia czy definiując poszczególne elementy struktury projektowej (jak członkowie zespołu, ekspert czy też dobre praktyki) możliwe jest ich definiowanie z punktu widzenia typowych atrybutów agenta przyjętych w modelu szczegółowym **a2M**. Atrybuty agenta określane w modelu szczegółowego **a2M** dotyczą dwóch aspektów – cech oraz zachowań. Na rysunku 6.5 przedstawiono metodę definiowania agentów w RTC (cechy i zachowania).

Rys. 6.5 Definiowanie agentów w RTC (cechy i zachowania)

Źródło: opracowanie własne

Każdemu agentowi można przypisać takie cechy, które określają jego kompetencje oraz parametry komunikacyjne (adres e-mail, telefon). Podczas implementacji natrafiono jednak na problemy ze zdefiniowaniem zachowań. Podczas korzystania z RTC stwierdzono, że zachowań nie można zdefiniować w tym samym widoku, w którym definiuje się cechy. Przyjęto więc, że zachowania agenta (stanowiące zgodnie z definicją agentów, stany jakie może przyjmować agent w określonych momentach projektu informatycznego) należy definiować w formie możliwych do wykonania jednostek pracy. Takie podejście wydaje się zgodne z założeniami podejścia agentowego. Po pierwsze pozwala przypisać niezależne zachowania każdemu agentowi, po drugie pozwala parametryzować te zachowania (kiedy dany agent będzie uruchomiony). Dodatkowo narzędzie RTC pozwala przypisać określonym jednostkom pracy, a tym samym zachowaniom agentów, określony status (np. w trakcie, zakończono) co uznano za dodatkową możliwość implementacji.

Wnioski

Przeprowadzony eksperyment pozwolił na ocenę możliwości implementacji poziomów i warstw modelu szczegółowego **a2M**. Wnioski z badania 3-P przedstawiono w tabeli 6.7

Tabela 6.7 Wnioski z badania 3-P

		Kod eksperymentu: 3-P	
Realizacja celu badania	Cel badania osiągnięty		
Zidentyfikowane poziomy modelu szczegółowego a2M	Definiowania	Istnieją procesy adaptacji struktur agentowych	
Zidentyfikowane warstwy modelu szczegółowego a2M	wiedzy przetwarzania funkcjonalności	Istnieją procesy adaptacji struktur agentowych	

Źródło: opracowanie własne

Przeprowadzony eksperyment pozwolił na wykazanie na ile opracowana koncepcja modelu szczegółowego **a2M** daje się zastosować w określonym środowisku do zarządzania projektami informatycznymi. Podczas eksperymentu wykazano, że korzystając z tego narzędzia, możliwe jest zaimplementowanie zarówno podstawowych elementów struktury projektowej jak klient czy zespół. Wykazano również, że możliwe jest przekształcenie podstawowych elementów do postaci agentów, zgodnie z obecnym w modelu podejściem agentowym. Zauważono ponadto, że możliwe jest implementowanie podstawowych parametrów (atrybutów) poszczególnych agentów, czyli ich cech i zachowań.

Dodatkowym spostrzeżeniem było to, że podczas definiowania zachowań zaobserwowano również możliwość określania statusu zachowań poszczególnych agentów (w trakcie realizacji, zakończono itp.), co miało być jednym z elementów kolejnego eksperymentu, gdzie planowano zbadanie możliwości kontrolowania przepływu zleceń między agentami. Ponadto przeprowadzony eksperyment pozwolił na stwierdzenie, że specyfikacja modelu szczegółowego **a2M** opracowana w niniejszej rozprawie jest wystarczająco czytelna do tego, aby zaimplementować model w przykładowym narzędziu wspomagającym zarządzanie projektami informatycznymi.

6.1.5 Eksperyment 4-P. Implementacja relacji pomiędzy agentami oraz zarządzania przepływem komunikatów modelu szczegółowego

Kolejny eksperyment przeprowadzony w celu weryfikacji predykcyjnej modelu szczegółowego **a2M** dotyczył sprawdzenia możliwości tworzenia relacji pomiędzy agentami oraz zarządzania przepływem komunikatów między agentami. Wszystkie warunki środowiska przeprowadzenia eksperymentu pozostały bez zmian (ta sama osoba implementująca, ten sam sposób komunikacji, przesłanie specyfikacji przed implementacją. Na rysunku 6.8 przedstawiono założenia dla eksperymentu 4-P.

Tabela 6.8 Założenia do badania 4-P

		Kod eksperymentu 4-P	
Cel	Celem eksperymentu jest implementacja przepływów informacji (zleceń) pomiędzy agentami. Podczas eksperymentu należy wykazać możliwości implementacji podstawowych relacji hierarchicznych pomiędzy agentami oraz możliwości definiowania komunikatów dla poszczególnych agentów. Poddano także ocenie możliwość odwzorowania w wybranym narzędziu elementów rozproszonego środowiska (zgodnie z podejściem agentowym).		
Środowisko badania	IBM Rational Team Concert, narzędzie do kompleksowej realizacji projektów informatycznych, wspomagające procesy zarządcze i wytwórcze. Implementacja: mgr inż. Paweł Madej, specjalista RTC w UCK Gdańsk		
Poziomy model a2M do identyfikacji	do potwierdzenia – poziom zarządzania	inne poziomy?	Procesy adaptacji?
Warstwa modelu a2M do identyfikacji	do potwierdzenia – warstwa funkcjonalności	inne warstwy?	Procesy adaptacji?
Początek realizacji:	Styczeń 2012	wykonano	
Termin zakończenia:	Luty 2012	wykonano	

Źródło: opracowanie własne

Opis eksperymentu

Eksperyment 4-P dotyczył badania możliwości implementacji modelu szczegółowego **a2M** w narzędziu wspomagającym zarządzanie projektami informatycznymi (definiowania relacji pomiędzy agentami). Celem eksperymentu było sprawdzenie, czy w RTC możliwe jest określenie hierarchicznych zależności pomiędzy agentami. Kolejnym etapem eksperymentu było określenie możliwości przesyłania zleceń pomiędzy agentami (wysyłania komunikatów, w zależności od których agent podejmuje określoną akcję).

Podobnie jak w przypadku pierwszego eksperymentu, przygotowano specyfikację dotyczącą poziomu zarządzania, którą przekazano specjalście do narzędzia RTC w celu określenia możliwości implementacji. Obejmowała ona: strukturę projektu, wysyłanie komunikatów przez agenty, badanie niezależności działania agentów, dobieranie i uruchamianie agentów w zależności od potrzeb oraz możliwość weryfikowania poziomu dojrzałości (implementacja proponowanego w pracy instrumentu pomiarowego do pomiaru dojrzałości).

Definiowanie struktury (hierarchii)

Podczas implementacji hierarchii pomiędzy agentami zauważono pewne problemy natury technicznej RTC. Narzędzie to nie umożliwia bowiem graficznego modelowania zależności pomiędzy obiektami (tym samym agentami). Umożliwia jednak tworzenie kategorii dla poszczególnych obiektów (w tym wypadku agentów). Uznano zatem, że rozwiązaniem problemu dotyczącego definiowania relacji hierarchicznych między agentami, jest ujęcie ich w określonej kategorii. Oznacza to konieczność definiowania kategorii dla jednego agenta (np. AGENT MENERDZER). Następnie definiuje się kolejną kategorię dla agenta niższego szczebla (np. AGENT LIDER ZESPOŁU).

Takie zastosowanie narzędzia RTC ma jednak swoje minusy, wymaga bowiem nałożenia na wszystkie istniejące elementy dodatkowej kategorii, co niekoniecznie może być zrozumiałe dla przykładowego kierownika projektu, który będzie wybierał szablon modelu szczegółowego a2M do zarządzania swoim projektem. Dlatego też dla prezentacji zależności między agentami zaproponowano dwa warianty.

- Wariant pierwszy oznaczał poszukiwanie takich funkcjonalności w samym narzędziu RTC, które pozwolą na tworzenie zależności hierarchicznych między poszczególnymi agentami.
- Wariant drugi oznaczał konieczność rozbudowy narzędzia RTC o odpowiednią wstyczkę (ang. *plug-in*).

Realizując wariant pierwszy zdecydowano się uwzględnić hierarchie agentów poprzez definiowanie niestandardowych atrybutów dla poszczególnych zadań wykonywanych przez agenty oraz definiując ich dodatkowe atrybuty w ramach uprzednio zdefiniowanych cech. Na rysunku 6.6 zaprezentowano odzwierciedlenie relacji pomiędzy przykładowymi agentami projektu.

Działania	każdy (dom...	AGENT_EKS...	AGENT_LIDER	AGENT_MEN...
Budowanie	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Elementy pracy	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Usunięty zapytanie (serwer)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Zapisz element pracy (serwer)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Zapisz zapytanie (serwer)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Konektory elementów	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Synchronizuj z obiektami zewnętrznymi (serwer)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Kontrola kodu źródłowego	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Dostarcz (serwer)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Zapisz komponent (serwer)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Zapisz linię bazową (serwer)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Zapisz obraz stanu (serwer)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Zapisz strumień (serwer)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Panele kontrolne	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Zapisz panel kontroli zespołu (serwer)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Planowanie	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Usunięty plan (serwer)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Zapisz plan (serwer)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Proces	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Generuj zaproszenie do zespołu (serwer)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Zapisz obszar zespołu (serwer)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Raporty	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Wdróż raport (serwer)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Wdróż szablon raportu (serwer)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Wyświetl raport (serwer)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Zarządzaj folderem raportów (serwer)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Rys. 6.6 Odzwierciedlenie relacji pomiędzy przykładowymi agentami

Źródło: opracowanie własne

Wariant drugi związany z budową plug-inu nie został ostatecznie wdrożony. Stworzono jednak koncepcję wykorzystania wtyczek z innego narzędzia z rodziny Rational. Okazało się bowiem, że w narzędziu Rational Method Composer - RMC (narzędzie do planowania projektów informatycznych) istnieje szereg komponentów pozwalających na tworzenie relacji hierarchicznych. Ponadto narzędzie RMC posiada wbudowany edytor graficzny w którym można tworzyć relacje pomiędzy obiektami na zasadzie przeciągnij i upuść (ang. *drag-and-drop*).

Wysyłanie komunikatów przez agenty

Kolejnym procesem weryfikacji modelu szczegółowego **a2M** (w ramach eksperymentu 4-P) było sprawdzenie na ile opracowany model po procesach implementacji umożliwia organizowanie przepływów informacji między agentami. Zgodnie z przyjętym podejściem agentowym, każdy agent reaguje na określony komunikat i wykonuje swoje zadanie (zgodnie ze zdefiniowanym dla niego zachowaniem). Należało więc sprawdzić czy przyjęty w modelu przepływ komunikatów jest możliwy do implementacji.

Pierwsza analiza wskazywała na konieczność definiowania parametrów komunikatów na poziomie serwera JTS (ang. *Jazz Team Server*), który zapewnia obsługę większości procesów realizowanych w narzędziu RTC. Na poziomie serwera można bowiem skonfigurować przepływy informacyjne pomiędzy poszczególnymi uczestnikami projektu. W odniesieniu do modelu szczegółowego **a2M**, można na poziomie serwera ustawić parametry przepływu informacji pomiędzy agentami. Każdy agent mający wykonać zadanie otrzyma wiadomość na adres e-mail. Można takie komunikaty przysyłać również za pośrednictwem RTC. Osoba pracująca jako np. AGENT KOORDYNATOR po zalogowaniu do RTC może sprawdzić jakie komunikaty otrzymała (np. wykonaj zadanie X).

Ostatnim badanym procesem było zarządzanie przepływem zleceń (czyli komunikatów). Zgodnie z przyjętą hierarchią agentów, zlecenia może dokonywać określony agent innym, określonym uprzednio agentom. Postanowiono zatem stosując się do tego założenia, poddać implementacji ograniczenia w przepływie komunikatów. Na rysunku 6.7 przedstawiono mechanizm przesyłania komunikatów między agentami za pośrednictwem RTC.

Twórca	Właściciel	Modyfikujący	Subskrybent	Opis zmiany
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Dodano nowy komentarz
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Zmieniono podsumowanie lub opis
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Zmieniono: Istotność, Priorytet lub Zaplanowane dla
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Zmieniono właściciela
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Dodano lub usunięto znaczniki
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Dodano lub usunięto subskrybenta
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Dodano lub usunięto połączenie
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Element pracy został utworzony lub otwarty ponownie
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Element pracy został zamknięty
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Właściciel rozpoczął/zakończył pracę z elementem pracy
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Inna zmiana stanu
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Inna zmiana elementu pracy niewymieniona powyżej

Rys. 6.7 Mechanizm przesyłania komunikatów między agentami za pośrednictwem RTC

Źródło: opracowanie własne

Na rys. 6.7 przedstawiono sposób, w jaki kierownik projektu korzystając z narzędzia RTC może określać sposób przepływu komunikatów poprzez wybranie kanału (poczta elektroniczna, komunikacja wewnątrz RTC). Może również parametryzować te komunikaty nadając im określoną rangę. Na poziomie serwera możliwe jest również określenie atrybutów komunikatów, dzięki czemu nie każdy agent może dokonać zlecenia. Można zatem zarządzać przepływem komunikatów zgodnie z założeniami przyjętymi w modelu szczegółowym **a2M**.

Niezależność działania agentów

Kolejnym procesem weryfikacji modelu szczegółowego **a2M** jest sprawdzenie działania agentów w środowisku rozproszonym. Większość prowadzonych badań (rozdział V) wykazało, że w przeważającej większości projektów, uczestnicy projektów pracowali w środowisku rozproszonym.

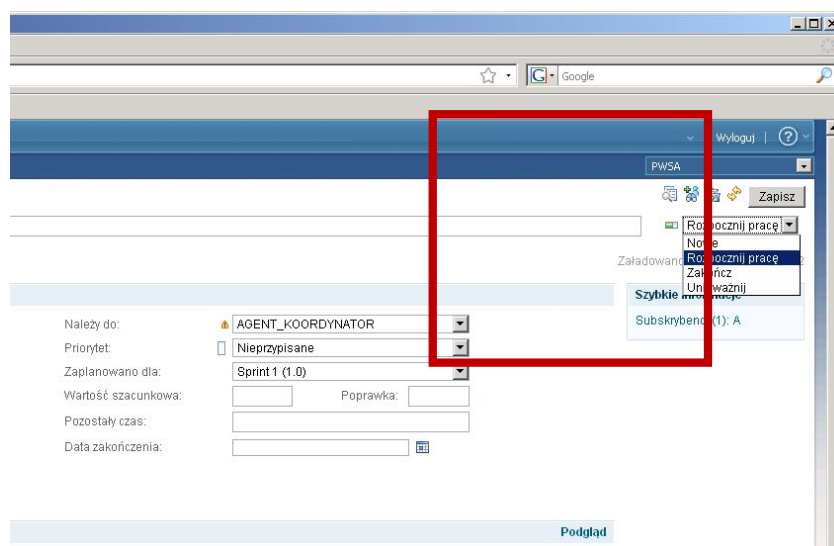
Jest to typowe zjawisko funkcjonowania zespołu dostawy. Postanowiono poddać weryfikacji, czy środowisko RTC stwarza warunki do implementacji środowiska rozproszonego.

Analiza modelu **a2M** i jego implementacji w RTC wykazała, że po zdefiniowaniu agentów i określeniu ich atrybutów, agenci mogą pracować w dowolnym miejscu pod warunkiem posiadania dostępu do Internetu. Za pomocą repozytorium projektowego wbudowanego w Jazz Team Serwer każdy agent może wykonywać swoje zadania zdalnie, a efekty pracy umieszczać w repozytorium. Agenty mogą również integrować/dzielić swoją pracę lub pracować sekwencyjnie, gdyż RTC zapewnia także kontrolę wersji poprzez moduł zarządzania zmianami i konfiguracją. Komunikacja pomiędzy agentami oraz wbudowane w RTC opcje kontroli postępów pozwalają monitorować AGENTOWI Menedżerowi działania wszystkich agentów bez konieczności spotykania się z nimi.

