ZAŁĄCZNIK 11
Standardy analizy i projektowania

SPIS TREŚCI

1. WSTĘP ... 3
1.1 BAZA ... 3
1.2 MDA ... 3
1.3 PROCES WYTWÓRCZY ... 3
1.4 UML ... 4
1.5 MODELE ... 4
1.6 ZARZĄDZANIE WYMAGANIAMI ... 4
1.6.1 PODSTAWY ... 4
1.6.2 ANALIZA WYMAGAŃ ... 5
1.6.3 SPECYFIKACJA WYMAGAŃ ... 5
1.6.4 ŚLADOWANIE MODELU WYMAGAŃ ... 5
1.7 ANALIZA BIZNESOWA ... 6
1.8 ANALIZA SYSTEMOWA ... 6
1.8.1 MODELE/DIAGRAMY ... 6
1.8.2 MODEL PRZYPADKÓW UŻYCIA ... 7
1.8.3 DIAGRAMY AKTYWNOŚCI ... 9
1.8.4 MODEL DZIEDZINY SYSTEMU ... 11
1.8.5 PROTOTYP FUNKCJONALNY ... 12
1.9 PROJEKT ... 12
1.9.1 RODZAJE MODELI ... 12
1.9.2 ZASTOSOWANIE MDA ... 12
1.9.3 TRANSFORMACJA ... 12
1.9.4 ŚLADOWANIE TRANSFORMACJI ... 13
1.10 STANDARDY WYKONAWCZE ... 13
2. WYMIAROWANIE MODYFIKACJI METODĄ COSMIC ... 14

2.1 WYMIAROWANIE COSMIC 14
2.1.1 INTEGRACJA WYMIAROWANIA Z ANALIZĄ SYSTEMOWĄ 14
2.1.2 INTEGRACJA WYMIAROWANIA Z ZARZĄDZANIEM 14
2.2 SZACOWANIE WSTĘPNE 16
2.2.1 ISTNIEJĄCE I ZMIENIANE PRZYPADKI UŻYCIA 16
2.2.2 NOWE PRZYPADKI UŻYCIA 17
2.2.3 USUWANE PRZYPADKI UŻYCIA 17
2.2.4 WYNIKI SZACOWANIA WSTĘPNEGO 18
2.3 WYMIAROWANIE PEŁNE 18
2.3.1 ISTNIEJĄCE I ZMIENIANE PRZYPADKI UŻYCIA 18
2.3.2 NOWE PRZYPADKI UŻYCIA 19
2.3.3 USUWANE PRZYPADKI UŻYCIA 19
2.3.4 STRATEGIA ZASTOSOWANIA METODY COSMIC 19
2.3.5 REGUŁY 19
2.4 WYMIAROWANIE WYMAGAŃ POZAFUNKCJONALNYCH MODYFIKACJI 21

1. WSTĘP

1.1 BAZA

W całym procesie analizy i projektu przywoływane będą następujące standardy i notacje:

- *Model Driven Architecture* (**MDA**)
  http://www.omg.org/mda/specs.htm
- *Rational Unified Process* (**RUP**)
  http://www-01.ibm.com/software/awdtools/rup/
- *Unified Modeling Language* (**UML**) w wersji 2.4
  http://www.omg.org/spec/UML/
- *Business Process Model and Notation* (**BPMN**) w wersji 2.0
  http://www.omg.org/spec/BPMN/
- *Business Analysis Body of Knowledge* (**BABoK**)

1.2 MDA

MDA to idea pozwalająca na odseparowanie "poziomów" modelu (analizy biznesowej, systemowej, projektowego itd.) oraz określeniu sposobu przejść pomiędzy tymi modelami (transformacji).

Model systemu to każdy opis lub specyfikacja systemu oraz jego otoczenia opisująca go z konkretnej perspektywy i w konkretnym celu. Model może być kombinacją diagramów i opisu wykonanego z pomocą języka naturalnego lub wybranej notacji.

MDA przewiduje trzy perspektywy i modele systemu:

- **CIM** *(Computation Independent Model)*– opisuje system w całkowitym oderwaniu od informatyki; jest "przeznaczony" głównie dla osób znających zasady funkcjonowania modelowanej organizacji.
- **PIM** *(Platform Independent Model)* - opisuje system w oderwaniu od metody i narzędzi implementacji; traktowany jest jako rodzaj maszyny wirtualnej, obrazuje logikę jaką będzie realizowało oprogramowanie, które ma powstać (produkt); jest specyfikowany poprzez usługi jakie będzie realizował.
- **PSM** *(Platform Specific Model)* - opisuje sposób implementacji systemu; stanowi model PIM uzupełniony o szczegóły specyficzne dla platformy i technologii, jakie zostaną użyte w celu jego wykonania; opisuje system z perspektywy platformy na jakiej zostanie zaimplementowany

1.3 PROCES WYTWÓRCZY

Zastosowanie wyżej opisanego podejścia pozwala na jasne określenie miejsca poszczególnych dyscyplin w klasycznym procesie produkcji oprogramowania:

- **Analiza biznesowa** - operuje na modelu **CIM**.
- **Analiza systemowa** - operuje na modelu **PIM**.
- **Projektowanie** - operuje na modelu wytworzonym przez analizę systemową (**PIM**), uwzględniając model platformy (**PSM**).

1.4 UML

Notacja UML przewiduje szereg typów modeli i diagramów. Ich minimalny zbiór na potrzeby projektu został opisany w kolejnym rozdziale i powiązany z pozostałymi definicjami metodycznymi.

1.5 MODELE

Opisane w niniejszym rozdziale założenia, wynikające z przyjętych standardów, są ze sobą mocno skorelowane. W poniższej tabeli przedstawiono mapowania poszczególnych pojęć. Poniższe przypisanie nie ogranicza wykorzystania innych środków wyrazu spośród właściwych dla danego dokumentu/modelu.