Dobieranie i uruchamianie agentów w zależności od potrzeb

W ramach eksperymentu 4-P model szczegółowy **a2M** poddany był ocenie możliwości implementacji zmiany struktury zespołu projektowego (dodawania i uruchamiania agentów). Uznano za konieczne sprawdzenie na ile narzędzie, w którym poddana jest implementacji ta funkcjonalność modelu wspomaga dobieranie i uruchamianie agentów w zależności od potrzeb.

Założono zgodnie z podejściem agendowym, że poziom zarządzania modelu szczegółowego **a2M** w warstwie funkcjonalności prezentuje zasady doboru i uruchomienia agentów w zależności od potrzeb. Obserwacje poczynione podczas weryfikacji predykcyjnej pozwalały zauważyć, że w projekcie informatycznym zapotrzebowanie na określone działania agentów nie zawsze mają charakter ciągły. Oznacza to zatem tyle, że w określonej sytuacji uruchamiany jest odpowiedni agent, który wykonuje swoje zadanie. Zgodnie więc z podanym założeniem sprawdzono jak można poddać implementacji to zdarzenie w RTC. Istnieje możliwość uruchamiania agentów poprzez zmianę statusu przydzielonego zadania. Na rysunku 6.8 przedstawiono ten sposób uaktywnienia agenta do wykonania określonego zadania



Rys. 6.8 Sposób uaktywnienia agenta do wykonania zadania

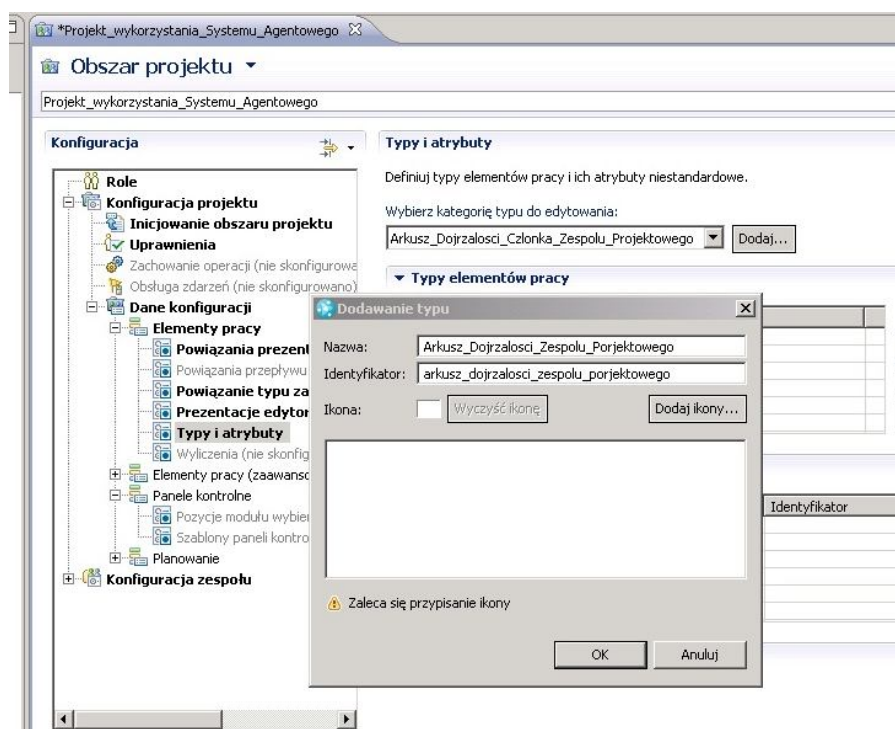
Źródło: opracowanie własne

Czy możliwe jest weryfikowanie dojrzałości? (implementacja instrumentu pomiarowego)

Ostatni z procesów eksperymentu 4-P dotyczył możliwości adaptacji poszczególnych elementów struktury projektowej w zależności od wartości zmiennych decyzyjnych (dojrzałość zespołu, dojrzałość klienta, entropia projektu). W tym celu postanowiono zbadać czy narzędzie RTC jest w stanie umożliwić dokonywanie pomiarów wartości tych zmiennych. Zgodnie z przyjętym

założeniem modelu szczegółowego **a2M**, kierownik projektu dobiera dobre praktyki zarządzania projektami w zależności od zmierzonych poziomów dojrzałości klienta i zespołu, oraz w zależności od zmierzonej entropii projektu.

Ponieważ takie podejście do pomiaru tych zmiennych nie jest typowym podejściem spotykanym np. w metodach zarządzania projektami informatycznymi, w RTC nie przewidziano odpowiednich funkcjonalności umożliwiających takie analizy. Uznano jednak, że pomiar ten jest na tyle ważny, że należy poddany implementacji w RTC model szczegółowy **a2M** i rozszerzyć o odpowiednie instrumenty pomiarowe. Na wstępie sprawdzono możliwości rozbudowywania RTC o odpowiednie kwestionariusze ankietowe. Na rysunku 6.8 przedstawiono metodę tworzenia formularzy w RTC



Rys. 6.9 Metoda tworzenia formularzy w RTC

Źródło: opracowanie własne

W ramach eksperymentu 4-P na podstawie wstępnej analizy stwierdzono możliwość implementacji instrumentów pomiarowych w modelu szczegółowym **a2M** w RTC. Narzędzie to umożliwia bowiem nadawanie elementom pracy parametrów (w tym pól do wpisywania informacji czy też pól wyboru). Uznano więc, że atrybuty elementu pracy będą wystarczające do zdefiniowania instrumentów pomiarowych zmiennych decyzyjnych jako dedykowane formularze.

Potraktowanie instrumentu pomiarowego, jako element pracy ma też uzasadnienie z punktu widzenia zarządzania projektem zgodnie z założeniami modelu szczegółowego **a2M**. Zakłada się bowiem, że każdy uczestnik projektu wypełnia odpowiedni kwestionariusz, który jest później analizowany przez kierownika projektu. Stąd też każdemu agentowi uczestniczącemu w projekcie można przydzielić zadanie związane z wypełnieniem takiego formularza. Dzięki takiemu podejściu możliwy będzie pomiar poziomów dojrzałości wartości poszczególnych zmiennych decyzyjnych.

Sprawdzenie możliwości rozbudowy RTC o instrumenty pomiaru zmiennych nowego trójkąta ograniczeń było niezbędne z punktu widzenia planowania kolejnego eksperymentu.

Wnioski

Przeprowadzony eksperyment pozwolił na ocenę możliwości implementacji poziomu zarządzania i warstwy przetwarzania oraz funkcjonalności modelu szczegółowego **a2M**. Wnioski z badania 3-P przedstawiono w tabeli 6.9

Tabela 6.9 Wnioski z badania 4-P

		Kod eksperymentu: 4-P	
Realizacja celu badania	Cel badania osiągnięty		
Zidentyfikowane poziomy modelu szczegółowego a2M	Zarządzania	Istnieją procesy adaptacji struktur agentowych	
Zidentyfikowane warstwy modelu szczegółowego a2M	wiedzy przetwarzania funkcjonalności	Istnieją procesy adaptacji struktur agentowych	

Źródło: opracowanie własne

Przeprowadzony eksperyment dotyczył implementacji poziomu zarządzania modelu szczegółowego **a2M**. Wykazał, że narzędzie wspomagające zarządzanie projektami informatycznymi umożliwia implementację modelu. Warto dodać, że sprawdzono nie tylko możliwość definiowania komunikatów między agentami, ale również możliwość dobierania i uruchamiania agentów do realizacji poszczególnych zadań.

Największym problemem implementacyjnym była wizualizacja zależności hierarchicznych. Prezentacja graficzna tych relacji w formie ogólnie dostępnej dla wszystkich uczestników (widocznej) dostarcza wielu problemów i wymaga dodawania do RTC dodatkowych elementów (wtyczek).

Ważnym osiągnięciem eksperymentu było sprawdzenie możliwości implementowania instrumentów do pomiaru zmiennych decyzyjnych wchodzących w skład nowego trójkąta ograniczeń. Umożliwienie budowania takich instrumentów pozwala w następnym eksperymencie przeprowadzenie pełnego eksperymentu, w którym umożliwiające zostanie przedłożenie uczestnikom odpowiednich formularzy badających poziom zmiennych decyzyjnych.

Zauważono również, że niezbędne jest przygotowanie procedur korzystania z modelu szczegółowego **a2M** poddanego implementacji w RTC dla kierowników, którzy nie znają założeń tego modelu.

6.1.6 Eksperyment 5-P. Implementacja dobrych praktyk modelu szczegółowego a2M

Eksperyment 5-P miał na celu implementację dobrych praktyk w zależności od wartości zmiennych decyzyjnych nowego trójkąta ograniczeń. Na rysunku 6.10 przedstawiono założenia dla eksperymentu 5-P.

Tabela 6.10 Założenia do badania 5-P

		Kod eksperymentu 5-P	
Cel	Celem niniejszego eksperymentu jest sprawdzenie, czy zdefiniowane uprzednio elementy struktury projektowej w postaci agentów (klient, członkowie zespołu, ekspert oraz dobre praktyki) ułożone w odpowiednich		

	relacjach na poziomie zarządzanym, mogą być adaptowane w zależności od nowego trójkąta ograniczeń. W eksperymencie należy wykazać możliwość pomiaru zmiennych decyzyjnych nowego trójkąta ograniczeń oraz możliwość zastosowania odpowiednich dobrych praktyk w zależności od wartości zmiennych decyzyjnych.	
Środowisko badania	IBM Rational Team Concert, narzędzie do kompleksowej realizacji projektów informatycznych, wspomagające procesy zarządcze i wytwórcze Implementacja: mgr inż. Paweł Madej, specjalista RTC w UCK Gdańsk	
Poziomy model a2M do identyfikacji	do potwierdzenia – poziom adaptacji inne poziomy?	Procesy adaptacji?
Warstwa modelu a2M do identyfikacji	do potwierdzenia – warstwa funkcjonalności inne warstwy?	Procesy adaptacji?
Początek realizacji:	Styczeń 2012	wykonano
Termin zakończenia:	Luty 2012	wykonano

Opis eksperymentu

W następstwie poprzedniego eksperymentu, uznano za konieczne sprawdzenie możliwości poddania implementacji instrumentów pomiarowych zmiennych decyzyjnych (dojrzałości zespołu, klienta oraz entropii projektu). Postanowiono również umożliwić za pomocą kwestionariusza w RTC analizę odpowiedzi udzielanych przez osoby wypełniające kwestionariusze. Należy przypomnieć, że formularze traktowane są w RTC jako zadania, które muszą wykonać osoby przystępujące do projektu. Ostatnim etapem eksperymentu jest przeprowadzenie przykładowej sesji z modelem szczegółowym **a2M** poddanym implementacji w RTC. Kierownik projektu korzystając z narzędzia RTC wybiera model szczegółowy **a2M** jako środowisko swojej pracy i po analizie dwu pierwszych poziomów modelu (definiowalnego i zarządzanego) na najwyższym poziomie dobiera dobre praktyki w zależności od nowego trójkąta ograniczeń. Celem tego eksperymentu jest zaprezentowanie sposobu doboru.

Procedura postępowania

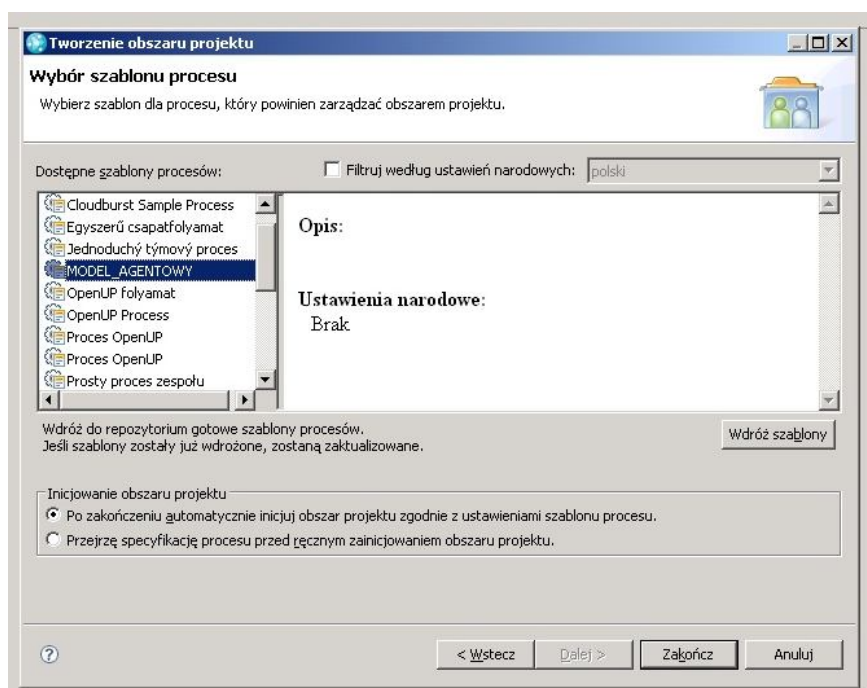
Celem eksperymentu 5-P jest wykazanie, że model szczegółowy **a2M** jest w pełni implementowany i stanowi dla kierownika projektu środowisko wpierające procesy zarządzania projektem. Aby korzystać z modelu **a2M** kierownik projektu powinien wstępnie zdefiniować elementy struktury projektowej jako odpowiednie agenty, następnie powiązać je relacjami a następnie uwzględniając poziomy dojrzałości zespołu i klienta oraz poziom entropii projektu, aby dobrać odpowiednie dobre praktyki zarządzania projektami.

Postępowanie zgodne z założeniami modelu ogólnego i szczegółowego **a2M** dotyczy realizacji określonych procesów przed przystąpieniem do realizacji projektu. Po wykazaniu możliwości implementacji podstawowych elementów struktury projektowej (eksperymenty 3-P, 4-P) uznano za celowe dostosowanie szablonu szczegółowego **a2M** do warunków pracy kierowników projektów informatycznych. Postanowiono przygotować procedurę postępowania złożoną z określonych kroków, które powinien zrealizować kierujący projektem. Kroki wymienione w procedurze odpowiadają działaniom podejmowanym zgodnie z założeniami modelu szczegółowego **a2M**. Zaprezentowano je poniżej:

1. Zdefiniuj elementy struktury projektowej, jako agenty projektu
2. Określ cechy i zachowania poszczególnych agentów
3. Określ przepływy komunikatów

4. Zapoznaj się z dobrymi praktykami zarządzania projektami
5. Określ skład zespołu/zespołów projektowych
6. Zbadaj poziomy dojrzałości

Tak określone kroki poddano implementacji, jako zadania dla kierownika projektu. Kierownik projektu informatycznego korzystający z narzędzia RTC w pierwszej kolejności decyduje się na wybór modelu szczegółowego **a2M** jako szablonu projektu, wg którego chce realizować swój projekt. Szablon projektu zgodny z założeniami modelu szczegółowego **a2M** jest dostępny z listy wszystkich możliwych szablonów dostępnych w RTC. Na rysunku 6.9 przedstawiono sposób wyboru szablonu modelu szczegółowego **a2M** przez kierownika projektu.



Rys. 6.10 Wybór szablonu modelu szczegółowego a2M przez kierownika projektu

Źródło: opracowanie własne

Po wybraniu szablonu przez kierownika projektu prezentowana jest lista zadań zgodnych z opisaną wcześniej procedurą. Dzięki takiemu podejściu kierownik projektu, który nie korzystał z szablonu modelu szczegółowego **a2M**, jest w stanie określić, które zadania powinien wykonać na tym etapie. Każde zadanie zawarte w procedurze może zostać uszczegółowione w załączonych do zadań opisach, dzięki czemu kierownicy uzyskują wskazówki (przewodniki) dotyczące ich realizacji.