Etap procesu	Model pojęciowy (niniejszydokument)	MDA	Model UML
Analiza systemowa	Model przypadków użycia	PIM	Przypadków użycia
		PIM	Pakietów
			Obiektów
		PIM	Aktywności
	Model dziedziny systemu (logiczny model danych)	PIM	Klas, Obiektów
			Pakietów
			Maszyn stanowych
Projekt	Model komponentów	PIM/PSM	Komponentów
	Model interakcji	PIM/PSM	Sekwencji lub Komunikacji
	Fizyczny model danych	PSM	Klas
			Pakietów
	Model wdrożenia / Model platformy	PSM	Wdrożenia / rozmieszczenia

1.6 ZARZĄDZANIE WYMAGANIAMI

1.6.1 PODSTAWY

Wymagania dzielą się na:

- **Wymaganie funkcjonalne** – określa, jakie usługi ma oferować system widziany od strony użytkownika, jak ma reagować na określone dane wejściowe oraz jak ma się zachowywać w określonych sytuacjach.
- **Wymaganie niefunkcjonalne** (pozafunkcjonalne) - określa ograniczenia nakładane na funkcjonalność i/lub specyfikuje jakość, którą rozwiązanie musi spełniać.

1.6.2 ANALIZA WYMAGAŃ

W ramach procesu analizy, w związku ze zgłaszanymi potrzebami użytkowników, przeprowadzana jest analiza wymagań, której wynikiem musi być:

- Nadanie wymaganiu unikalnego **identyfikatora**.
- Przypisanie odpowiedniej **krytyczności** do wymagania.
- **Grupowanie** wymagań - wymagania powinny być grupowane w pakiety.
- Umieszczenie wymagania w modelu w odpowiednim **miejscu** względem typu, obszaru merytorycznego itp.
- **Zweryfikowanie** wymagania pod względem poprawności jego opisu oraz pod względem jego poprawności w kontekście grupy wymagań oraz całego modelu. Weryfikacja sprawdza spójność, kompletność, zgodność z innymi wymaganiami w ramach modelu, poprawność, niedwuznaczność, możliwość przetestowania etc.
- **Zwalidowanie** wymagania - sprawdzenie pod względem zgodności z celami biznesowymi, np. poprzez porównanie danego wymagania z wymaganiami nadrzędnymi (np. biznesowymi).

Analiza wymagań jest procesem, w który wpisane są "możliwości nawrotów". Oznacza to, że wykrycie np. wymagań zbyt ogólnych w trakcie analizy (np. niemożliwych w związku z tym do przetestowania), będzie wiązało się z doprecyzowaniem tych wymagań poprzez dodanie wymagań bardziej szczegółowych (możliwych do przetestowania). Może powodować to konieczność przeprowadzenia ponownej analizy wymagania, grupy wymagań albo całej analizowanej dziedziny.

1.6.3 SPECYFIKACJA WYMAGAŃ

Specyfikacja wymagań jest zbiorem (rejestrem) wszystkich zidentyfikowanych wymagań. Specyfikacja musi umożliwiać efektywne przeszukiwanie polegające na określaniu kryteriów według wartości atrybutów wymagań.

Specyfikacja taka musi zawierać następujące atrybuty:

- **Atrybuty wymagania** - przypisane do wymagań w ramach analizy wymagań (opisane w poprzednim rozdziale).
- **Atrybuty procesowe** - związane z procesem wytwórczym:
    - historia przypisania wymagania do formalnych produktów (takich jak etapy, fazy, zmiany itp.),
    - historia statusów związana z cyklem życia wymagania (min. zaproponowane, zatwierdzone).

<u>Modelowanie</u>: Wymagania są definiowane w postaci modelu UML:

- element wymagania: *Requirement,*
- model/diagram wymagań: *Requirements Diagram*.

1.6.4 ŚLADOWANIE MODELU WYMAGAŃ

Model wymagań musi dostarczać informacje o powiązaniach wymagań zarówno pomiędzy sobą, jak również z innymi elementami. W szczególności powiązane muszą być wymagania i realizujące je elementy na poszczególnych poziomach MDA. Jest to jeden z przejawów **śladowalności** modelu, którą należy rozumieć jako możliwość przejścia od dowolnego elementu do elementów, z których on wynika.

Każde wymaganie w modelu, oprócz wymagań najwyższego poziomu, musi realizować wymaganie/wymagania wyższego poziomu - wprost albo poprzez wymagania, których jest częścią.

1.7 ANALIZA BIZNESOWA

Zgodnie z opisanymi w dokumencie głównym założeniami dotyczącymi **MDA**, celem analizy biznesowej jest wytworzenie modelu niezależnego od informatyki (**CIM**).

W przypadku przyjętego modelu procesu wytwórczego zakłada się, że modele procesów biznesowych są podstawą do wytworzenia modelu analizy biznesowej. Strony w ramach spotkań dotyczących produktu, doprecyzowują modele, dokonują próby optymalizacji procesów itd.

Z drugiej strony model ten jest podstawą do wytworzenia modelu dla analizy systemowej (**PIM**). W związku z tym elementy modelu analizy biznesowej są uwzględniane w modelach dla analizy systemowej, w celu zobrazowania zależności (realizacji) - przypadki użycia realizują zadania procesów biznesowych z analizy biznesowej

<u>Modelowanie</u>: Analiza biznesowa jest modelowana w notacjach:

- BPMN (procesy);
- ew. UML (model danych/informacyjny).

1.8 ANALIZA SYSTEMOWA

Zgodnie z opisanym w założeniach do metodyki podejściem MDA, celem analizy systemowej jest wytworzenie modelu opisującego system z punktu widzenia użytkownika, niezależnego od narzędzi implementacji i technologii.

1.8.1 MODELE/DIAGRAMY

W ramach dokumentacji analitycznej powinny powstać co najmniej następujące modele UML:

- Model przypadków użycia:
  - Przypadków użycia
  - Aktywności
- Model dziedziny:
  - Klas, ew. Obiektów
  - Maszyn stanowych

1.8.2 MODEL PRZYPADKÓW UŻYCIA

Definiowanie przypadków użycia ma na celu określenie funkcjonalności systemu z punktu widzenia aktorów.

Modelowanie: model przypadków użycia jest definiowany w notacji UML za pomocą:

- diagramów przypadków użycia;
- diagramów aktywności dla przypadków użycia.

1.8.2.1 PRZYPADKI UŻYCIA

Przypadek użycia to sekwencja akcji realizowanych przez system, w efekcie których dostarczane są obserwowalne rezultaty (dla przypadków na poziomie celu użytkownika, dla konkretnego aktora), gdzie akcja, to atomowe działanie, tj. wykonywane albo w całości albo wcale.

Obserwowalny rezultat to sekwencja akcji dla przypadków użycia na poziomie celu użytkownika, musi zakończyć się wyprodukowaniem czegoś, co posiada użyteczną wartość - nie ma potrzeby wykonywania kilku przypadków użycia, dla uzyskania tej wartości. (Pomocnym kryterium wskazującym, że przypadek użycia na poziomie celu użytkownika nie dostarcza obserwowalnych rezultatów jest to, że nie można opisać celu przypadku w trybie dokonanym.)

Specyfikacja pojedynczego przypadku użycia powinna posiadać:

- identyfikator,
- nazwę,
- opis celu i kontekst,
- wskazywać realizowane **wymagania** (o ile realizuje wymagania),
- wskazywać wspierane zadania **procesów biznesowych** (o ile wspiera zadania procesów biznesowych).

1.8.2.2 ABSTRAKCYJNE PRZYPADKI UŻYCIA

Specyficzną kategorię stanowią **abstrakcyjne przypadki użycia**, które z definicji nie mają wystąpień (instancji) w systemie. Mogą mieć one zastosowanie tam, gdzie:

- istnieje potrzeba opisana konceptu analitycznego dla pewnej hierarchii przypadków konkretnych (istnieje wówczas możliwość wyspecyfikowania bazowego, abstrakcyjnego przypadku użycia, który będzie wspólną generalizacją dla wszystkich takich przypadków konkretnych).
- służą przypadki włączane oraz rozszerzenia dla innych przypadków użycia (aktorzy nie są z nimi bezpośrednio powiązani poprzez asocjacje - takie przypadki można również rozumieć jako przypadki na poziomie podfunkcji, w przeciwieństwie do przypadków poziomu celu użytkownika).

W zależności od potrzeby, przypadek abstrakcyjny może nie spełniać niektórych opisanych poniżej ograniczeń specyfikacji, na przykład: może nie posiadać przebiegu/przebiegów, może nie realizować "istotnego" celu z perspektywy aktorów.

1.8.2.3 PROCESY BIZNESOWE

Model przypadków użycia posiada "styk" z modelem analizy biznesowej.

Przypadki użycia wynikają z zadań w procesie biznesowym, ale również mogą wynikać z funkcjonalności "pozaprocesowych" - takich, które nie są (tylko) bezpośrednio wykonywane jako zadania w procesie lecz mogą być (również) wykonywane w dowolnym momencie (ad hoc). Dany przypadek może być wielokrotnie użyty w ramach tego samego albo różnych procesów biznesowych.

Modelowanie. Jeżeli przypadek użycia realizuje (wspiera) zadania procesu biznesowego, wówczas musi być w modelu powiązany z zadaniem relacją realizacji (*realize*).

1.8.2.4 REALIZOWANE WYMAGANIA

Przypadki użycia na poziomie celu użytkownika bezpośrednio wynikają z **wymagań funkcjonalnych**, które *de facto* realizują. Realizacja polega na tym, że wymagania funkcjonalne albo powodują powstanie nowego przypadku użycia, albo przypadek "jest zmieniany pod wpływem" wymagań funkcjonalnych.

Jedno wymaganie funkcjonalne może tworzyć/zmieniać wiele przypadków użycia, jak również jeden przypadek użycia może być utworzony/zmieniany przez wiele wymagań. Również w ramach cyklu życia pojedynczego przypadku użycia, mógł być on wielokrotnie modyfikowany przez kolejne wymagania funkcjonalne.

Modelowanie: Przypadek użycia powinien być powiązany relacją realizacji (*realize*) ze wszystkimi wymaganiami jakie go "utworzyły" a następnie "zmieniały".

Wymagania niefunkcjonalne mogą ograniczać przypadek użycia. Specyfikacja takich wymagań powinna obejmować listę wszystkich wymagań niefunkcjonalnych, które ograniczają przypadek użycia.

1.8.2.5 DZIEDZINA PRZYPADKU UŻYCIA

W ramach opisu celu i kontekstu przypadku użycia, może zostać określona dziedzina przypadku użycia, czyli specyfikacja sieci obiektów, na których operuje przypadek. Dziedzina ta bazuje na modelu danych (dziedziny) systemu. Poszczególne kluczowe elementy powinny tu zostać nazwane tak, aby w dalszej części opisu można się było do nich odnosić przez ich nazwy.

Opisana dziedzina przypadku użycia jest niezależna od przyjętego sposobu specyfikowania przepływów danych (*data flow*)– zdefiniowane w dalszych rozdziałach.

1.8.2.6 REGUŁY SYSTEMOWE

Szczególną uwagę w kontekście modelu przypadków użycia należy zwrócić na aspekt użycia reguł systemowych. Reguły systemowe w danym przypadku użycia, często mogą mieć szerokie zastosowanie w ramach całego systemu. Na przykład reguły dotyczące walidacji często przetwarzanych danych, mogą pojawiać się w wielu przypadkach użycia, w różnej formie zapisu, ale ekwiwalentne co do znaczenia (operujące na takich samych danych wejściowych i zwracające takie same wyniki).

Reguła systemowa musi być wyrażeniem typowanym, tj. zwracającym podczas wyliczenia wartość określonego typu. Reguła służy więc do wyliczenia wartości - wartość może być wyliczana na podstawie innych dostępnych dla reguły wartości, np.: obiektów wejściowych do przypadku, wartości wcześniej wyliczonych i przechowywanych w kontekście przypadku użycia, danych odczytanych, wprowadzonych z interfejsu użytkownika.

Reguły są używane w różnych kontekstach specyfikacji przypadku użycia jako:

- **warunki brzegowe** - typ reguły musi być wówczas typem logicznym (PRAWDA/FAŁSZ):
    - **warunki początkowe** - przypadek nie może zostać uruchomiony, jeżeli reguła nie jest spełniona;
    - **warunki końcowe** - zakładany stan obiektów dziedziny przypadku użycia po wykonaniu przypadku użycia;
- **reguły przetwarzania** - reguła musi być związana z co najmniej z jedną akcją (krokiem przebiegu/przebiegów) przypadku użycia i określa sposób wyliczania, sposób zmiany stanu przetwarzanych obiektów etc.