Model szczegółowy a2M jako rozwiązanie alternatywne wobec klasycznych metod zarządzania projektami

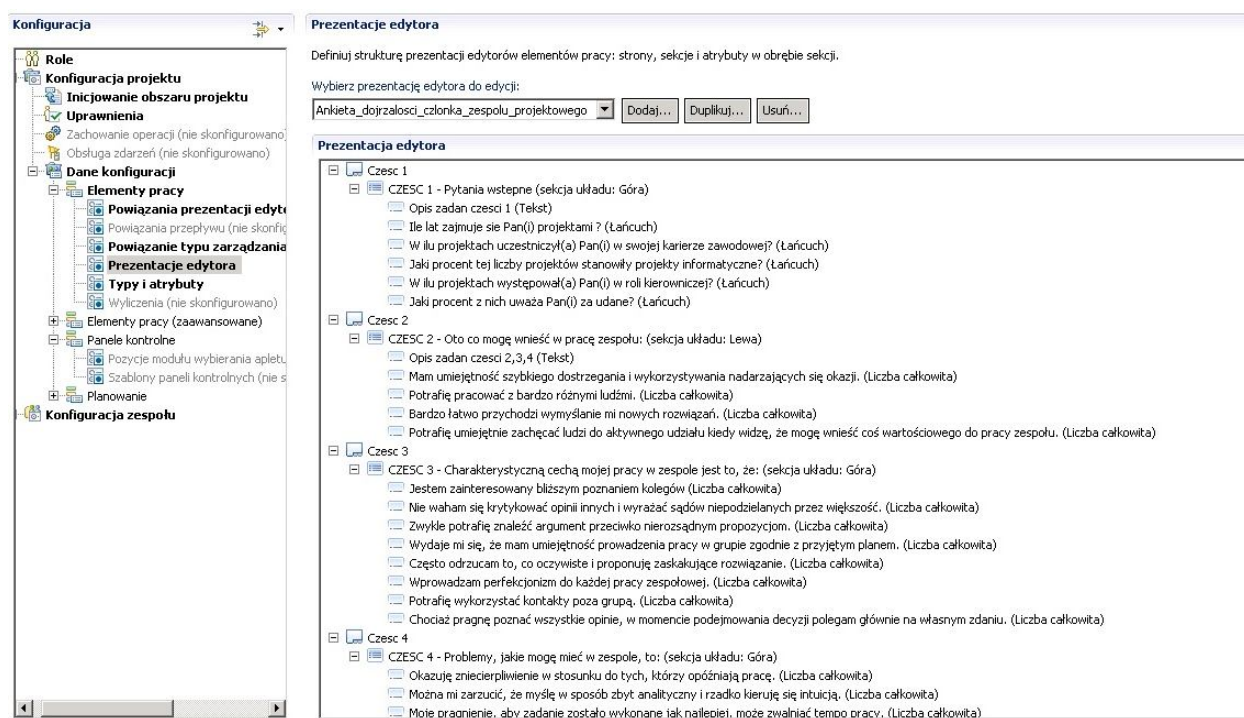
Omawiając możliwości implementacyjne opracowanego modelu szczegółowego **a2M** warto odnieść proponowane rozwiązanie do metod zarządzania projektami informatycznymi. Zgodnie z przyjętymi założeniami, zadaniem modelu jest usprawnienie zarządzania projektem informatycznymi poprzez zastosowanie podejścia agentowego oraz adaptację dobrych praktyk zarządzania projektami do nowego trójkąta ograniczeń. Analizując w ramach eksperymentu możliwości zastosowania modelu przez kierowników projektów korzystających z określonej technologii informatycznej postanowiono potraktować model jako rozwiązanie alternatywne

względem już istniejących metod zarządzania projektami. Umieszczenie modelu pośród szablonów metod zarządzania projektami jak RUP czy SCRUM związane jest z ograniczeniami narzędzia RTC, które wymaga od menedżera wyboru konkretnej metody zarządzania projektami lub tworzenia własnego procesu realizacji projektu. Należy jednak zaznaczyć, że model szczegółowy **a2M** nie jest typową metodą zarządzania projektami informatycznymi. Model szczegółowy **a2M** należy rozumieć szerzej ze względu na dualny charakter, usprawniający tworzenie struktur organizacyjnych (związanych z porządkowaniem przepływów informacji, dopasowywaniem dobrych praktyk itp.) oraz możliwości implementacji. Dzięki traktowaniu modelu jako adaptacyjne organizacyjno-implementationary rozwiązanie z jednej strony gromadzimy wiedzę o projekcie a drugiej wykorzystujemy ją do diagnozowania i prognozowania stanów projektów informatycznych. Stąd też niniejszy eksperyment bada przydatność tego modelu.

Umieszczenie więc modelu szczegółowego **a2M** pośród metod zarządzania projektami podyktowane było wyłącznie możliwościami narzędzia w którym implementowano model. Nie można modelu szczegółowego **a2M** traktować, jako dobrej praktyki także ze względu na fakt, że czerpie z wielu dobrych praktyk. Nie należy go także traktować jako metody zarządzania projektem.

Zdefiniowanie poziomów dojrzałości organizacji i klienta oraz entropii projektu

Kolejnym etapem eksperymentu jest sprawdzenie poziomów dojrzałości klienta, zespołu dostawcy oraz poziomu entropii. Aby to zrealizować konieczne jest poddanie procesowi implementacji instrumentów pomiarowych (ocena dojrzałości i entropii projektu). Wypełnione formularze (kwestionariusze ankietowe) mogą zostać złożone (zgrupowane) w repozytorium projektowym, które jest częścią RTC. Po ich zgromadzeniu agent menedżer może sprawdzić otrzymane wyniki, może do tego celu wykorzystać również innego agenta. Zgodnie z koncepcją proponowaną w modelu szczegółowym **a2M** zadanie to może zrealizować agent wnioskujący, którego celem może być wsparcie agenta menedżera m.in. w takich analizach. Na rysunku 6.10 przedstawiono przykład implementacji instrumentu pomiarowego do oceny dojrzałości zespołu dostawcy.



Rys. 6.11 Przykład implementacji instrumentu pomiarowego do oceny dojrzałości zespołu dostawcy

Źródło: opracowanie własne

Na rysunku 6.10 przedstawiono również arkusze kontrolne dla kierownika projektu, po wypełnieniu których będzie on w stanie ocenić wartości zmiennych decyzyjnych nowego trójkąta ograniczeń. W podobny sposób poddano implementacji pozostałe formularze stwarzające warunki do pomiaru dojrzałości klienta oraz entropii projektu.

Poddano również analizie możliwość automatycznej oceny wyników. Po wstępnym sprawdzeniu okazało się, że takie działanie jest możliwe poprzez rozbudowanie RTC o dodatkowy zewnętrzny moduł obliczeniowy. Oznacza to rozbudowanie funkcjonalności AGENTA WNIOSKUJĄCEGO o którym wspomniano wcześniej. Takie podejście pozwala na wprowadzenie do modelu nowego agenta – narzędzia informatycznego, które również jest adaptowane do potrzeb projektu i uruchomione w określonym czasie (narzędzie traktowane jako agent i wspomagające wnioskiowania – np. regułowa baza wiedzy).

Adaptacja dobrych praktyk

Po analizie możliwości oceny dojrzałości, kierownik projektu w ramach eksperymentu 5-P przyporządkowuje dobre praktyki zarządzania projektami do AGENTA MENERŻERA w formie elementów pracy. Przyjęto, że każda dobra praktyka jest opisana atrybutami (kontekstem jest stosowania) kierownik projektu wybiera te, które są niezbędne dla realizacji projektu. Jeżeli kierownik nie dobiera tych praktyk samodzielnie (aby nie popełnić błędów) zleca się wykonanie tych zadań agentowi który wykorzystuje funkcję A_MATCH (zdefiniowanej na potrzeby niniejszej rozprawy). Funkcja A_MATCH zawiera wskazówki dotyczące kontekstu (wartości zmiennych decyzyjnych), przy których należy stosować określone dobre praktyki. Funkcję A_MATCH zestawiono za pomocą arkusza kalkulacyjnego, który został podłączony jako zewnętrzne źródło wiedzy w RTC i potraktowany jako agent.

Sesja z modelem a2M poddanym implementacji w RTC

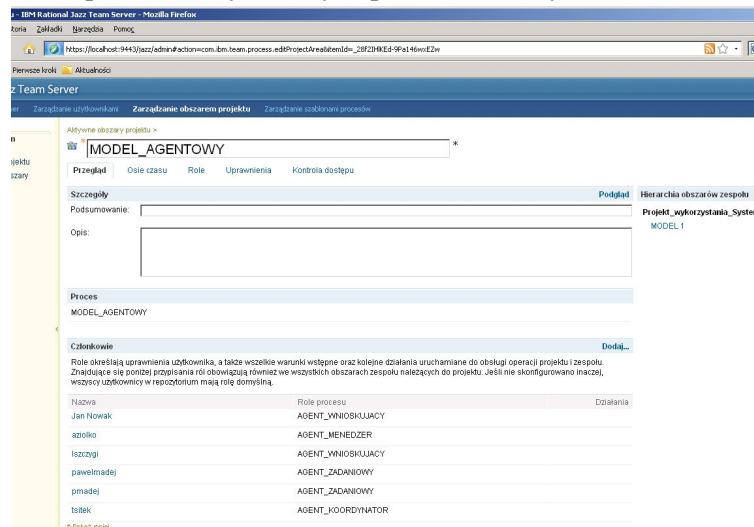
Po określeniu sposobu adaptacji dobrych praktyk zarządzania projektem informatycznym w zależności od zmiennych decyzyjnych nowego trójkąta ograniczeń (po zaimplementowaniu założeń dla najwyższego poziomu modelu szczegółowego **a2M**) w ramach eksperymentu 5-P postanowiono zorganizować przykładową sesję z modelem. Do sesji zaproszono osobę, która nie znała szczegółowo modelu **a2M**, ale znała zasady prowadzenia projektów informatycznych. Podczas sesji z modelem obserwowano reakcje kierownika projektu. Pytano również o opinie związane z użytkowaniem.

Po przeprowadzeniu sesji kierownik projektu (na potrzeby sesji nazwany testerem) podkreślił znaczenie dostępności procedury postępowania z modelem **a2M**. Definiowanie elementów struktury projektowej przebiegało bez większych problemów. Kilka problemów pojawiło się przy określeniu relacji pomiędzy agentami. Tester nie miał problemów z wykorzystaniem instrumentów pomiarowych do oceny dojrzałości, jak również formularzy końcowych dedykowanych kierownikowi. Kwestionariusze dojrzałości przekazano dwom przykładowym członkom zespołu. Tester po zastosowaniu modelu szczegółowego **a2M** stwierdził potrzebę automatycznego wnioskiowania dotyczącego prognoz poziomu dojrzałości klienta czy zespołu dostawcy.

W ostatnim kroku sesji z modelem **a2M** tester przystąpił do planowania projektu z zastosowaniem modelu szczegółowego **a2M**. Dobrał agenty do realizacji pierwszego etapu projektu. Zdecydował się na zastosowanie dobrej praktyki z metod lekkich planując pierwszy etap jako dwutygodniowy sprint a następnie wybrał parę agentów do realizacji pierwszego sprintu. Przydzielił zadania oraz uruchomił zadanie, przez co uruchomił komunikaty do agentów (członków zespołu), którzy mogli przystąpić do realizacji zadań. Po pilotażowej sesji z udziałem testera uznano implementację modelu szczegółowego **a2M** za zakończoną.

Możliwość realizacji prac i zarządzanie wszystkimi agentami (kontrolowanie, raportowanie)

W ramach eksperymentu 5-P stwierdzono, że oprócz komunikatów o zrealizowaniu zadań, AGENT Menedżer może śledzić postępy w realizacji prac oraz monitorować funkcjonowanie poszczególnych agentów za pomocą dedykowanych paneli kontrolnych.



Rys. 6.12 Panel kontrolny agentów wykorzystywanych w projekcie

Źródło: opracowanie własne

Jak można zauważyć na rysunku 6.12, poddany implementacji model szczegółowy a2M umożliwia także śledzenie postępów w realizacji prac przez powołanie agentów do realizacji określonego projektu. Warto również dodać, że takie panele kontrolne mogą być rozwijane i modyfikowane w zależności od potrzeb określonego agenta. Istnieje także możliwość zdefiniowania dla AGENTA Menedżera panelu kontrolnego tylko dla wybranej grupy agentów. Podobnie dla agentów niższego szczebla można ograniczyć możliwości kontrolne tylko do podległych agentów.

Wnioski

Przeprowadzony eksperyment pozwolił na ocenę możliwości pełnej przydatności modelu szczegółowego a2M do wspomagania zarządzania projektem informatycznym. Wnioski z badania 5-P przedstawiono w tabeli 6.11

Tabela 6.11 Wnioski z badania 5-P

		Kod eksperymentu: 5-P
Realizacja celu badania	Cel badania osiągnięty	
Zidentyfikowane poziomy modelu szczegółowego a2M	definiowania zarządzania adaptacji	Istnieją procesy adaptacji struktur agentowych
Zidentyfikowane warstwy modelu szczegółowego a2M	wiedzy przetwarzania funkcjonalności	Istnieją procesy adaptacji struktur agentowych

Źródło: opracowanie własne

Przeprowadzony eksperyment wykazał możliwość implementacji warstwy funkcjonalnej modelu szczegółowego a2M na poziomie adaptacji. Dobre praktyki zostały poddane procesowi implementacji

jako elementy pracy w narzędziu RTC zostały dopasowane do struktury projektu jako odpowiednie agenty opisywane cechami oraz zachowaniami. Ponadto przeprowadzony eksperyment pokazał możliwości implementacji instrumentów pomiarowych pozwalających na sprawdzenie poziomów dojrzałości członków zespołu czy klienta. Poddano implementacji również arkusz pozwalający określić poziom entropii na podstawie odpowiedzi kierownika projektu (zgodnie ze wzorem przyjętym w załączniku)

Największym problemem związanym z implementacją poziomu adaptacji modelu szczegółowego **a2M** w RTC okazało się jednak zapewnienie automatycznego adaptowania dobrych praktyk zarządzania projektami do struktury projektu w oparciu o zmienne nowego trójkąta ograniczeń. Poddany implementacji model szczegółowy **a2M** wykorzystał zewnętrzny arkusz wnioskujący podłączony do modelu w RTC. Uznano za konieczne rozbudowanie RTC o mechanizm wnioskujący, który na podstawie danych wprowadzonych przez uczestników podczas wypełniania dołączonego do modelu instrumentu pomiarowego, pozwala określić poziomy dojrzałości zespołu oraz klienta a także poziom entropii projektu. Plan rozbudowy RTC o takie mechanizmy wnioskujące potraktowano jako osobny projekt informatyczny w ramach Uniwersyteckiego Centrum Kompetencyjnego Technologii Rational na Politechnice Gdańskiej.

6.2 Podsumowanie

Przeprowadzone eksperymenty związane z implementacją modelu szczegółowego **a2M** w narzędziu wspierającym zarządzanie projektami informatycznymi RTC wykazały, że przyjęte w model szczegółowy **a2M** jest możliwy do implementacji. Po sesji z kierownikiem projektu model szczegółowy **a2M** może być wykorzystywany do wspomaganie zarządzania projektami informatycznymi

W środowisku narzędzia RTC istnieje możliwość implementacji modelu jako szablonu wybieranego alternatywnie przez kierowników w stosunku do metod zarządzania projektami jak SCRUM czy RUP. Dzięki temu kierownik korzystający z szablonu stworzonego dla modelu szczegółowego **a2M** rozpoczyna proces zarządzania projektem informatycznym wg założeń modelu szczegółowego **a2M**. Na początku trwania projektu definiuje elementy struktury projektowej a następnie przeprowadza proces transformacji tych elementów do postaci agentów opisywanych zestawem cech i zachowań. Posiada również możliwość predefiniowania agentów w szablonie modelu szczegółowego **a2M**. Wówczas wybiera tylko te agenty, które odpowiadają projektowi.