Przypadek użycia operuje na określonej sieci obiektów, które są zdefiniowane na modelu dziedziny. Dane (typowane klasami modelu dziedziny), które są "dostępne" dla danej reguły (mogą być przez nią przetwarzane), wynikają wprost z przebiegu przypadku użycia - akcji poprzedzających akcje/punkty decyzyjne, w których reguła została użyta. Oznacza to, że reguła nie może operować na danych, które nie są jawnie dostępne, czyli wcześniej (w przebiegu) odczytane, wprowadzone przez użytkownika, utworzone w przebiegu nowych obiektach, bądź danych dostępne w kontekście wykonania przypadku użycia (listy wartości wynikające z enumeracji, słowniki, parametry systemowe itp.).

1.8.3 DIAGRAMY AKTYWNOŚCI

1.8.3.1 PRZEBIEGI PRZYPADKU UŻYCIA

Przebiegi przypadków użycia powinny być zobrazowane w formie diagramów aktywności, które odpowiadają przepływowi sterowania.

Niezależnie od sposobu specyfikowania, przebiegi <u>nie powinny</u> odnosić się do:

- Konkretnego interfejsu użytkownika - nie wskazywać co, jak i gdzie użytkownik klika, wciska, wpisuje itp. Dotyczy to również wskazywania konkretnych nazw kontrolek z interfejsu.

- Technologii - nie wskazywać jak system realizuje zadania, gdzie co zapisuje, jakie technologie używa itp.

Modelowanie: model/diagramy aktywności w notacji UML.

1.8.3.2 PARTYCJE

Każda akcja musi definiować wykonującego daną akcję - najczęściej są to warstwy:

- użytkownik,
- interfejs użytkownika,
- system.

Modelowanie: element *Partition.*

1.8.3.3 KROKI PRZEBIEGÓW

Każda akcja nie będąca akcją użytkownika musi wiązać się z konkretnym działaniem, na przykład:

- odczytem, zapisem itd. danych związanych z dziedziną przypadku użycia,
- przetwarzaniem tych danych przez system - zgodnie z regułami systemowymi,
- wywołaniem włączanego przypadku użycia (relacja „include” na modelu przypadków użycia).

W przypadku kiedy akcja "wywołuje" włączany przypadek użycia, przypadek ten musi również być włączany na poziomie modelu przypadków użycia. Analogicznie jeżeli na poziomie modelu przypadków użycia dany przypadek użycia włącza inny przypadek użycia, w przebiegu musi istnieć akcja (akcje), która "wywołuje" odpowiednio włączany przypadek użycia.

Na diagramach przebiegów przypadków użycia powinny być również odpowiednio oznaczone punkty rozszerzeń dla ew. rozszerzających przypadków użycia (relacja „extend” na modelu przypadków użycia).

Modelowanie: elementy*Action, Decision* etc.

1.8.3.4 PRZEPŁYW DANYCH

Na diagramach aktywności konieczne jest konsekwentne stosowanie modelowania przepływów obiektów z wykorzystaniem notacji „pinowej”.

Należy używać w tym celu poprawnie:

- obiektów istniejących klasy z modelu dziedziny,
- poprawnych liczności
- typu przepływu (in-out).

Akcja lub kilka akcji na partycji interfejsu użytkownika może odnosić się do klasy interfejsu użytkownika. Klasa interfejsu użytkownika może w odpowiednim miejscu modelu definiować zakres (przy wykorzystaniu klas/atrybutów modelu dziedziny).

Modelowanie: elementy *ActionPin*, obiekty klas, klasy.

Powyższa notacja jest szczególnie ważna dla wyznaczania rozmiaru funkcjonalnego oprogramowania - daje to możliwość deterministycznego, obiektywnego i zautomatyzowanego wymiarowania oprogramowania metodą COSMIC. Szczegółowy opis metody COSMIC znajduje się w części dalszej tego dokumentu.

1.8.3.5 KOMUNIKATY

Systemowe akcje przebiegu mogą skutkować między innymi wygenerowaniem informacji dla użytkownika - komunikatu. W celu usystematyzowania i zestandaryzowania treści prezentowanych komunikatów muszą one być niezależnym elementem modelu i muszą posiadać:

- **identyfikator** - musi być unikalny w skali specyfikacji,
- **opis** - treść prezentowaną użytkownikowi, z ewentualnymi znacznikami dla części dynamicznych (generowanych przez system w zależności od kontekstu przypadku użycia, np. numer dokumentu, identyfikator płatnika).

Podejście takie umożliwia ponowne użycie raz zdefiniowanego komunikatu oraz elastyczne zarządzanie treścią komunikatów.

Komunikaty podzielić można na:

- **nieinteraktywne** ("bierne") - takie, które "pojawiają" się w przebiegu przypadku aby poinformować użytkownika o jakimś fakcie,
- **interaktywne** - takie, które niosą za sobą możliwość wyboru dalszej "ścieżki postępowania" - wiążą się z decyzją użytkownika.

Komunikaty muszą być przypisane do akcji (kroku przebiegu), która komunikat "generuje" (jest źródłem komunikatu).

1.8.4 MODEL DZIEDZINY SYSTEMU

1.8.4.1 MODEL KLAS

Model dziedziny powinien być specyfikowany zgodnie ze szczegółowymi założeniami modelowania zorientowanego obiektowo oraz specyfikacji notacji UML.

1.8.4.2 MODEL STANÓW

W przypadku obiektów, dla których istnieje potrzeba zamodelowania dynamiki ich cyklu życia, czyli przejść pomiędzy stanami (w szerokim kontekście zachowania systemu, przypadków użycia etc.), powinno się zastosować maszyny stanów zachowania- nie powinno się natomiast modelować zachowania z użyciem elementów modelu strukturalnego (np. atrybutów klasy).