W trakcie prowadzonych eksperymentów zweryfikowano także możliwość tworzenia relacji pomiędzy elementami struktury projektowej. Napotkano na przeszkody wynikające z ograniczeń w narzędziu jakim jest RTC. Nie udało się bowiem w sposób intuicyjny dla użytkownika umożliwić tworzenia relacji między agentami. Można te relacje definiować w oparciu o inne funkcjonalności narzędzia (atrybuty, role) lub poprzez zastosowanie wtyczek pobranych z innego narzędzia z rodziny Rational. Następnie zweryfikowano możliwość zarządzania agentami poprzez przesyłanie zleceń (komunikatów).

W ostatnim przeprowadzonym eksperymencie potwierdzono możliwość mierzenia poziomów dojrzałości zespołu dostawcy, klienta oraz entropii projektowej za pomocą zaimplementowanych instrumentów pomiarowych oraz dobieranie dobrych praktyk w oparciu o wartości tych zmiennych. Takie rozwiązanie jest możliwe do implementacji w RTC, jednak wbudowane funkcjonalności narzędzia nie zapewniają automatycznego wnioskowania w oparciu o wprowadzone dane. Zauważono także konieczność usprawnienia RTC o automatyczne wnioskowanie. Takie podejście pozwoli na generowanie wartości zmiennych decyzyjnych traktowanych jako podstawa do automatycznego doboru dobrych praktyk. W ten sposób agent menedżer (albo koordynator) nie będzie samodzielnie analizował wyników, tylko skorzysta z modułu wnioskującego, o który zostanie rozbudowane RTC.

Rozdział VII

Podsumowanie i wnioski

Celem niniejszej rozprawy było opracowanie adaptacyjnego agentowego modelu zarządzania projektami informatycznymi stanowiącego w stosunku do metod zarządzania projektami podejście alternatywne. Do opracowania modelu zastosowano podejście agentowe traktujące wszystkie elementy struktury projektowej (członków zespołu, technologie, zasoby wiedzy oraz dobre praktyki zarządcze) jako niezależne agenty realizujące projekt informatyczny. Zdefiniowano w ramach modelu nowy (uzupełniający w stosunku do istniejącego) trójkąt ograniczeń projektowych obejmujący dojrzałość klienta, dojrzałość zespołu dostawcy oraz entropię projektu. Następnie przyjmując, że zarówno nowy jak i istniejący trójkąt ograniczeń stanowią warunki (właściwego, zgodnego z wymaganiami projektu) doboru dobrych praktyk zarządczych, opracowany model wykorzystano do diagnozy zakończonych projektów, jak też prognozowania procesów zarządczych w nowych i rozpoczętych. Wykazano pełną przydatność modelu do wspomagania procesów zarządzania potwierdzoną badaniami 1-R do 7-R (rozdział V pracy) oraz eksperymentami badawczymi 1-P do 5-P (rozdział VI pracy).

Dla osiągnięcia postawionego celu rozprawa została podzielona na sześć części.

Pierwsza część pracy została poświęcona prezentacji stanu wiedzy z zakresu zarządzania projektami informatycznymi. Zaprezentowano typy projektów informatycznych, jakimi zarządzają kierownicy w branży IT, oraz wskazano na główne problemy decyzyjne kierowników takich projektów. Zaprezentowano również metody zarządzania projektami, które dobierane są przez kierowników projektów. Zwrócono uwagę również, że stosowanie metod zarządzania projektami nie gwarantuje sukcesu projektu (co potwierdzają statystyki dostarczane przez firmy konsultingowe badające stan projektów informatycznych na świecie). Uznano, że dekompozycja metod zarządzania do poziomu dobrych praktyk stworzy warunki do diagnozowania przyczyn niepowodzeń projektów informatycznych oraz stworzy warunki do analizy możliwości implementacji tych praktyk z punktu widzenia proponowanego w pracy modelu wspomagającego procesy zarządcze. Wskazano jednocześnie, że jednym ze źródeł niepowodzeń projektów jest brak dopasowania dobrych praktyk do istniejących ograniczeń projektowych (trójkąta ograniczeń: budżet, harmonogram, zakres). Uznano, że dobór dobrych praktyk przez kierowników nie powinien wynikać ze specyfiki projektu informatycznego, którą stanowią nowe dyskutowane i proponowane w rozdziale drugim ograniczenia projektowe.

Dlatego też w drugiej części pracy, uznając, że szereg decyzji podejmowanych przez kierowników zapada przed rozpoczęciem realizacji projektu, wprowadzono nowy, uzupełniający w stosunku do istniejącego trójkąt ograniczeń. W tym celu wyodrębniono zagregowane zmienne decyzyjne jak dojrzałość zespołu dostawcy, dojrzałość klienta oraz entropię projektu. W rozdziale drugim wykazano również konieczność pomiaru wartości tych zmiennych w celu późniejszego dopasowania dobrych praktyk. Rozdział zakończono prezentacją funkcji dopasowania (A_MATCH), która pokazała stopień dopasowania wyodrębnionych w rozdziale pierwszym dobrych praktyk względem poziomów zmiennych decyzyjnych. Na podstawie wyników osiągniętych w tej części pracy uznano za konieczne opracowanie modelu, który ułatwi kierownikom projektów informatycznych dobieranie dobrych praktyk w ramach istniejącego i proponowanego w pracy trójkąta ograniczeń projektowych.

W trzeciej części pracy omówiono założenia do budowy adaptacyjnego modelu zarządzania projektami informatycznymi, prezentując na wstępie metodę budowy modelu oraz etapy jego budowy. Zaprezentowano podejście agentowe oraz procesowe jako stwarzające warunki do implementacji

procesów zarządzania (implementacji procesów i struktury projektowej). Przedstawiono etapy budowy modelu z wykorzystaniem hybrydowego podejścia agentowo-procesowego i pokazano etapy rozwojowe budowy adaptacyjnego agentowego modelu zarządzania projektami informatycznymi **a2M** (submodeli MAS, MAS-FUNK oraz MAS-WEB). Zwrócono uwagę, że model **a2M** powinien mieć nie tylko wymiar implementacyjny prowadzący do stworzenia decyzyjnego systemu wsparcia bazującego na architekturze modelu agentowego, ale również wymiar organizacyjny. Uznano za konieczne rozpatrywanie modelu także w wymiarze organizacyjnym, w którym dokonuje się działań porządkujących funkcjonowanie poszczególnych elementów struktury projektowej. Przyjęto organizacyjno-implementacyjny charakter modelu i traktowania go jako modelu adaptacyjnego.

W czwartej części pracy zaprezentowano adaptacyjny agentowy model zarządzania projektami z uwzględnieniem jego dualnego charakteru. Model zaprezentowano w postaci ogólnej oraz w postaci szczegółowej. W postaci szczegółowej model został podzielony na 3 poziomy — definiowania, zarządzania i adaptacji procesów zarządczych. Na pierwszym poziomie porządkowane są elementy struktury projektowej i definiowane jako agenty reprezentowane przez zestawy cech i zachowań. Na poziomie drugim modelu stworzono podstawowe relacje pomiędzy elementami struktury projektowej. Zidentyfikowano relacje hierarchiczne między agentami oraz przepływy zleceń wykonania określonych zadań przesyłanych w postaci komunikatów. Na poziomie trzecim modelu (poziom adaptacji) przeprowadzono procesy dopasowania dobrych praktyk zarządzania projektami do zmiennych stworzonego na potrzeby struktury projektowej nowego trójkąta ograniczeń.

Po opracowaniu modelu **a2M** postaci szczegółowej przedstawiono w części piątej procesy badania zasadności replikatywnej. Przeprowadzono serię siedmiu badań, których celem było sprawdzenie, na ile model **a2M** może być wykorzystany do diagnozy zakończonych projektów, w których uczestniczył autor rozprawy. Badania miały na celu wykazanie, na ile warstwy i poziomy modelu miały odzwierciedlenie w procesach zarządzania projektami i na ile możliwe było zastosowanie w zakończonych projektach struktur agentowych.

W szóstej części zbadano możliwość implementacji modelu do prognozowania procesów zarządzania projektami. W tym celu przeprowadzono w środowisku realizowanych projektów serię pięciu eksperymentów badawczych. Dwa z nich poświęcone były prognozowaniu składu zespołu projektowego i definiowania procesów zarządzania nimi, trzy miały na celu wykazanie możliwości implementacji modelu **a2M** do struktur agentowych. Do tego celu wybrano narzędzie do zarządzania projektami informatycznymi z rodziny Rational (Rational Team Concert), a następnie odwzorowywano w narzędziu poziomy i warstwy modelu szczegółowego **a2M**. Dla wykorzystania modelu przez kierowników projektów stworzono szablon **a2M**, z którego kierownicy mogą korzystać podczas zarządzania własnymi projektami.

Niniejszą rozprawę zamyka część zawierająca załączniki. Nie stanowią one integralnej części pracy, ale zawierają procedury postępowania opracowane podczas realizacji prac badawczych na potrzeby niniejszej rozprawy. Dlatego też w załącznikach do pracy przedstawiono sposób kwantyfikacji dojrzałości klienta, dojrzałości zespołu dostawcy oraz entropii projektowej. Zaprezentowano instrumenty pomiarowe w postaci kwestionariuszy oraz sposób interpretacji wyników pozwalający kierownikom projektów określić poziomy zmiennych decyzyjnych tworzących nowy trójkąt ograniczeń.

Przeprowadzone badania (rozdział piąty pracy) oraz zrealizowane eksperymenty (rozdział szósty pracy) nad adaptacyjnym agentowym modelem zarządzania projektami pozwoliły na weryfikację głównej hipotezy pracy.

„Istnieje możliwość implementacji procesów zarządzania projektami do postaci agentowego adaptacyjnego modelu zarządzania projektami informatycznymi. Model ten stanowi alternatywne (w stosunku do istniejących metod) podejście do zarządzania projektami informatycznymi.”

Hipoteza ta (jak również dwie hipotezy pomocnicze) została potwierdzona dzięki zrealizowaniu celu pracy (poprzez realizację dwu celów szczegółowych), czyli budowie adaptacyjnego agentowego modelu zarządzania przedsięwzięciami informatycznymi (rozdział IV pracy) i jego zastosowaniu w środowisku projektów informatycznych (rozdziały piąty i szósty).

Po opracowaniu modelu oraz jego weryfikacji nasunęło się kilka uwag dotyczących wykorzystania modelu. Pierwsza, wskazująca na ile model **a2M** bazujący na dopasowywaniu dobrych praktyk do nowego trójkąta ograniczeń jest rozwiązaniem kompletnym. Zastanawiano się, czy problematyka zarządzania projektami informatycznymi nie jest na tyle złożona, że zagregowane zmienne decyzyjne nowego trójkąta ograniczeń nie oddają tej złożoności. Wydaje się, że trudno jednoznacznie stwierdzić, czy opracowany model oraz koncepcja pomiaru zmiennych decyzyjnych (rozdział III pracy) odnoszących się do entropii projektu, dojrzałości zespołu dostawcy oraz dojrzałości klienta agreguje wszystkie problemy decyzyjne kierowników projektów. Uznano jednak, że trzy zagregowane zmienne określone na podstawie procesów zarządzania projektami i dobrych praktyk (rozdział I i II tej pracy) są na tyle pojemne, że agregują w sobie większość typowych problemów kierowników projektów. Potwierdzeniem tych uwag i wątpliwości jest fakt, że podczas prezentacji modelu przed doświadczonymi kierownikami projektów (m.in. z firm takich jak HP czy IBM) zwracali oni uwagę, że borykają się właśnie z problemami, jakie skupiają się wokół tych trzech zmiennych decyzyjnych.

Drugą uwagą, jaka nasuwa się podczas opracowywania niniejszego modelu, jest kwestia ryzyka projektowego. Zagadnienie ryzyka projektowego jest jednym z bardziej problematycznych dla kierowników projektów. Jest ono trudne do zmierzenia, a jego identyfikacja również dostarcza kierownikom projektów problemów. Nasuwa się pytanie, dlaczego nie poświęcono ryzyku projektowemu szczególnej uwagi w opracowanym modelu. Odpowiedzią na tę kwestię jest niewątpliwie potrzeba kwantyfikacji zmiennych decyzyjnych. Określając bowiem dojrzałość zespołu dostawcy, dojrzałość klienta oraz entropię projektu, a następnie dobierając dobre praktyki, dokonuje się minimalizacji niektórych kategorii ryzyka (zawartych m.in. w entropii projektu — rozdział III pracy). Adaptacja dobrych praktyk (proces modelu **a2M**) eliminuje ryzyko organizacyjne projektu (ryzyko wynikające z niedopasowania metody zarządzania do klienta czy członków zespołu). Pomiar entropii projektowej nie eliminuje co prawda ryzyka, ale dostarcza wystarczającą wiedzę kierownikowi o tym, czy możliwe jest podejmowanie decyzji w długim czy krótkim okresie. Wydaje się również, że identyfikowanie struktur projektowych oraz ich parametryzowanie za pomocą agentów (cech i zachowań) odniesione nie tylko do członków zespołu, ale również do zasobów wiedzy i technologii, ogranicza ryzyko techniczne realizacji projektu. Skoro bowiem ryzyko techniczne wynika najczęściej z niedopasowania technologii do realizowanego projektu, to dobór właściwych agentów (integrujących członków zespołu i technologie) je ogranicza. Model jest także odpowiedzią na problematykę ryzyka biznesowego. Analiza dojrzałości klienta (zmienna nowego trójkąta ograniczeń) nie wskazuje na sytuację wycofania się klienta z projektu, ale pozwala na dobór takich dobrych praktyk, które eliminują możliwość nieporozumień z klientem.

Uwagi, które pojawiły się podczas budowy modelu, dotyczą również sposobu pomiaru zmiennych decyzyjnych. Na potrzeby niniejszej rozprawy przyjmowano wartości ostre, które następnie przekładano na wartości lingwistyczne (mały, średni, duży). Wydaje się, że kwestia pomiaru dojrzałości klienta czy zespołu oraz mierzenie entropii projektowej powinna zostać rozszerzona o mechanizmy np. modelowania rozmytego. Wydaje się również, że opracowane instrumenty pomiarowe powinny poddawać się stałej weryfikacji przez kierowników projektów w celu dalszych udoskonaleń.

Warto także rozważyć relacje pomiędzy elementami struktury projektowej, a zwłaszcza relacji między członkami zespołu projektowego uczestniczącymi w projektach. Zastanawiano się już podczas badania zasadności replikatywnej, czy oprócz typowych relacji hierarchicznych, nie powinno się uwzględnić w modelu relacji nieformalnych. Obserwowano bowiem, że na jakość pracy zespołu

dostawcy bardzo często wpływają wzajemne relacje dotyczące koleżeństwa czy wspólnych zainteresowań. Uchwycenie tych relacji jest jednak tak złożone, że nie udało się ich odzwierciedlić w modelu. Można tworzyć pewne kategorie instancji niemerytorycznych, które mają znaczenie dla funkcjonowania zespołów projektowych. Takie podejście pozwala na oddanie dokładnego obrazu relacji tworzących się między elementami struktury projektowej. Analiza takich relacji wymaga jednak włączenia ekspertów zarówno z zakresu zarządzania wiedzą czy kapitałem intelektualnym, jak również psychologii bądź socjologii.