1.8.5 PROTOTYP FUNKCJONALNY

Prototyp funkcjonalny jest uzupełnieniem modeli analizy systemowej – prezentuje zakres informacyjny formatek i przejść między nimi. Prototyp funkcjonalny obejmuje wszystkie przypadki użycia i wszystkie akcje ich przebiegów, w tym możliwe nawigacje do innych przypadków użycia (włączanych, rozszerzających).

1.9 PROJEKT

1.9.1 RODZAJE MODELI

W ramach dokumentacji technicznej powinny powstać następujące modele:

- Model komponentów i interakcji – opracowany w UML za pomocą:
  - Modelu pakietów i komponentów
  - Modelu sekwencji i/lub komunikacji dla reprezentatywnych typów komponentów i interakcji
- Fizyczny model danych – opracowany za pomocą modelu klas
- Model rozmieszczenia – opracowany za pomocą modelu rozmieszczenia
- Model platformy – opracowany za pomocą modelu rozmieszczenia

1.9.2 ZASTOSOWANIE MDA

Wszelkie modele powstałe w ramach projektowania, o których mowa powyżej powinny być wynikiem przekształcenia modeli analizy systemowej (wywodzą się z nich, są ich pochodną). W przypadku pewnych modeli może to nie być zależność wprost, ponieważ mogą one powstawać z innych modeli powstałych w ramach projektowania.

Wraz z dokumentacją modelu wynikowego musi zostać dostarczona wiedza o:

- modelu źródłowym, z którego wykonana została transformacja,
- modelu platformy, na jaką wykonana została transformacja,
- regułach transformacji, które zostały zastosowane aby z modelu źródłowego uzyskać model wynikowy.

1.9.3 TRANSFORMACJA

Transformacja musi posiadać jednoznacznie wyspecyfikowane reguły transformacji, które z kolei powinny powoływać się (wskazywać) na wzorce lub w przypadku braku odpowiedniego wzorca, powinien zostać wskazany inny wzorzec ogólnie znany, opublikowany. Wówczas, gdy przyjęty został wzorzec własny wykonawcy, musi zostać jednoznacznie i precyzyjnie zdefiniowana specyfikacja tego wzorca.

Na przykład przy transformacji modelu dziedziny systemu do modelu bazy danych, reguły transformacji typów danych użytych w klasach modelu dziedziny, specjalizacji klas na schemat docelowy (model bazy) będą wymagały wskazania/wyspecyfikowania wzorców.

1.9.4 ŚLADOWANIE TRANSFORMACJI

Niezależnie od przyjętego modelu transformacji, dla elementów modelu wynikowego, musi istnieć możliwość sprawdzenia, z jakich elementów modelu źródłowego, wywodzi się dany element. Czyli: musi zostać zapewnione śladowanie.

1.10 STANDARDY WYKONAWCZE

W ramach dokumentacji projektowej należy opracować następujące standardy:

1. Zasady inicjalizacji i utrzymania repozytorium modeli
2. Szczegółowe zasady notacji dla Analizy Systemowej i Projektu, w szczególności:
    a. Zasady notacji dla poszczególnych typów diagramów
    b. Reguły transformacji i śladowanie
    c. Specyfikacja stereotypów/profili i ich semantyki
    d. Format dokumentacji w formie dokumentów pakietu Office tworzonej automatycznie na podstawie modeli
3. Szczegółowe zasady tworzenia prototypów funkcjonalnych (w tym wykorzystania istniejących)
4. Standard tworzenia Analitycznego Opisu Modyfikacji
5. Standard tworzenia Planu Testów Akceptacyjnych
6. Standard tworzenia dokumentacji (użytkownika, administratora etc.)
7. Standard interfejsu użytkownika
8. Standard specyfikacji interfejsów programistycznych
9. Standard opisu schematu danych
10. Standard opisu kodu źródłowego

2. WYMIAROWANIE MODYFIKACJI METODĄ COSMIC

2.1 WYMIAROWANIE COSMIC

Podstawą dla obliczania złożoności oprogramowania dostarczanego w ramach Modyfikacji jest metoda COSMIC w wersji 3.0.1 zdefiniowana w dokumencie "*Measurement Manual: The COSMIC Functional Size Measurement Method Version 3.0.1*" opublikowanym przez organizację COSMIC oraz modyfikacje i rozszerzenia opisane w niniejszym dokumencie. W przypadku powstania w trakcie realizacji Umowy nowej wersji dokumentu „*Measurement Manual*" będzie ona wykorzystywana jedynie w przypadkach, gdy naprawia ona wady dokumentu "*Measurement Manual: The COSMIC Functional Size Measurement Method Version 3.0.1"*.

Przez metodę COSMIC rozumie się metodę opracowaną i utrzymywaną przez organizację *Common Software Measurement International Consortium* (www.cosmicon.com).

2.1.1 INTEGRACJA WYMIAROWANIA Z ANALIZĄ SYSTEMOWĄ

W celu maksymalnej integracji procesu wymiarowania z procesem analitycznym dla stosowania metody COSMIC w KRUS przyjęto następujące fundamentalne założenia:

1. **Proces funkcjonalny jest tożsamy z przypadkiem użycia** . Oznacza to, iż zdefiniowane w ramach analizy systemowej przypadki użycia należy traktować, wprost jako procesy funkcjonalne COSMIC z wszelkimi tego konsekwencjami odnośnie zasad identyfikacji przesunięć danych w ramach procesu funkcjonalnego (w ramach przypadku użycia).
2. **Grupa danych jest tożsama z klasą modelu dziedziny systemu**. Oznacza to, iż zdefiniowane w ramach analizy systemowej klasy należy traktować, jako grupy danych COSMIC z wszelkimi tego konsekwencjami odnośnie zasad identyfikacji przesunięć grup danych (klas modelu dziedziny systemu).
3. **Użytkownik funkcjonalny jest tożsamy z aktorem** posiadającym asocjację (wprost lub przechodnio) do danego przypadku użycia. Oznacza to, iż inicjatorami zdarzeń inicjującymi procesy funkcjonalne są aktorzy inicjujący wykonanie danego przypadku użycia – mogą to być zarówno aktorzy będący użytkownikami systemu jak i aktorzy będący oprogramowaniem wykorzystującym dany przypadek użycia.