Zwrócono także uwagę na kwestie implementacyjne modelu. Pojawiło się bowiem pytanie, na ile kierownicy projektów będą zainteresowani wykorzystaniem zbudowanego modelu **a2M** i poddanego implementacji w RTC. Pierwsze prezentacje modelu na konferencjach branżowych wykazywały, że model został zaakceptowany przez środowiska IT w kraju (Konferencja Komputerowo Zintegrowane Zarządzanie, Zakopane 2008 i 2009). Być może należy na stałe włączyć model do narzędzia RTC jako szablon dostępny dla korzystających z narzędzia kierowników. Na tym etapie prowadzenia badań nie planuje się komercjalizacji modelu, a raczej jego promowanie w formie ogólnodostępnej za pośrednictwem serwisu WWW. Wydaje się bowiem, że akceptacja modelu przez kierowników projektów, m.in. z branży IT oraz informatycznych organizacji wsparcia (świadczących usługi dla banków czy instytucji pożytku publicznego) stwarza warunki do jego modyfikacji i pełnego dostosowania do wymagań kierowników projektów.

Reasumując, warto podkreślić, że opracowany model w odczuciu autora stanowi odpowiedź na problemy zarządzania projektami informatycznymi wyodrębnione na początku niniejszej pracy. Stanowi również próbę rozwiązania problemów decyzyjnych kierowników, sugerując dopasowywanie dobrych praktyk do projektu, a nie dopasowywanie projektu do konkretnej metody, jak ma to miejsce zazwyczaj.

Opracowany w niniejszej pracy model powstał w Zakładzie Zarządzania Technologiami Informatycznymi Politechniki Gdańskiej, którego zadaniem jest propagowanie rozwiązań usprawniających zarządzanie technologiami informatycznymi, w tym także technologii wspomagających zarządzanie projektami informatycznymi. Model ten był wykorzystywany do realizacji projektów prowadzonych bądź koordynowanych przez Zakład na przestrzeni lat 2006–2011. Należy także zwrócić uwagę, że obecnie Zakład realizuje 11 nowych projektów (m.in. rozbudowa centrów kompetencyjnych, budowa centrum przetwarzania danych, rozwój oprogramowania HP Service Manager, rozwój narzędzi klasy ERP — MS Dynamics AX) z wykorzystaniem opracowanego modelu agentowego prezentowanego w niniejszej rozprawie

Spis literatury

1. Alkhatib G., *Agent Technologies and Web Engineering*, Ideal Group Publishing, Hershey, 2008
2. Alonso E., Kudenko D., *Adaptive Agents and Multi-Agent Systems II: Adaptation and Multi-Agent Learning*, Springer, Berlin, 2008
3. Angryk R., Galant V., Paprzycki I. M., *Travel Support System - an Agent-Based Framework, Proceedings of the International Conference on Internet Computing*, CSREA Press, Las Vegas, 2002,
4. Barley M. W., *Intelligent Agents and Multi-Agent Systems: 7th Pacific Rim International Workshop on Multi-Agents, Prima 2004, Auckland, New Zealand, August 8-13, 20*, Springer, Berlin, 2008
5. Bass L., Clements P., Kazman R., *Software Architecture in Practice*, Addison-Wesley, Boston, 2003.
6. Bazzan A. L. C., Klugl F., *Multi-Agent Systems for Traffic and Transportation Engineering*, Information Science Publishing, Hershey, 2009
7. Belbin M., *Twoja rola w zespole*, Gdańskie Wydawnictwo Psychologiczne, Gdańsk, 2008
8. Bergenti F., *Methodologies and Software Engineering for Agent Systems: The Agent-Oriented Software Engineering Handbook*, Springer, Berlin, 2004
9. Bergstrom S., Reberg L., *Adopting the Rational Unified Process: Success with the Rup*, Addison-Wesley Professional, Boston, 2003
10. Białynicki-Birula I., Białynicka-Birula Z., *Modelowanie rzeczywistości*, Prószyński i S-ka, Warszawa, 2002.
11. Bigus, J. P., Bigus, J., *Constructing Intelligent Agents Using Java: Professional Developer's Guide*, Wiley, Chichester, 2001
12. Boehm B. W. and Papaccio P. N., Understanding and controlling software costs. *IEEE Transactions on Software Engineering*, vol. 14, no. 10, ss. 1462-1477, 1988.
13. Bordini R. H., Wooldridge M., *Programming Multi-Agent Systems in AgentSpeak Using Jason*, Wiley-Interscience, Chichester, 2007
14. Cadle J., Yeates D., *Zarządzanie procesem tworzenia systemów informacyjnych*, Wydawnictwa Naukowo-Techniczne, Warszawa, 2004
15. Chmielewski J. M., Waćkowski K., *Wspomaganie zarządzania projektami informatycznymi. Poradnik dla menedżerów*, Helion, Gliwice, 2007
16. Clements P., Bachmann F., Bass L., Garlan D., Ivers J., Reed L., Merson P., Nord R., Stafford J., *Documenting Software Architectures: Views and Beyond*, Addison-Wesley, Boston, 2010.
17. Cooper K., Torczon L., *Engineering a Compiler*, 2nd Edition, Morgan Kaufmann, San Francisco, 2011.
18. Czarnecki A., Orłowski C., Sitek T., Ziółkowski A., *Information technology assessment using a functional prototype of the agent based system*, [w:] Foundations of Control and Management Sciences, Wydawnictwo Politechniki Poznańskiej, Poznań, 2008
19. Czarnecki A., Orłowski C., Sitek T., Ziółkowski A., *Ocena technologii informatycznych z wykorzystaniem prototypu funkcjonalnego systemu agentowego*, [w:] Zarządzanie Przedsiębiorstwem, Gdańsk, 2008
20. Czarnecki A., Orłowski C., Ziółkowski A., *Validation of an Agent and Ontology-based Information Technology Assessment System*, [w:] Cybernetics and Systems. - Vol. 41, Iss. 1, 2010

21. Dąbrowski W., Subieta K., *Podstawy inżynierii oprogramowania*, Wydawnictwo PWJSTK, Warszawa 2005
22. Dastani M., *Programming Multi-Agent Systems*, Springer, Berlin, 2008
23. Flasiński M., *Zarządzanie projektami informatycznymi*, PWN, Warszawa 2006.
24. Frączkowski K., *Zarządzanie projektem informatycznym: projekty w środowisku wirtualnym, czynniki sukcesu i niepowodzeń projektów*, Oficyna Wydawnicza PW, Wrocław, 2003
25. Galant V., Tubyrcy I. J., *Inteligentny Agent Programowy, [The Intelligent Programme Agent]* Prace Naukowe AE, Wrocław, 2001,
26. Garcia A., *Software Engineering for Multi-Agent Systems IV*, Springer, Berlin, 2006
27. Górski J. (red.), *Inżynieria oprogramowania w projekcie informatycznym*, Mikom, Warszawa, 2005
28. Heerkens G. R., *Jak zarządzać projektami?*, Wydawnictwo RM, Warszawa, 2003
29. Henderson-Sellers B., Gorton I., *Agent-based Software Development Methodologies. International Conference on Object-Oriented Programming, Systems, Languages and Applications*, OOPSLA, Seattle USA, 2002
30. Hendler J., *Is There an Intelligent Agent in Your Future?*, Nature, 11, March 1999
31. Jones R., *Zarządzanie projektami. Sztuka przetrwania*, MT Biznes Sp. z o.o., Warszawa, 2009.
32. Koszłajda A., *Zarządzanie projektami IT. Przewodnik po metodykach*, Helion, Gliwice, 2010
33. Kroll P., Krutchen P., *Rational Unified Process od strony praktycznej*, Wydawnictwa Naukowo-Techniczne, Warszawa, 2006
34. Kroll P., Macisaac B., *Agility and Discipline Made Easy: Practices from OpenUP and RUP*, Addison-Wesley Professional, Boston, 2006
35. Krutchen P., *Rational Unified Process od strony teoretycznej*, Wydawnictwa Naukowo-Techniczne, Warszawa, 2006
36. Kuwabara K., *Intelligent Agents and Multi-Agent Systems: 5th Pacific Rim International Workshop on Multi-Agents, Prima 2002, Tokyo, Japan, August 18-19, 2002. Proc*, Springer, Berlin, 2008
37. Leffingwell D., Widrig D., *Zarządzanie wymaganiami*, WNT, Warszawa, 2005
38. Lock D., *Podstawy zarządzania projektami*, Polskie Wydawnictwo Ekonomiczne, Warszawa, 2009
39. Madej P., Orłowski C., Ziółkowski A., *Modele integracji systemów informatycznych przedsiębiorstw*, [w:] : Inżynieria oprogramowania w procesach integracji systemów informatycznych / eds. Janusz Górski, Cezary Orłowski, Pomorskie Wydawnictwo Naukowo-Techniczne, Gdańsk, 2010
40. Madej P., Orłowski C., Ziółkowski A., *Models of Information Systems Integration in Companies*, [w:] Information Systems Architecture and Technology : IT Models in Management Process/ eds. Zofia Wilimowska, Leszek Borzowski, Adam Grzech, Jerzy Świątek, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, 2010
41. Marik V., *Multi-Agent Systems and Applications III*, Springer, Berlin, 2007
42. Nguyen N. T., *Computational collective intelligence and knowledge inconsistency in multi-agent environments*, [w:] Lecture Notes in Computer Science. Lecture Notes in Artificial Intelligence, vol. 5357, Berlin, 2008
43. Nguyen N. T., Szczerbicki E., *Intelligent systems for knowledge management*, Springer, Berlin, 2009
44. Nguyen N. T., Trawiński B., *New challenges for intelligent information and database systems*, Springer, Berlin, 2011
45. Orłowski C., Szczygielski Ł., Ziółkowski A., *Implementacja procesów zarządzania zmianami w przedsiębiorstwie*, [w:] Inżynieria oprogramowania w procesach integracji systemów

- informatycznych / eds. Janusz Górski, Cezary Orłowski, Pomorskie Wydawnictwo Naukowo-Techniczne, Gdańsk, 2010
46. Orłowski C., Szczygielski Ł., Ziółkowski A., *Implementing Change Management Processes in an Enterprise*, [w:] : Information Systems Architecture and Technology, IT Models in Management Process / eds. Zofia Wilimowska, Leszek Borzemski, Adam Grzech, Jerzy Świątek, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, 2010
 47. Orłowski C., Ziółkowski A., *An Approach to Agent-Based Supporting System for IT Projects*, [w:] Agent and Multi-Agent Systems: Technologies and Applications: 4th KES International Symposium, Gdynia, 2010
 48. Orłowski C., Ziółkowski A., *Concept of the agent system for the information technology evaluation*, [w:] Information systems architecture and technology : information systems and computer communication networks / ed. Adam Grzech, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, 2007
 49. Orłowski C., Ziółkowski A., *Definicja zadań inteligentnych agentów do oceny technologii informatycznych*, [w:] Komputerowo zintegrowane zarządzanie. T. 2 / pod red. Ryszarda Knosali, : Oficyna Wydawnicza Polskiego Towarzystwa Zarządzania Produkcją, Opole, 2008
 50. Orłowski C., Ziółkowski A., *Evolution of concept of agent system for information technology evaluation*, [w:] Information systems architecture and technology : information systems and computer communication networks / editors: Adam Grzech, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, 2008
 51. Orłowski C., Ziółkowski A., *Koncepcja zastosowania systemów agentowych do oceny technologii informatycznych*, [w:] Komputerowo zintegrowane zarządzanie. T. 2 / pod red. Ryszarda Knosali, Oficyna Wydawnicza Polskiego Towarzystwa Zarządzania Produkcją, Opole, 2007
 52. Orłowski C., Ziółkowski A., *Ocena technologii informatycznych z wykorzystaniem prototypu funkcjonalnego systemu agentowego*, [w:] Zarządzanie wiedzą i technologiami informatycznymi [Dokument elektroniczny] / redakcja: Cezary Orłowski, Zdzisław Kowalczyk, Edward Szczerbicki, Pomorskie Wydawnictwo Naukowo-Techniczne, Gdańsk, 2008
 53. Orłowski C., Ziółkowski A., *Supporting Software Project Management Processes Using the Agent System*, [w:] Knowledge-Based and Intelligent Information and Engineering Systems: KES 2010, 14th International Conference, Cardiff, 2010
 54. Orłowski C., Ziółkowski A., *Verification of agent system for it project management support*, [w:] Information systems architecture and technology : IT technologies in knowledge oriented management process / editors: Zofia Wilimowska, Leszek Borzemski, Adam Grzech, Jerzy Świątek, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, 2009
 55. Orłowski C., Ziółkowski A., *Weryfikacja agentów systemu agentowego do oceny technologii informatycznych*, [w:] Komputerowo zintegrowane zarządzanie. T. 2 / pod red. Ryszarda Knosali, Oficyna Wydawnicza Polskiego Towarzystwa Zarządzania Produkcją, Opole, 2009
 56. Orłowski C., Ziółkowski A., *Wytwarzanie architektury korporacyjnej jako środowisko weryfikacji systemu agentowego do oceny technologii informatycznych*, [w:] Zastosowanie technologii informatycznych w zarządzaniu wiedzą / redakcja: Cezary Orłowski, Zdzisław Kowalczyk, Edward Szczerbicki, Pomorskie Wydawnictwo Naukowo-Techniczne, Gdańsk, 2009
 57. Orłowski C., Ziółkowski A., *Zarządzanie przedsięwzięciami informatycznymi wspomagane systemem agentowym*, [w:] Komputerowo zintegrowane zarządzanie, t.2/ ed. Ryszard Knosala, Oficyna Wydawnicza Polskiego Towarzystwa Zarządzania Produkcją, Opole, 2010

58. Parsons S., *Game Theory and Decision Theory in Agent-Based Systems*, Kluwer Academic Publishers, Dordrecht, 2002
59. Parth F., Snyder C., *Introduction to IT Project Management*, Management Concepts, Vienna, 2007
60. Pawlak M., *Zarządzanie projektami*, Wydawnictwo Naukowe PWN, Warszawa, 2006,
61. Pechoucek M., Petta P., *Multi-Agent Systems and Applications IV: 4th International Central and Eastern European Conference on Multi-Agent Systems, Ceemas 2005, Budapest, Hung*, Springer, Berlin, 2007
62. Phillips J., *Zarządzanie projektami IT*, Helion, Gliwice, 2005.
63. Pressman R., *Praktyczne podejście do inżynierii oprogramowania*”, WNT, Warszawa, 2004
64. Project Management Institute, *The Project Management Body of Knowledge*, 2008
65. Sacha K., *Inżynieria oprogramowania*, PWN, Warszawa, 2010
66. Schumacher M., *Objective Coordination in Multi-Agent System Engineering*, Springer, Berlin, 2007
67. Schwaber K., Sutherland J., *Przewodnik po SCRUMIE: Reguly Gry*, Lipiec, 2011
68. Schwalbe K., *Information technology Project Management*, Course Technology, Boston, 2010
69. Skrorulski M., *IBM Rational Method Composer – portal procesowy*, Zespół Przed wszystkim, grudzień 2007
70. Sommerville I., *Inżynieria oprogramowania*, WNT, Warszawa, 2003
71. Stackpole C., *A User's Manual to the PMBOK Guide*, Wiley, New Jersey, 2010
72. Swithinbank P., Chessell M., Gardner Dr. T., Griffin C., Man J., Wylie H., Yusuf L., *Patterns: Model-Driven Development Using IBM Rational Software Architect*, IBM, Zurich, 2005
73. Szwabe M. (red.), *Zarządzanie projektami współfinansowanymi z funduszy publicznych*, Wolters Kluwer, Warszawa, 2005
74. Szyjewski Z., *Metodyki zarządzania projektami informatycznymi*, Placet, Warszawa, 2004
75. Szyjewski Z., *Zarządzanie projektami informatycznymi*, Placet, Warszawa, 2001
76. Thompson S. G., *Defence Applications of Multi-Agent Systems: International Workshop, Damas 2005, Utrecht, the Netherlands, July 25, 2005, Revised and Invited Papers*, Springer, Berlin, 2006
77. Trocki M., Grucza B., Ogonek K., *Zarządzanie projektami*, Polskie Wydawnictwo Ekonomiczne, Warszawa, 2009
78. Vouros G., Artikis A., Stathis K., *Organized Adaption in Multi-Agent Systems: First International Workshop, Oamas 2008, Estoril, Portugal, May 13, 2008. Revised and Invited Papers Organized Adaption in Multi-Agent Systems: First International Workshop, Oamas 2008, Estoril, Portugal, May 13, 2008. Revised and Invited Papers*, Springer, Berlin, 2009
79. Wachowiak P. (red.), *Pomiar kapitału intelektualnego przedsiębiorstwa [Measuring the intellectual capital of a company]*, Szkoła Główna Handlowa w Warszawie, Warszawa, 2005
80. Waćkowski K., Chmilewski J. M., *Wspomaganie zarządzania projektami informatycznymi. Poradnik dla menadżerów*, Helion, Gliwice, 2007
81. Wei G., *Adaptation and Learning in Multi-Agent Systems: Ijcai' 95 Workshop, Montreal, Canada, August 21, 1995. Proceedings*, Springer, Berlin, 2008
82. Weiss G., *Multi-Agent Systems*, MIT Press, Massachusetts, 1999.
83. Weyns D., *Architecture-based Design of Multi-agent Systems*, Springer, Berlin, 2010
84. Wiegers K. E., *Software requirements. Second edition*, Microsoft Press, Washington, 2003
85. Wróblewski P., *Zarządzanie projektami informatycznymi dla praktyków*, Helion, Gliwice, 2005
86. Wrycza S., *Informatyka ekonomiczna*, PWE, Warszawa, 2010