2.1.2 INTEGRACJA WYMIAROWANIA Z ZARZĄDZANIEM

Zgodnie z podręcznikiem metody COSMIC, pełna identyfikacja przesunięć w ramach procesów funkcjonalnych wymaga zdefiniowania wszystkich wymagań funkcjonalnych. W przypadku modelu przypadków użycia oznacza to, iż pełna identyfikacja przesunięć klas z modelu dziedziny wymaga wyspecyfikowania wszystkich przebiegów przypadków użycia. Przebiegi przypadków użycia oraz pełny model dziedziny systemu są opracowywane w ramach Analitycznego Opisu Modyfikacji (AOM).

Zakłada się dwa etapy wymiarowania modyfikacji:

- Szacowanie wstępne
- Wymiarowanie pełne

Szacowanie wstępne jest przedstawiane przez Wykonawcę wraz z Ofertą wykonawczą na Modyfikację. Szacowanie wstępne zawiera:

1. Zidentyfikowane przypadki użycia: **nowe, modyfikowane, usuwane**.
2. Informacje o przypisaniu zmiany istniejącego przypadku do jednej z kategorii złożoności zmiany: **znacząca, średnia, mała, bardzo mała.**
3. Informacje o przypisaniu danego przypadku użycia do jednej z grup złożoności: **złożony, średni, prosty**.

Wymiarowanie pełne jest integralną częścią AOM – przesunięcia klas modelu dziedziny systemu są identyfikowane w oparciu o przebiegi przypadków użycia zgodnie z zasadami COSMIC i zasadami niniejszego podręcznika.

2.1.3 ZAKRES STOSOWANIA METODY COSMIC

Rozmiar funkcjonalny oprogramowania jest definiowany, jako rozmiar oprogramowania opracowanego w wyniku implementacji konkretnego zbioru wymagań funkcjonalnych użytkownika.

Metoda COSMIC pozwala oszacować rozmiar funkcjonalny oprogramowania lub rozmiar funkcjonalny zmiany oprogramowania przy założeniu, iż definicja funkcjonalności nowej lub zmienianej wynika z wymagań funkcjonalnych użytkownika.

Rozmiar funkcjonalny oprogramowania jest niezależny od wymagań niefunkcjonalnych, w tym od technologii, architektury i sposobu realizacji danego oprogramowania. Oznacza to również, iż wszystkie zadania projektu / modyfikacji mające na celu wprowadzenie / zmianę technologii, architektury czy sposobu dostarczania oprogramowania nie podlegają szacowaniu metodą COSMIC. Jednakże standardowe prace wynikające z uwzględniania wymagań niefunkcjonalnych w trakcie realizacji zmian funkcjonalnych traktuje się jako prace wliczone w cenę punktu funkcyjnego:

- Migracji danych niezbędnej dla uruchomienia nowej / zmodyfikowanej funkcjonalności,
- Utworzenia lub aktualizacji danych w bazie danych na potrzeby działania nowej / zmodyfikowanej funkcjonalności,
- Utworzenia lub aktualizacji parametrów definiujących działanie aplikacji (w kontekście technologicznym jak i biznesowym) niezbędnych dla działania nowej / zmodyfikowanej funkcjonalności,
- Utworzenia lub aktualizacji słowników wspierających działanie aplikacji (w kontekście technologicznym jak i biznesowym) niezbędnych dla działania nowej / zmodyfikowanej funkcjonalności,
- Optymalizacji oprogramowania / struktury bazy danych w zakresie obowiązującego SLA,
- Prac konfiguracyjnych dotyczących elementów architektury, infrastruktury i oprogramowania standardowego niezbędnych dla uruchomienia nowej / zmodyfikowanej funkcjonalności,

- Zmiana wyglądu, koloru, układu kontrolek i danych na interfejsie użytkownika wynikające z nowych / zmodyfikowanych funkcjonalności.
- Przygotowanie dokumentacji analitycznej, technicznej, architektonicznej, testowej, administratora, użytkownika
- Wdrożenie, konfiguracja, parametryzacja
- Przygotowanie oprogramowania i jego testy
- Podstawowe szkolenia

W ramach funkcjonalności, dla której ma zastosowanie metoda COSMIC zakłada się, iż koszt punktu funkcyjnego (CFP) określony w umowie pomiędzy Zamawiającym i Wykonawcą zawiera w sobie koszt całego cyklu wytwórczego związanego z wytworzeniem oprogramowania o złożoności jednego punktu funkcyjnego.

2.2 SZACOWANIE WSTĘPNE

Szacowanie wstępne jest opracowywane na etapie uzgadniania Modyfikacji i jest przedstawiane przez Wykonawcę w **Ofercie wykonania**.

Liczba CFP (tj. punktów funkcyjnych) dla wytwarzanego lub zmienianego w Modyfikacji oprogramowania jest ustalana **z dokładnością do przypadku użycia**.

2.2.1 ISTNIEJĄCE I ZMIENIANE PRZYPADKI UŻYCIA

Dla istniejących lecz zmienianych przypadków użycia liczbę CFP zmiany wyznacza się w następujący sposób:

1. Jako bazową liczbę CFP dla zmienianego istniejącego przypadku użycia przyjmuje się jego **aktualny rozmiar** w CFP.
2. Dokonuje się wyboru **współczynnika złożoności zmiany** (danego zmienianego przypadku użycia) w oparciu o wiedzę ekspercką Wykonawcy. Współczynniki złożoności zmiany przypadku użycia (w zależności od typu zmiany) mogą być następujące:
   - Znacząca: **90%**
   - Średnia: **60%**
   - Mała: **30%**
   - Bardzo mała: **10%**
3. **Liczba CFP modyfikacji** dla danego istniejącego zmienianego przypadku użycia jest iloczynem:
   - bazowej liczby CFP danego przypadku użycia oraz
   - współczynnika złożoności zmiany.

Wyliczając rozmiary w oparciu o współczynnik zmiany stosuje się zaokrąglenia matematyczne by przedstawić końcowy wynik w postaci liczby całkowitej.