87. Wrycza S., *Projektowanie systemów informatycznych*, Wydawnictwo UG, Gdańsk, 1997.
88. Ying J.-J., *Principles of Practice in Multi-Agent Systems: 12th International Conference, PRIMA 2009, Nagoya, Japan, December 14-16, 2009, Proceedings*, Springer, Berlin, 2009
89. Ziółkowski A., *Ocena narzędzi wspomagających zarządzanie przedsięwzięciami informatycznymi na przykładzie IBM Rational Method Composer*, [w:] Zarządzanie technologiami informatycznymi : przykłady zastosowań IT / pod redakcją Cezarego Orłowskiego, Pomorskie Wydawnictwo Naukowo-Techniczne, Gdańsk, 2007
90. Ziółkowski A., Orłowski C., *Definicja zadań inteligentnych agentów do oceny technologii informatycznych* [The definition of the tasks of intelligent agents in the evaluation of IT], Komputerowo Zintegrowane Zarządzanie, t.2, Oficyna Wydawnicza Polskiego Towarzystwa Zarządzania Produkcją, Opole, 2008
91. Ziółkowski A., *Systemy agentowe – cechy, zastosowanie oraz przegląd narzędzi do ich tworzenia*, [w:] Zarządzanie technologiami informatycznymi : stan i perspektywy rozwoju / pod redakcją Cezarego Orłowskiego, Pomorskie Wydawnictwo Naukowo-Techniczne, Gdańsk, 2006
92. Żytniewski M., *Metodyki budowy hybrydowych systemów wieloagentowych Systemy Wspomagania Organizacji SWO'2008*, AE, Katowice, 2008

Źródła internetowe

93. http://www.4pm.pl/artykul/prince2%E2%BE_dla_it_fakty_i_mity-56-1497.html <30.01.2012 r.>
94. <http://www.prince-officialsite.com/> <22.01.2012 r.>
95. <http://www.prince-officialsite.com/nmsruntime/saveasdialog.asp?IID=603&SID=200> <21.01.2012 r.>
96. <http://www.anixe.pl/scrum.htm> <19.01.2012 r.>
97. Marcin Szczotok, *Techniki lekkie w tradycyjnym podejściu do zarządzania projektem IT*, www.pmanager.pl <18.01.2012 r.>
98. Kuliński R., *Inicjowanie i prowadzenie projektów innowacyjnych w firmie, 4PM*, 2008
99. <http://lean-management.pl/> <10.01.2012 r.>
100. Ganzha M., Gawinecki M., Kobzdej P., Paprzycki M., *Modelowy agentowy system e-commerce*, www.ibspan.waw.pl/~paprzyck/mp/cvr/.../ECOM_2008_PL.pdf <12.01.2012r.>
101. http://brasil.cel.agh.edu.pl/~10sdczerner/page/programowanie_ekstremalne_XP <02.02.2012r.>
102. *Sport dla programistów, czyli programowanie ekstremalne*, www.wtoman.awardspace.com/articles/xp.pdf <29.01.2012 r.>
103. <https://www.jazz.net> <20.01.2012 r.>

Spis rysunków

Rys. 1.1 Klasyczny trójkąt ograniczeń.....	10
Rys. 1.2 Schemat RUP.....	18
Rys. 1.3 Konstrukcja organizacji projektowej wg PRINCE2.....	24
Rys. 1.4 Proces realizacji projektu wg metody SCRUM.....	27
Rys. 1.5 Tablica informacyjna projektu realizowanego wg metody SCRUM (<i>Scrumboard</i>).....	28
Rys. 1.6 Realizacja projektu wg metody XP.....	31
Rys. 1.7 Realizacja projektu wg PMBoK.....	33
Rys. 1.8 Przykładowa wizualizacja procesu w PMBoK.....	34
Rys. 2.1 Nowy trójkąt ograniczeń a klasyczny trójkąt ograniczeń.....	40
Rys. 2.2. Macierz dopasowania i odpowiedniości klienta.....	44
Rys. 3.1 Metoda prowadzenia badań dla uzyskania modelu.....	60
Rys. 3.2 Źródła wiedzy o projektach informatycznych wykorzystane w opracowaniu modelu a2M ...	61
Rys. 3.3 Transformacja głównych problemów w projektach na zmienne decyzyjne.....	62
Rys. 3.4 Agent programowy a agent projektowy.....	68
Rys. 3.5 Ogólna koncepcja modelu systemu agentowego do zarządzania technologiami (architektura submodelu MAS).....	70
Rys. 3.6 Przepływ zadań pomiędzy agentami.....	73
Rys. 3.7 Zakres działania agentów w submodelu MAS_FUNK.....	73
Rys. 3.8 Struktura hierarchiczna agentów submodelu MAS_FUNK.....	74
Rys. 3.9. Prototyp funkcjonalny - submodel MAS_FUNK.....	74
Rys. 3.10 Kontakt z użytkownikiem w submodelu MAS_WEB.....	76
Rys. 3.11 Okno zapytań i wygenerowana sugestia dotycząca metody w submodelu MAS_WEB.....	77
Rys. 3.12 Submodel przetwarzania wstępnego dla generowania decyzji.....	79
Rys. 3.13 Dualny charakter adaptacyjnego agentowego modelu zarządzania projektami informatycznymi.....	82
Rys. 4.1 Model a2M w postaci ogólnej.....	90
Rys. 4.2. Model w postaci szczegółowej.....	96
Rys. 4.3 Poziom definiowania modelu szczegółowego a2M.....	97
Rys. 4.4 Poziom zarządzania modelem szczegółowego a2M.....	99
Rys. 4.5 Poziom adaptacji modelu a2M.....	102
Rys. 5.1 Konstrukcja modelu po badaniu zasadności replikatywnej.....	130
Rys. 6.1 Zdefiniowanie obszaru roboczego dla modelu szczegółowego a2M (model szczegółowy a2M jako szablon w RTC).....	143
Rys. 6.2 Definiowanie elementów struktury projektu wg założeń modelu szczegółowego a2M.....	144
Rys. 6.3 Implementacja dobrych praktyk w modelu szczegółowego a2M.....	145
Rys. 6.4 Definiowanie atrybutów elementu struktury projektowej (eksperta).....	146
Rys. 6.5 Definiowanie agentów w RTC (cechy i zachowania).....	146
Rys. 6.6 Odzwierciedlenie relacji pomiędzy przykładowymi agentami.....	149
Rys. 6.7 Mechanizm przesyłania komunikatów między agentami za pośrednictwem RTC.....	150
Rys. 6.8 Sposób uaktywnienia agenta do wykonania zadania.....	151
Rys. 6.9 Metoda tworzenia formularzy w RTC.....	152
Rys. 6.10 Wybór szablonu modelu szczegółowego a2M przez kierownika projektu.....	155
Rys. 6.11 Przykład implementacji instrumentu pomiarowego do oceny dojrzałości zespołu dost.....	156
Rys. 6.12 Panel kontrolny agentów wykorzystywanych w projekcie.....	158

Spis tabel

Tab. 1.1 Klasyfikacja projektów informatycznych wraz z możliwymi produktami	9
Tab. 1.2 Klasyfikacja ryzyka w projektach informatycznych.....	12
Tab. 1.3 Podstawowe dobre praktyki zawarte w metodach klasycznych	14
Tab. 1.4 Podstawowe dobre praktyki w metodach lekkich.....	15
Tab. 1.5 Wybrane dobre praktyki zarządzania projektami informatycznymi na podstawie RUP..	20
Tab. 1.6 Wybrane dobre praktyki zarządzania projektami informatycznymi wg PRINCE2	25
Tab. 1.7 Dobre praktyki wg metody SCRUM	29
Tab. 1.8 Dobre praktyki wg metody XP.....	32
Tab. 1.9 Zalecenia do konkretnych grup procesów w PMBoK	34
Tab. 1.10 Dobre praktyki w PMBoK	35
Tab. 2.1 Funkcja A_MATCH	56
Tab. 3.1 Wpływ komunikatów na agenty w submodelu MAS_FUNK	73
Tab. 4.1 Klasyfikacja agentów modelu a2M	89
Tab. 4.2 Przykłady usprawnionych decyzji projektowych przez zastosowanie modelu ogólnego	92
Tab. 4.3 Warstwy implementacyjne i oczekiwane produkty dla poziomu definiowania modelu szczegółowego a2M	98
Tab. 4.4 Agenty definiowane w warstwie definiowania modelu szczegółowego a2M.....	99
Tab. 4.5 Warstwy implementacyjne i oczekiwane produkty dla poziomu zarządzania modelem szczegółowego a2M	100
Tab. 4.6 Etap III Warstwy implementacyjne i oczekiwane produkty dla poziomu adaptacji modelu szczegółowego a2M.....	103
Tab. 7.1 Skala do pomiaru doświadczenia zespołu	171
Tab. 7.2 Tabela odpowiedzi.....	173
Tab. 7.3 Tabela objaśniająca skróty z tabeli wynikowej	173
Tab. 7.4 Ogólny wynik pomiaru dojrzałości zespołu	174
Tab. 7.5 Możliwe poziomy dojrzałości klienta	179
Tab. 7.6 Tabela do interpretacji wyników	182

ZAŁĄCZNIKI

ZAŁĄCZNIK 1

Propozycja pomiaru DOJRZAŁOŚCI ZESPOŁU DOSTACY

W celu pomiaru dojrzałości zespołu dostawcy zaproponowano najpierw badanie dojrzałości poszczególnych członków a następnie na podstawie indywidualnych wyników, określenie poziomu dojrzałości całego zespołu.

W celu badania dojrzałości członka zespołu projektowego proponuje się określenie preferencji odnośnie pełnienia funkcji (pozycji) w zespole, nazwanych kompetencjami projektowymi oraz określenie doświadczenia projektowego.

Pomiar doświadczenia projektowego

W ramach badania doświadczenia projektowego członków zespołu zaproponowano zadanie pięciu pytań a następnie przydzielenie odpowiedziom uzyskany od członków zespołu określonej metodą ekspercką liczby punktów. W tabeli zestawiono proponowane wartości punktowe przydzielane w zależności od odpowiedzi wraz z uzasadnieniem.

Tab. 7.1 Skala do pomiaru doświadczenia zespołu

PYTANIE	ODPOWIEDŹ	PUNKTY ZA ODPOWIEDŹ	UZASADNIENIE
Ile lat zajmuje się Pan(i) projektami?	0–3 lata 4–6 lat Więcej niż 6 lat	0 punktów 1 punkt 2 punkty	Przyjmuje się, że osoba, która działa w obszarze projektów więcej niż 6 lat, jest osobą z dużym doświadczeniem projektowym.
W ilu projektach uczestniczył(a) Pan(i) w swojej karierze zawodowej?	0–5 projektów 6–20 Powyżej 20	0 punktów 1 punkt 2 punkty	Uznano, że realizacja przynajmniej 20 projektów pozwala nabyć wystarczająco dużo doświadczenia projektowego.
Jaki procent tej liczby projektów stanowiły projekty informatyczne?	0–30% 30%–60% Powyżej 60%	0 punktów 1 punkt 2 punkty	Jeżeli projekty informatyczne stanowiły większość projektów, w których uczestniczyła dana osoba, wtedy uznaje się ją za doświadczoną.
W ilu projektach występował(a) Pan(i) w roli kierowniczej?	0–4 projekty 5–10 projektów Powyżej 10	0 punktów 1 punkt 2 punkty	Zakłada się, że optymalną liczbą do uznania kierownika za doświadczonego jest 10 projektów.
Jaki procent z nich uważa Pan(i) za udane?	80–100% 50%–70% Mniej niż 50%	0 punktów 1 punkt 2 punkty	Przyjęto, że doświadczony kierownik projektu raczej nie uważa zbyt dużej liczby projektów za udane, raczej stawia na doskonalenie i uczenie się na błędach.

Źródło: opracowanie własne

Kryteria oceniania przyjęto metodą ekspercką i stworzono je na potrzeby pomiaru dojrzałości zespołu w celu późniejszego umiejscowienia tej zmiennej w adaptacyjnym agentowym modelu zarządzania projektami informatycznymi. Ważne jest jednak, aby oprócz pomiarów w takim modelu zaimplementować możliwość interpretacji przeprowadzonych wyników. Zaproponowano w celach testowych następujące interpretacje:

W zależności od sumy punktów uzyskanych przez członka zespołu uznano, że jego poziom doświadczenia jest:

- MAŁY — jeżeli suma w części dotyczącej doświadczenia wynosi od 0 do 3,

- ŚREDNI — jeżeli suma z części dotyczącej doświadczenia wynosi od 4 do 7,
- WYSOKI — jeżeli suma wynosi od 8 do 10.

Po określeniu poziomu doświadczenia pojedynczego członka należy sprawdzić poziom doświadczenia całego zespołu. Przyjęto następujące reguły dla określenia poziomu doświadczenia całego zespołu:

- MAŁY — jeśli większość członków zespołów uzyskała jednostkowo poziom MAŁY,
- ŚREDNI — jeśli przynajmniej połowa członków uzyskała poziom ŚREDNI,
- WYSOKI — jeżeli bezwzględna większość (2/3) zespołu spełnia poziom WYSOKI.

Nie zaleca się dokonywania uśrednień (np. że gdy dla jednego członka zespołu wskazano poziom mały, a dla drugiego poziom wysoki, to poziom dojrzałości zespołu jest średni). W przypadku zespołu obowiązują zasady łańcucha (łańcuch jest tak mocny jak najsłabsze ogniwo), czyli jeżeli dominują członkowie o poziomie małym, poziom doświadczenia całego zespołu należy kwalifikować jako mały, jeżeli dominują osoby z poziomem ŚREDNI, to poziom doświadczenia zespołu jest średni.

Pomiar kompetencji zespołowych

Poziom kompetencji do pracy w zespole postanowiono mierzyć za pomocą wybranych pytań z testu Belblina. W zależności od tego, jak bardzo członek zespołu zgadza się z którymś ze zdań, przyznaje mu od 0 do 10 punktów, przy czym jeżeli wybiera więcej niż jedną odpowiedź, to rozdziela 10 punktów według własnego uznania.

Pytanie 1. Oto, co mogę wnieść w pracę zespołu:

- Mam umiejętność szybkiego dostrzegania i wykorzystywania nadarzających się okazji.
- Potrafię pracować z bardzo różnymi ludźmi.
- Bardzo łatwo przychodzi mi wymyślanie nowych rozwiązań.
- Potrafię umiejętnie zachęcać ludzi do aktywnego udziału, kiedy widzę, że mogę wnieść coś wartościowego do pracy zespołu.
- Potrafię dopilnować realizacji zadania.
- Potrafię znieść chwilową krytykę, jeżeli ostatecznie prowadzi to do osiągnięcia celów.
- Jestem w stanie błyskawicznie ocenić, jaka metoda będzie właściwa w sytuacji, z którą zetknąłem się w przeszłości.
- Potrafię podać rozsądne i obiektywne uzasadnienie różnych kierunków działania.