2.2.2 NOWE PRZYPADKI UŻYCIA

Dla nowych przypadków użycia liczbę CFP zmiany wyznacza się w następujący sposób:

2.2.2.1 NOWE PRZYPADKI UŻYCIA W ISTNIEJĄCYM PAKIECIE

1. Dokonuje się określenia **typu złożoności** przypadku użycia w kontekście pakietu funkcjonalnego, w ramach, którego przypadek użycia ma być umiejscowiony. Typ złożoności określa się w oparciu o wiedzę ekspercką Wykonawcy.
2. Dla modułów wylicza się następujące **wskaźniki**, zaokrąglając matematycznie do postaci liczby całkowitej:
   - N_Z = Średnia liczba CFP spośród **30%** przypadków użycia w danym module o największej liczbie CFP
   - $N_{ŚR}$ = Średnia liczba CFP spośród kolejnych **40%** przypadków użycia w danym module o największej liczbie CFP
   - N_P = Średnia liczba CFP spośród pozostałych **30%** przypadków użycia w danym module
3. Dokonuje się przypisania (na podstawie typu złożoności) liczby CFP na podstawie poniższej tabeli:

Typ złożoności nowego przypadku użycia	Liczba CFP dla nowego przypadku użycia w module
Złożony	N_Z
Średni	$N_{ŚR}$
Prosty	N_P

2.2.2.2 NOWE PRZYPADKI UŻYCIA W NOWYM PAKIECIE

4. W przypadku gdy dla tworzonych przypadków użycia tworzony jest nowy pakiet (np. w ramach projektu realizacji nowego oprogramowania) przyjmuje się następujące wartości **liczby CFP**:
   - złożony: **16 CFP**
   - średni: **11 CFP**
   - prosty: **6 CFP**

2.2.3 USUWANE PRZYPADKI UŻYCIA

Dla usuwanych przypadków użycia liczbę CFP zmiany wyznacza się w następujący sposób:

1. Jeden CFP za każdy usuwany przypadek użycia.

2.2.4 WYNIKI SZACOWANIA WSTĘPNEGO

Przedstawiając wyniki szacowania wstępnego Wykonawca przedstawia wszystkie dane opisowe oraz pośrednie niezbędne dla wyliczenia tej złożoności, czyli:

1. Sposób wyliczenia średnich złożoności.
2. Źródłowe dane w oparciu, o które zdefiniowano średnie oraz złożoności poszczególnych przypadków użycia.
3. Listę przypadków użycia objętych modyfikacją wraz z:
    a. Unikalnym i stałym identyfikatorem przypadku użycia.
    b. Identyfikatorami wymagań dla danego przypadku użycia.
    c. Złożonością danego przypadku sprzed modyfikacji oraz kategorią zmiany dla modyfikowanego przypadku użycia wraz z wyliczeniem liczby, CFP zmiany.
    d. Kategorią złożoności dla nowych przypadków użycia wraz ze wskazaniem liczby CFP w oparciu o wyznaczone średnie pakietu.

2.3 WYMIAROWANIE PEŁNE

Wymiarowanie pełne jest integralnym elementem Analitycznego Opisu Modyfikacji (AOM) i jest przedstawiane przez Wykonawcę do odbioru wraz AOM.

Wymiarowanie pełne jest realizowane w oparciu o zasady COSMIC oraz zasady uszczegóławiające opisane w niniejszym dokumencie.

Podstawą do przeprowadzenia wymiarowania są przypadki użycia, a w szczególności akcje przebiegów tychże przypadków użycia wraz z regułami przetwarzania oraz analitycznym modelem dziedziny w postaci klas i atrybutów, stanowiącymi elementy przekazywanego AOM.

Zgodnie z powyższym, w celu identyfikacji przesunięć grup danych COSMIC w ramach danego przypadku użycia wymagane jest wcześniejsze zdefiniowanie przebiegów przypadków użycia, które powinny zawierać opis interakcji pomiędzy użytkownikiem i systemem oraz sposób przetwarzania danych. Przebiegi te powinny posiadać również ewentualne odwołania do przypadków użycia w relacjach "gen-spec", "include" oraz "extend".

Zasady identyfikacji oraz modelowania przypadków użycia i klas modelu dziedziny zostały opisane w dokumencie „Standardy". Zakłada się tym samym, iż przed przystąpieniem do wymiarowania, przypadki użycia oraz klasy modelu dziedziny są zdefiniowane poprawnie.

Wymiarowanie pełne powinno być realizowane odrębnie dla każdego z przekazywanych do akceptacji przypadków użycia.

2.3.1 ISTNIEJĄCE I ZMIENIANE PRZYPADKI UŻYCIA

W celu wymiarowania zmiany przypadku użycia w AOM koniecznie jest:

- Zdefiniowanie modelu przypadku użycia zgodnie z wymaganym zakresem modelu dla przypadku użycia wraz z oznaczeniem na jego modelu zmian (z dokładnością do reguł, akcji przebiegu).

- Zdefiniowanie modelu dziedziny zgodnie z wymaganym zakresem opisu modelu dziedziny w AOM wraz z oznaczeniem na tym modelu zmian (z dokładnością do klas, atrybutów i asocjacji).

2.3.2 NOWE PRZYPADKI UŻYCIA

W celu zwymiarowania nowego przypadku użycia w AOM koniecznie jest:

- Zdefiniowanie przypadku użycia zgodnie z wymaganym zakresem modelu przypadku użycia.
- Zdefiniowanie modelu dziedziny zgodnie z wymaganym zakresem modelu dziedziny.

2.3.3 USUWANE PRZYPADKI UŻYCIA

W celu oszacowania usuwanego przypadku użycia w AOM konieczne jest oznaczenie tego przypadku na istniejącym modelu jako przypadku usuwanego.

2.3.4 STRATEGIA ZASTOSOWANIA METODY COSMIC

1. Wymiarowanie Modyfikacji realizowane jest w oparciu o **metodę COSMIC w wersji 3.0.1**.
2. Wymiarowanie Modyfikacji ma na celu:
    a. wyznaczanie rozmiaru Modyfikacji celem określania wartości zlecenia jej realizacji,
    b. weryfikacji harmonogramu realizacji w oparciu o produktywność Wykonawcy,
    c. zbierania danych statystycznych dla wewnętrznych potrzeb KRUS.
3. Proces wymiarowania realizowany jest w oparciu o model analityczny, na poziomie przebiegu przypadków użycia opracowanych zgodnie ze **standardami analitycznymi** (opisanymi w osobnym dokumencie „Standardy"). Model analizy systemowej w AOM danej Modyfikacji stanowi jedyne źródło Wymagań Użytkownika Funkcjonalnego (FUR) dla przeprowadzenia wymiarowania metodą COSMIC.
4. Proces wymiarowania realizowany jest w oparciu o zasady opisane w metodzie COSMIC oraz reguły zawarte poniżej.