Pytanie 2. Charakterystyczną cechą mojej pracy w zespole jest to, że:

- Jestem zainteresowany bliższym poznaniem kolegów.
- Nie waham się krytykować opinii innych i wyrażać sądów niepodzielanych przez większość.
- Zwykle potrafię znaleźć argument przeciwko nierozsądnym propozycjom.
- Wydaje mi się, że mam umiejętność prowadzenia pracy w grupie zgodnie z przyjętym planem.
- Często odrzucam to, co oczywiste i proponuję zaskakujące rozwiązanie.
- Wprowadzam perfekcjonizm do każdej pracy zespołowej.
- Potrafię wykorzystać kontakty poza grupą.
- Chociaż pragnę poznać wszystkie opinie, w momencie podejmowania decyzji polegam głównie na własnym zdaniu.

Pytanie 3. Problemy, jakie mogę mieć w zespole, to:

- Okazują zniecierpliwienie w stosunku do tych, którzy opóźniają pracę.
- Można mi zarzucić, że myślę w sposób zbyt analityczny i rzadko kieruję się intuicją.
- Moje pragnienie, aby zadanie zostało wykonane jak najlepiej, może zwalniać tempo pracy.
- Szybko się nudzę i liczę na to, że bardziej dynamiczni członkowie grupy pobudzą mnie do działania.
- Trudno mi rozpocząć pracę, dopóki cele nie są całkowicie jasne.
- Czasami mam trudności z wyjaśnianiem skomplikowanych treści.
- Zdaję sobie sprawę, że wymagam od innych tego, czego sam nie potrafię zrobić.
- Waham się przedstawić swoje poglądy w sytuacji, gdy inni mają odmienne zdanie.

Następnie wartość liczbową przypisaną przez odpowiedniego członka zespołu do zadanego pytania (w wierszu) należy umieścić w odpowiednim polu poniższej tabeli.

Tab. 7.2 Tabela odpowiedzi

PYTANIE	KO	RE	W	IN	PZ	KW	DZ	SW
1	d	g	f	c	a	h	b	e
2	h	d	b	e	g	c	a	f
3	g	e	a	f	d	b	h	c
SUMA								

Źródło: opracowanie własne na podstawie [7]

Na końcu należy zsumować ogólną wartość w ramach każdej z kolumn. Pozwoli to na zmierzenie, która pozycja w zespole (oznaczone są one dwuliterowymi skrótami w wierszu nagłówkowym) jest dominująca dla danego członka zespołu. Ta pozycja, dla której wynik będzie największy, określa jego preferowaną pozycję w zespole.

Poniżej zestawiono objaśnienia poszczególnych skrótów przyjętych w tabeli z wynikami, pokazując jednocześnie określone pozycje w zespole i preferowane zadania.

Tab. 7.3 Tabela objaśniająca skróty z tabeli wynikowej

LP.	SYMBOL	OBJAŚNIENIE	OPIS
1	KO	Koordinator	Dominujący ekstrawertyk, naturalnie kieruje ludźmi, posiada wysokie umiejętności komunikowania się, orientowany na osiągnięcie celu. Korzysta z potencjału zespołu. Może rywalizować.
2	RE	Realizator	Z teorii w czyn. Człowiek czynu, pragmatyk i praktyk. Zrównoważony emocjonalnie. Woli pracować w stabilnych strukturach zadaniowych, wtedy efektywny i systematyczny. Systematyczność minimalizuje innowacyjność.
3	W	Wódz (lokomotywa, prowadzący)	Ekstrawertyk o niespożytych dynamizmie i energii działania. Impulsywny, śmiały, lubiący wyzwania, wysoka potrzeba osiągnięć. Inspiruje działania pracowników, wyznacza kierunek działania. Dominujący, emocjonalny.
4	IN	Innowator (myśliciel)	Inteligencja i twórczy umysł. Innowacyjne pomysły, śmiałe poczynania, intelektualna zabawa. Niezależny indywidualista. Wyzwaniem są nowe rozwiązania. Jeśli niedoceniany — czuje się dotknięty.
5	PZ	Poszukiwacz źródeł	Ekstrawertyk o wysokich umiejętnościach interpersonalnych. Łatwo nawiązuje znajomości, otwarty, serdeczny. Znajduje rozwiązania poza grupą, kompilując informacje z innych źródeł. Racjonalnie myślący, dobrze rozumie się z myślicielem. Jest niezbędny w sytuacjach ryzykownych, gdy mamy za mało informacji.
6	KW	Krytyk wartościujący	Wysoka inteligencja. Poważny, spokojny, zrównoważony. Wnikliwie, krytycznie i obiektywnie analizuje różne pomysły. Logicznie i trafnie wskazuje słabe strony projektów. Może nie być lubiany przez zespół.
7	DZ	Dusza zespołu	Otwarty, towarzyski, o wysokich umiejętnościach interpersonalnych. Bezproblemowo nawiązuje kontakty z innymi, bezkonfliktowy, pozbawiony potrzeby dominacji. Najważniejszy jest cel zespołu, dlatego łagodzi spory. Buduje atmosferę zespołu. Można odnieść wrażenie że mało pracuje.
8	SW	Skrupulatny wykonawca	Introwertyk, skryty, skromny, sumienny. Pracowicie dba o najdrobniejsze szczegóły zadania, precyzyjny w wykonaniu do końca każdego projektu. „Zapina na ostatni guzik”. Podnosi efektywność pracy zespołu, gdy wszyscy są już zmęczeni. Zamartwia się, nadmiernie przywiązuje uwagę do szczegółów.

Źródło: [7]

Ważne jest jednak, aby przełożyć odpowiedzi w ramach testu kompetencji Belblina na określenie kompetencji zespołowych w sposób mierzalny. Do tego celu uznano, aby odpowiedzi członka zespołu mierzące poziom jego kompetencji rozpatrywać na trójstopniowej skali w zależności od tego, jakie wyniki uzyskano w teście Belblina. Stąd przyjmuje się, że poziom kompetencji zespołowych członka zespołu jest:

- MAŁY — jeżeli brakuje wyraźnie zaznaczonej preferencji co do pozycji w zespole (wartości w kilku kolumnach tabeli wyników są zbliżone),
- ŚREDNI — jeżeli uwidocznione są dwie zasadnicze pozycje w zespole,
- DUŻY — jeżeli występuje jedna, dominująca pozycja w zespole.

W celu określenia poziomu dojrzałości zespołu zaproponowano 5 reguł, które rozstrzygają na jakim poziomie znajduje się zespół. Reguły wraz z wnioskiem określającym poziom dojrzałości zespołu zestawiono w tabeli

Tab. 7.4 Ogólny wynik pomiaru dojrzałości zespołu

REGUŁA	POZIOM DOJRZAŁOŚCI ZESPOŁU
Jeżeli poziom doświadczenia mały i poziom kompetencji mały	Poziom 1 — WYKONYWALNY
Jeżeli poziom doświadczenia mały i poziom kompetencji średni	Poziom 2 — ZARZĄDZALNY
Jeżeli poziom doświadczenia średni i poziom kompetencji średni	Poziom 3 — DEFINIOWALNY
Jeżeli poziom doświadczenia średni i poziom kompetencji wysoki	Poziom 4 — ILOŚCIOWO ZARZĄDZALNY
Jeżeli poziom doświadczenia wysoki i poziom kompetencji wysoki	Poziom 5 — OPTYMALIZACJI

Źródło: opracowanie własne

W celu ułatwienia analiz dokonywanych przez kierowników projektów, zaproponowano również kwestionariusze do pomiaru poziomu dojrzałości oraz do tworzenia zestawień zbiorczych. Dokumenty stanowią również szablony do implementacji w narzędziu wspomagającym zarządzanie projektami informatycznymi w celu zmierzenia poziomu dojrzałości zespołu dostawcy.



Ankieta dla członków zespołu

LIDER:

Uprzejmie prosimy o wypełnienie kwestionariusza badające kompetencje członków zespołów projektowych. Badanie zostanie wykorzystane do celów rozprawy doktorskiej dotyczącej systemu wsparcia zarządzania zespołami projektowymi.

Bardzo dziękujemy za wypełnienie ankiety

Ile lat zajmuje się Pan(i) projektami?	
W ilu projektach uczestniczył(a) Pan(i) w swojej karierze zawodowej?	
Jaki procent tej liczby projektów stanowiły projekty informatyczne?	
W ilu projektach występował(a) Pan(i) w roli kierowniczej?	
Jaki procent z nich uważa Pan(i) za udane?	

W każdej sekcji dysponuje Pan(i) 10 punktami. Należy rozdzielić je pomiędzy zdania, które najlepiej opisują Pana(i) zachowanie. Możliwe jest przydzielenie punktów wszystkim zdaniom, kilku wybranym jak również tylko jednemu. Przydzielone punkty proszę wpisywać w zacieniowanych polach

Oto co mogę wnieść w pracę zespołu:

Mam umiejętność szybkiego dostrzegania i wykorzystywania nadarzających się okazji.	
Potrafię pracować z bardzo różnymi ludźmi.	
Bardzo łatwo przychodzi mi wymyślanie nowych rozwiązań.	
Potrafię umiejętnie zachęcać ludzi do aktywnego udziału kiedy widzę, że mogę wnieść coś wartościowego do pracy zespołu.	
Potrafię dopilnować realizacji zadania.	
Potrafię znieść chwilową krytykę, jeżeli ostatecznie prowadzi to do osiągnięcia celów.	
Jestem w stanie błyskawicznie ocenić, jaka metoda będzie właściwa w sytuacji, z którą zetknąłem się w przeszłości.	
Potrafię podać rozsądne i obiektywne uzasadnienie różnych kierunków działania.	
Suma:	10

Charakterystyczną cechą mojej pracy w zespole jest to, że:

Jestem zainteresowany bliższym poznaniem kolegów.	
Nie waham się krytykować opinii innych i wyrażać sądów niepodzielanych przez większość.	
Zwykle potrafię znaleźć argument przeciwko nierozsądnym propozycjom.	
Wydaje mi się, że mam umiejętność prowadzenia pracy w grupie zgodnie z przyjętym planem.	
Często odrzucam to, co oczywiste i proponuję zaskakujące rozwiązanie.	
Wprowadzam perfekcjonizm do każdej pracy zespołowej.	
Potrafię wykorzystać kontakty poza grupą.	
Chociaż pragnę poznać wszystkie opinie, w momencie podejmowania decyzji polegam głównie na własnym zdaniu.	
Suma:	10

Problemy, jakie mogę mieć w zespole, to:

Okazuję zniecierpliwienie w stosunku do tych, którzy opóźniają pracę.	
Można mi zarzucić, że myślę w sposób zbyt analityczny i rzadko kieruję się intuicją.	
Moje pragnienie, aby zadanie zostało wykonane jak najlepiej, może zwalniać tempo pracy.	
Szybko się nudzę i liczę na to, że bardziej dynamiczni członkowie grupy pobudzą mnie do działania.	
Trudno mi rozpocząć pracę, dopóki cele nie są całkowicie jasne.	
Czasami mam trudności z wyjaśnianiem skomplikowanych treści.	
Zdaję sobie sprawę, że wymagam od innych tego, czego sam nie potrafię zrobić.	
Waham się przedstawić swoje poglądy w sytuacji, gdy inni mają odmienne zdanie.	
Suma:	10



ARKUSZ INDYWIDUALNEJ OCENY DOJRZAŁOŚCI CZŁONKA ZESPOŁU

ID:

ANALIZA DOŚWIADCZENIA

PYTANIE	SKALA	ODPOWIEDŹ	PRYZYMANE PUNKTY
Ile lat zajmuje się Pan(i) projektami?	0–3 — 0 punktów 3–5 — 1 punkt Pow. 6 — 2 punkty		
W ilu projektach uczestniczył(a) Pan(i) w swojej karierze zawodowej?	0–5 — 0 punktów 6–20 — 1 punkt Pow. 20 — 2 punkty		
Jaki procent tej liczby projektów stanowiły projekty informatyczne?	0–30 — 0 punktów 30–60 — 1 punkt Pow. 60 — 2 punkty		
W ilu projektach występował(a) Pan(i) w roli kierowniczej?	0–4 — 0 punktów 5–10 — 1 punkt Pow. 10 — 2 punkty		
Jaki procent z nich uważa Pan(i) za udane?	100–80 — 0 punktów 79–50 — 1 punkt 49–0 — 2 punkty		
SUMA			

MAŁY — jeżeli suma z części dotyczącej doświadczenia wynosi od 0 do 3 punktów.

ŚREDNI — jeżeli suma z części dotyczącej doświadczenia wynosi od 4 do 7 punktów.

WYSOKI — jeżeli suma wynosi od 8 do 10 punktów.

Wpisz uzyskane punkty do poniższej tabeli. Suma punktów w poszczególnych kolumnach wskaże preferowaną pozycję w zespole.

ANALIZA KOMPETENCJI ZESPOŁOWYCH

PYTANIE	KO	RE	W	IN	PZ	KW	DZ	SW
1	d	g	f	c	a	h	b	e
2	h	d	b	e	g	c	a	f
3	g	e	a	f	d	b	h	c
SUMA								

DUŻY — jeżeli występuje jedna preferowana pozycja zespołowa dominująca nad innymi.

ŚREDNI — jeżeli zaznaczają się nie więcej jak dwie dominujące cechy.

NISKI — jeżeli brakuje wyraźnie zaznaczonej preferencji co do pozycji w zespole.

OCENA DOJRZAŁOŚCI

DOŚWIADCZENIE	KOMPETENCJE ZESPOŁOWE



ARKUSZ DOJRZAŁOŚCI ZESPOŁU

ZESPÓŁ:

Zestawienie zbiorcze dla członków zespołu

CZŁONEK ZESPOŁU	DOŚWIADCZENIE	KOMPETENCJE ZESPOŁOWE
#1		
#2		
#3		
#4		
#5		
#6		
#7		
POZIOM ZESPOŁU*		

*** proponowana skala:**

MAŁY — jeśli większość członków zespołów spełnia poziom mały.

ŚREDNI — jeśli przynajmniej połowa członków spełnia poziom średni.

WYSOKI — jeżeli bezwzględna większość (2/3) zespołu spełnia poziom wysoki.

Reguły określające poziom dojrzałości zespołu:

Jeżeli poziom doświadczenia mały i poziom kompetencji mały — poziom 1 (

Jeżeli poziom doświadczenia mały i poziom kompetencji średni — poziom 2 (zarządzalny)

Jeżeli poziom doświadczenia średni i poziom kompetencji średni — poziom 3 (definiowalny)

Jeżeli poziom doświadczenia średni i poziom kompetencji wysoki — poziom 4 (ilościowo zarządzalny)

Jeżeli poziom doświadczenia wysoki i poziom kompetencji wysoki — poziom 5 (optymalizacji)

OGÓLNY POZIOM DOJRZAŁOŚCI ZESPOŁU:	
------------------------------------	--

ZAŁĄCZNIK 2**Propozycja pomiaru DOJRZAŁOŚCI KLIENTA**

W celu pomiaru poziomu dojrzałości klienta zaproponowano zadanie klientowi serii pytań zamkniętych. Na podstawie odpowiedzi na pytania związane z odpowiedniością oraz dopasowaniem klienta określa się poziom dojrzałości klienta. Przyjęto pytania zamknięte, w celu uzyskania jednoznacznych odpowiedzi w sposób możliwie szybki. Uznano, że zadając pytania zamknięte, uda się dokładnie określić poziom dojrzałości klienta (wymuszając niejako precyzyjne odpowiedzi). Po za tym klient może nie mieć czasu na udzielanie odpowiedzi na pytania otwarte stąd też taki typ pytań do pomiaru dojrzałości klienta. Przyjęto następujące pytania:

Pytania dotyczące ODPOWIEDNIOŚCI KLIENTA

1. Czy zna Pan(i) środowiska wytwarzania oprogramowania? (.NET, Eclipse)
2. Czy stosuje Pan(i) w swojej firmie infrastrukturę IT jak serwery, bazy danych?
3. Czy uczestniczył(a) Pan(i) do tej pory w jakimś projekcie?
4. Czy liczba projektów informatycznych w których Pan(i) uczestniczył(a) jest większa niż 10?
5. Czy zna Pan(i) zasadę iteracyjnego wytwarzania?
6. Czy zna Pan(i) jakąś formalną metodę zarządzania projektami informatycznymi?
7. Czy zna Pan(i) zasady podejścia procesowego w zarządzaniu?
8. Czy zna Pan(i) języki modelowania systemów informatycznych UML?
9. Czy zna Pan(i) język modelowania procesów BPMN?
10. Czy umie Pan(i) zamodelować samodzielnie funkcjonalności systemu?
11. Czy uważa Pan(i) za słuszne modyfikowanie wymagań po uruchomieniu projektu?
12. Czy zdaje Pan(i) sobie sprawę z konsekwencji zmiany wymagań w trakcie trwania projektu?

Pytania dotyczące DOPASOWANIA KLIENTA

1. Czy preferuje Pan(i) pracę w grupie ponad indywidualną realizację zadań?
2. Czy pełnił(a) Pan(i) kiedyś rolę lidera w jakiejś grupie?
3. Czy przynależał(a) Pan(i) do jakiegoś zespołu, niekoniecznie projektowego? (muzyczny, sportowy)
4. Czy chętnie pomaga Pan(i) kolegom w pracy?
5. Czy nie traci Pan(i) cierpliwości podczas konieczności tłumaczenia różnych zagadnień innym ludziom?
6. Czy jest Pan(i) skłonny(a) do udzielania szczegółowych odpowiedzi?
7. Czy chętnie korzysta Pan(i) z wiedzy innych osób?
8. Czy dopuszcza Pan(i) możliwość uczestnictwa w spotkaniach zespołu poza ramami pracy? (sytuacje specjalne)
9. Czy lubi Pan(i) angażować się w nowe wyzwania?
10. Czy korzysta Pan(i) z elektronicznych form komunikacji?
11. Czy poszedłby(poszłaby) Pan(i) na integracyjne spotkanie poza miejscem pracy?
12. Czy Pana(i) zdaniem w pracy zespołowej najważniejsza jest dobra współpraca?

Sposób przeliczania wyników:

Obliczenie wyniku końcowego pozwala na pozycjonowanie klienta w jednym z pól macierzy dojrzałości klienta. Kwestionariusz pomiarowy skonstruowano w taki sposób, aby za każdą odpowiedź „TAK” na zadane pytanie przyznać 1 punkt. W zależności od sumy punktów uzyskanych z każdej sekcji można klienta umiejscowić w jednym z 4 możliwych pól określających poziom dojrzałości.

Tab. 7.5 Możliwe poziomy dojrzałości klienta

SEKCJA	WYNIK W SEKCJI	POZYCJA
ODPOWIEDNIOŚĆ	Od 0 do 8	nieodpowiedni
	Od 9 do 12	odpowiedni
DOPASOWANIE	Od 0 do 6	niedopasowany
	Od 7 do 12	dopasowany

Źródło: opracowanie własne

W interpretacji wyników przyjęto kryteria ostre, traktując że odpowiedni klient to taki, który odpowiada twierdząco na zdecydowaną większość (2/3) pytań (więcej niż 8). Natomiast dopasowany klient to taki, który przynajmniej w połowie odpowiada oczekiwany cechom związanym ze współpracą w zespole. Takie nierównomierne rozłożenie punktów uzyskanych przez klienta wynika z tego, że dopasowanie klienta wiąże się z kompetencjami, które klient może nabywać w trakcie trwania projektu (uczyć się współpracy w grupie), więc za dopasowanego klienta uznaje się takiego, który na przynajmniej **4 pytania** odpowiada twierdząco. W przypadku kompetencji informatycznych związanych z realizacją projektów elastyczność jest już dużo mniejsza.

W celu pomiaru dojrzałości klienta arkusz kwestionariusz dojrzałości klienta.



ARKUSZ DOJRZAŁOŚCI KLIENTA

KLIENT:

Proszę postawić znak „X” w polu odpowiedzi TAK lub NIE, w zależności od tego czy zgadza się Pan(i) z postawionym pytaniem

ODPOWIEDNIOŚĆ

#	PYTANIE	TAK	NIE
1	Czy zna Pan(i) środowiska wytwarzania oprogramowania? (.NET, Eclipse)		
2	Czy stosuje Pan(i) w swojej firmie infrastrukturę IT jak serwery, bazy danych?		
3	Czy uczestniczył(a) Pan(i) do tej pory w jakimś projekcie?		
4	Czy liczba projektów informatycznych w których Pan(i) uczestniczył(a) jest większa niż 10?		
5	Czy zna Pan(i) zasadę iteracyjnego wytwarzania?		
6	Czy zna Pan(i) jakąś formalną metodę zarządzania projektami informatycznymi?		
7	Czy zna Pan(i) zasady podejścia procesowego w zarządzaniu?		
8	Czy zna Pan(i) języki modelowania systemów informatycznych UML?		
9	Czy zna Pan(i) język modelowania procesów BPMN?		
10	Czy umie Pan(i) zamodelować samodzielnie funkcjonalności systemu?		
11	Czy uważa Pan(i) za słuszne modyfikowanie wymagań po uruchomieniu projektu?		
12	Czy zdaje Pan(i) sobie sprawę z konsekwencji zmiany wymagań w trakcie trwania projektu?		
SUMA			

DOPASOWANIE

#	PYTANIE	TAK	NIE
1	Czy preferuje Pan(i) pracę w grupie ponad indywidualną realizację zadań?		
2	Czy pełnił(a) Pan(i) kiedyś rolę lidera w jakiejś grupie?		
3	Czy przynależał(a) Pan(i) do jakiegoś zespołu, niekoniecznie projektowego? (muzyczny, sportowy)		
4	Czy chętnie pomaga Pan(i) kolegom w pracy?		
5	Czy nie traci Pan(i) cierpliwości podczas konieczności tłumaczenia różnych zagadnień innym ludziom?		
6	Czy jest Pan(i) skłonny(a) do udzielania szczegółowych odpowiedzi?		
7	Czy chętnie korzysta Pan(i) z wiedzy innych osób?		
8	Czy dopuszcza Pan(i) możliwość uczestnictwa w spotkaniach zespołu poza ramami pracy? (sytuacje specjalne)		
9	Czy lubi Pan(i) angażować się w nowe wyzwania?		
10	Czy korzysta Pan(i) z elektronicznych form komunikacji?		
11	Czy poszedłby(poszłaby) Pan(i) na integracyjne spotkanie poza miejscem pracy?		
12	Czy Pana(i) zdaniem w pracy zespołowej najważniejsza jest dobra współpraca?		
SUMA			



ARKUSZ INTERPRETACJI WYNIKÓW POMIARU DOJRZAŁOŚCI KLIENTA

Wypełnia KIEROWNIK

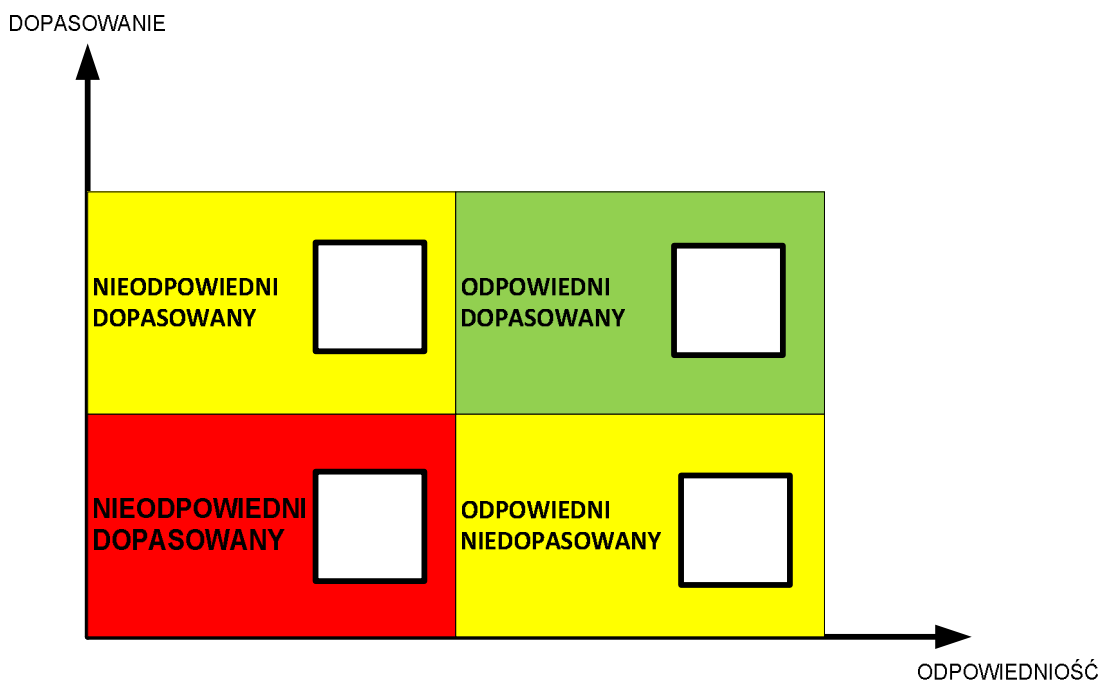
KLIENT:	
----------------	--

WYNIK W SEKCJI ODPOWIEDNIOŚĆ	
WYNIK W SEKCJI DOPASOWANIE	

Interpretacja wyników

SEKCJA	WYNIK W SEKCJI	POZYCJA
ODPOWIEDNIOŚĆ	Od 0 do 8	nieodpowiedni
	Od 9 do 12	odpowiedni
DOPASOWANIE	Od 0 do 6	niedopasowany
	Od 7 do 12	dopasowany

POZIOM ODPOWIEDNIOŚCI	POZIOM DOPASOWANIA



ZAŁĄCZNIK 3

Propozycja pomiaru ENTROPII PROJEKTU

Kwestionariusz pomiaru entropii projektowej oparto na 12 pytaniach do kierownika projektu. W odpowiedzi na każde z pytań kierownik podaje wartość liczbową z przedziału od 1 do 3, zgodnie z własnym odczuciem, przy czym 1 oznacza, że kierownik w pełni zgadza się z podanym stwierdzeniem, 2 — częściowo się zgadza, 3 — zupełnie się nie zgadza.

12 pytań do kierownika projektu

1. Wizja projektu została określona precyzyjnie.
2. Cele projektu są znane.
3. Cele projektu są mierzalne.
4. Klient sformułował wymagania dla całego projektu.
5. Klient potrafi wyraźnie formułować wymagania dla dowolnego etapu.
6. Zakres projektu jest znany w całości.
7. Przepływy komunikacyjne w projekcie są precyzyjne.
8. Ustalono hierarchie ważności dla komunikatów.
9. Zadania w projekcie są proste.
10. Szacowanie pracochłonności zadań jest stabilne i przewidywalne.
11. Oszacowano wszystkie rodzaje ryzyka (technologiczne, społeczne, biznesowe).
12. Obieg informacji w projekcie jest kontrolowany przez kierownika.

Sposób przeliczania wyników:

Na podstawie odpowiedzi kierowników sumuje się wartości (1, 2 lub 3) wprowadzone do kwestionariusza. Tym samym uzyskuje się sumaryczne wartości pozwalające zaliczyć projekt do jednego z trzech przedziałów zestawionych w tabeli Tab. 2.6.

Tab. 7.6 Tabela do interpretacji wyników

POZIOM ENTROPII	Liczba punktów
Projekt o MAŁEJ entropii	12–20
Projekt o ŚREDNIEJ entropii	21–26
Projekt o DUŻEJ entropii	27–36

Źródło: opracowanie własne

Przyjęty arkusz pomiaru entropii jest instrumentem prototypowym. Przyjęto wartości ostre, jednak rozważa się możliwość rozbudowy instrumenty o mechanizmy logiki rozmytej.



ARKUSZ OCENY ENTROPII PROJEKTU

KIEROWNIK:

PROJEKT:

Wstaw w polu ODPOWIEDŹ jedną z trzech wartości, w zależności do tego czy zgadzasz się ze sformułowaniem w polu PYTANIE

- 1- zupełnie się nie zgadzasz
- 2- częściowo się zgadzasz
- 3- zupełnie się nie zgadzasz

#	PYTANIE	ODPOWIEDŹ
1	Wizja projektu została określona precyzyjnie.	
2	Cele projektu są znane.	
3	Cele projektu są mierzalne.	
4	Klient sformułował wymagania dla całego projektu.	
5	Klient potrafi wyraźnie formułować wymagania dla dowolnego etapu.	
6	Zakres projektu jest znany w całości.	
7	Przepływy komunikacyjne w projekcie są precyzyjne.	
8	Ustalono hierarchie ważności dla komunikatów.	
9	Zadania w projekcie są proste.	
10	Szacowanie pracochłonności zadań jest stabilne i przewidywalne.	
11	Oszacowano wszystkie rodzaje ryzyka (technologiczne, społeczne, biznesowe).	
12	Obieg informacji w projekcie jest kontrolowany przez kierownika	
SUMA		

Interpretacja wyników:

POZIOM ENTROPII	SUMA punktów
Projekt o MAŁEJ entropii	12–20
Projekt o ŚREDNIEJ entropii	21–26
Projekt o DUŻEJ entropii	27–36

POZIOM ENTROPII PROJEKTU	
--------------------------	--

ZAŁĄCZNIK 4

Arkusz obserwacji koordynatora eksperymentu

RUNDA

TEMAT PROJEKTU	LIDER	UWAGI
1. Budowa centrów kompetencyjnych firm informatycznych dla wsparcia procesów dydaktyki i biznesu (uczelnie i biznes),	AGNIESZKA JANCZULEWICZ	
2. Wspólne przedsięwzięcie IBM i PG - "wsparcie cyklu wytwarzania oprogramowania poprzez integrację narzędzi informatycznych IBM"	PAWEŁ MADEJ (PG)	
3. Rozbudowa Centrów Kompetencyjnych w oparciu o program Smarter Planet i zaproszenie do współpracy i realizacji wspólnego projektu partnerów z Litwy i Obwodu Kaliningradzkiego	ŁUKASZ SZCZYGIELSKI	
4. Biznesowe wsparcie Centrum Kompetencyjnego PG dla wdrażania środowiska RTC dla potrzeb partnerów zewnętrznych.	Tomasz Górski (RS)	
5. Oprogramowanie Service Manager — wsparcie procesów budowy organizacji informatycznych architekturą SOA/korporacyjną/ITIL	ADAM CZARNECKI	
6. Budowa Data Center —dobór i monitorowanie zapotrzebowania na zasoby „chmury”; analiza kosztów outsourcingu przetwarzania danych.	MAREK OŻAROWSKI (PG)	
7. Prognozowanie stanu (dojrzałości) organizacji informatycznej IT w oparciu o systemy oparte na wiedzy	TOMASZ SITEK (PG)	
8. MS Dynamics AX 2009 — centrum kompetencyjne Microsoft i rozwój dydaktyki; wykorzystanie usług systemu dla tworzenia rozwiązań dla sektora MSP	JANUSZ CZUCHNOWSKI (PG)	
9. Semantyczny portal wiedzy środowiskiem wsparcia procesów integracji uczelni i biznesu	IRENA BACH (PK)	