2.3.5 REGUŁY

1. Przypadek użycia jest tożsamy z procesem funkcjonalnym COSMIC, przy czym przypadek generalizujący i jego specjalizacje traktuje się rozdzielnie w kontekście analizy unikalności przesunięć COSMIC. Przebiegi danego przypadku użycia traktuje się jak jeden proces funkcjonalny o różnych wariantach przebiegu.
2. Przesunięć COSMIC przypadku generalizującego nie zlicza się do rozmiaru przypadku specjalizującego.
3. W przypadku opcjonalnych rozszerzeń przypadku użycia (relacja *extend*) w przebiegu akcji potencjalnie rozszerzanego przypadku użycia nie określa się przesunięć ani manipulacji COSMIC obrazujących możliwość tego rozszerzenia. Właściwe przesunięcia i manipulacje obrazujące rozszerzenie są wymiarowane w rozszerzających przypadkach użycia zgodnie z ich przebiegami.

4. W przypadku włączeń (relacja *include*) przypadku użycia w przebiegu w którym te włączanie jest specyfikowane zlicza się odpowiednią liczbę przesunięć typu E i X – po jednym dla każdej przenoszonej grupy danych, przy czym – jeśli w danym przebiegu określony przypadek użycia jest włączany wielokrotnie to zlicza się wyłącznie jedno przesunięcie danego typu dla każdej grupy danych – niezależnie od liczby włączeń.
5. Dla generalizującego przypadku użycia, który jest abstrakcyjny nie identyfikuje się przesunięcia inicjalnego.
6. Rozmiar zmiany usuwanego przypadku użycia wynosi 1 CFP.
7. Przesunięcia danych zawsze są skojarzone z konkretną akcją przebiegu przypadku użycia.
8. Za poprawnie zdefiniowaną grupę danych można uznać wyłącznie taką, która ma swoje odzwierciedlenie w modelu dziedziny systemu danego modułu (istniejące wyjątki od tej reguły są opisane w regułach niniejszego Podręcznika).
9. W sytuacji, gdy przypadek użycia operuje wyłącznie na generalizacji danej grupy klas to, jako grupę danych traktuję się tą generalizację. W sytuacji, gdy przypadek użycia operuje na specjalizacjach danej grupy klas to, jako grupy danych traktuje się każdą ze specjalizacji (generalizacja jest pomijana, jako grupa danych gdyż traktuje się ją w takiej sytuacji, jako objętą specjalizacją).
10. Enumeracji i słowników prostych (transformacji kod – wartość) nie traktuje się, jako grup danych.
11. Pliki importowane lub eksportowane przez aplikację traktuje się, jako jedną grupę danych – tzn. nie dekomponuje na grupy danych struktury tych plików. Przykładem pliku eksportowanego jest również wygenerowany plik raportu.
12. Kryteria wyszukania określonych grup danych traktuje się, jako jedną grupę danych, unikalną dla danego wyszukiwania. Dane pomocnicze prezentowane przez system w celu ułatwienia wprowadzenia kryteriów szacuje się, jako odpowiednie przesunięcia R i X.
13. Oszacowanie złożoności utworzenia postaci raportu polega na zidentyfikowaniu przesunięć typu „X" tworzących finalną postać raportu:

    a. Jedno przesunięcie dla każdej grupy danych, której atrybuty są przedstawiane na raporcie – niezależnie od liczby tych atrybutów.

    b. Jedno przesunięcie dla każdej agregacji względem określonego kryterium niezależnie od liczby atrybutów wyliczanych w oparciu o tą agregacje – przy czym niezależnie liczone są agregacje po wierszach i kolumnach.

    c. Jedno zbiorcze przesunięcie dla parametrów, standardowych oznaczeń raportu oraz podsumowań wierszy / kolumn bazujących na już oszacowanych prezentacjach.
14. W przypadku zmian przypadku użycia:
    a. Zmiana formy wizualnego przedstawienia raportu jest oznaczana, jako modyfikacja przesunięcia X prezentującego parametry i oznaczenia raportu o ile nie jest to zmiana globalna dla danego mechanizmu generowania raportów.
    b. W sytuacji, gdy zmiana polega na zmianie manipulacji związanej z danym przesunięciem to zmiana ta musi być odzwierciedlona w regułach przypadku użycia.
    c. Rozmiar zmiany usuwanej akcji w przypadku użycia wynosi 1 CFP niezależnie od liczby przesunięć z tą akcją skojarzonych.

2.4 WYMIAROWANIE WYMAGAŃ POZAFUNKCJONALNYCH MODYFIKACJI

Metoda COSMIC, zgodnie ze swoją definicją służy wymiarowaniu wymagań funkcjonalnych. Niektóre Zlecenia modyfikacji mogą jednakże zawierać w zakresie prac realizację wymagań pozafunkcjonalnych, które nie są związane z realizacją zmian funkcjonalnych.

W przypadku obecności w zakresie Uzgodnienia Projektowego w/w wymagań pozafunkcjonalnych w ramach Oferty wykonania Wykonawca zobowiązany jest podać liczbę godzin pracochłonności na ich realizację wraz z wyczerpującym uzasadnieniem. Iloraz tej liczby godzin przez wydajność procesu produkcji zaoferowanego przez Wykonawcę w ofercie stanowi oszacowanie kosztów (w CFP) realizacji wymagań pozafunkcjonalnych – zarówno w Szacowaniu wstępnym jak i Wymiarowaniu pełnym. Tak wyznaczona liczba CFP dla wymagań pozafunkcjonalnych dodawana jest do rozmiaru Modyfikacji wyznaczonego dla wymagań funkcjonalnych.