

Metodyka zarządzania wymaganiami dla systemu Geoportal 2 wraz z systemami dziedzinowymi

Informacje o dokumencie:

Autor	Infovide-Matrix S.A.
Tytuł	Metodyka zarządzania wymaganiami dla Systemu Geoportal 2 wraz z systemami dziedzicznymi
Projekt	Geoportal 2
Wersja	2.4
Liczba stron	52
Data utworzenia	2013-08-06
Data ost. modyfikacji	2014-05-22
Odbiorca dokumentu	Główny Urząd Geodezji i Kartografii
Kontakt do uwag	mmazur@ivmx.pl
Nazwa pliku	G2_Metodyka zarządzania wymaganiami v.2.4.docx

Wersja	Data Wersji	Autor	Opis
0.9	2013-08-06	Infovide-Matrix S.A.	Opracowanie inicjalnej wersji dokumentu.
1.0	2013-08-16	Infovide-Matrix S.A.	Uzupełnienie dokumentu o kolejne sekcje, uwzględnienie uwag Zamawiającego z dnia 6.08.2013 r.
1.1	2013-08-26	Infovide-Matrix S.A.	Uzupełnienie dokumentu o kolejne sekcje.
2.0	2013-09-10	Infovide-Matrix S.A.	Uzupełnienie dokumentu o uwagi przekazane podczas Rady Architektury z dn. 04.09.2013 r.
2.1	2013-09-27	Infovide-Matrix S.A.	Uwzględnienie uwag Zamawiającego z dnia 27.09.2013 r.
2.2	2014-02-12	GUGiK	Ujednolicenie nazewnictwa dla Projektu Geoportal2
2.3	2014-04-14	Infovide-Matrix S.A.	Wprowadzenie zmian związanych z implementacją metodyki
2.4	2014-05-22	Infovide-Matrix S.A.	Uspójnienie metodyki z zapisami metody szacowania rozmiaru oprogramowania

Spis treści

1	WPROWADZENIE	5
1.1	Cel dokumentu	5
1.2	Zakres dokumentu.....	5
1.3	Słownik użytych terminów	5
1.4	Dokumenty powiązane.....	7
2	ANALIZA DOSTĘPNYCH NARZĘDZI	8
3	METODYKA ZARZĄDZANIA WYMAGANIAMI.....	14
3.1	Cykl życia wymagań	15
3.1.1	<i>Identyfikacja wymagań</i>	<i>16</i>
3.1.2	<i>Definiowanie wymagań.....</i>	<i>19</i>
3.1.3	<i>Zatwierdzenie wymagań</i>	<i>22</i>
3.1.4	<i>Realizacja wymagania</i>	<i>24</i>
3.1.5	<i>Ponowne użycie wymagań</i>	<i>24</i>
3.1.6	<i>Modyfikacja wymagań</i>	<i>25</i>
3.2	Statusy wymagań.....	27
3.3	Śledzenie wymagań	30
3.4	Weryfikacja wymagań	31
4	JEDNOLITA BAZA WYMAGAŃ	33
4.1	Metamodel	33
4.2	Organizacja ról w zarządzaniu wymaganiami.....	38
4.3	Struktura bazy wymagań.....	41
4.4	Procedury bezpieczeństwa.....	42
4.4.1	<i>Procedura umieszczania wymagania</i>	<i>43</i>
5	PLAN WDROŻENIA METODYKI ZARZĄDZANIA WYMAGANIAMI	44
5.1	Źródła wymagań	44
5.2	Szablony raportów.....	45
5.3	Etapy wdrażania metodyki	46
5.4	Udział Wsparcia i Wykonawców w zarządzaniu wymaganiami	48
6	KORZYŚCI Z ZASTOSOWANIA METODYKI ZARZĄDZANIA WYMAGANIAMI.....	51

Spis rysunków

Rysunek 1: Podstawowy cykl życia wymagania.....	15
Rysunek 2: Skrócona ścieżka cyklu życia wymagania	26
Rysunek 3: Diagram stanów wymagań.....	28
Rysunek 4: Przypadek użycia	33
Rysunek 5: Przypisanie potrzeby do przypadku użycia	34
Rysunek 6: Przypisanie wymagania do przypadku użycia	35
Rysunek 7: Wymaganie	35
Rysunek 8: Aktor.....	36
Rysunek 9: Asocjacja pomiędzy aktorem a przypadkiem użycia	36
Rysunek 10: Związek includes między przypadkami użycia	37
Rysunek 11: Związek extends między przypadkami użycia	37
Rysunek 12: Związek generalization między przypadkami użycia	37
Rysunek 13: Diagram przypadków użycia	38
Rysunek 14: Diagram wymagań	38
Rysunek 15: Elementy wykorzystane do tworzenia struktury bazy wymagań	41
Rysunek 16: Ogólna struktura bazy wymagań	42
Rysunek 17: Przebieg współpracy z Wykonawcą	49

Spis tabel

Tabela 1: Zastosowane pojęcia i skróty.....	5
Tabela 2: Wykaz dokumentów powiązanych	7
Tabela 3: Porównanie metod pracy nad repozytorium.....	8
Tabela 4: Cechy wersji Enterprise Architect.....	10
Tabela 5: Procedura tworzenia modelu przypadków użycia.....	17
Tabela 6: Przejścia pomiędzy stanami wymagań	29

1 Wprowadzenie

1.1 Cel dokumentu

Dokument przeznaczony jest dla zespołu organizującego i wdrażającego koncepcję SIG w Głównym Urzędzie Geodezji i Kartografii.

Celem dokumentu jest przedstawienie metodyki zarządzania wymaganiami dla systemu Geoportal wraz z systemami dziedzinowymi. Metodyka pozwoli usystematyzować i ujednolicić zarządzanie wymaganiami w poszczególnych projektach i stanowić będzie podstawę dla budowy jednolitego repozytorium wymagań Geoportalu wraz z systemami dziedzinowymi.

1.2 Zakres dokumentu

Dokument obejmuje proces zarządzania wymaganiami oraz ich podstawowe funkcje. Dokument składa się z następujących rozdziałów:

- Rozdział 1 Wprowadzenie – rozdział zawiera opis celu dokumentu, słownik pojęć oraz wykaz dokumentów powiązanych
- Rozdział 2 Analiza dostępnych narzędzi – rozdział zawiera analizę dostępnych narzędzi wspierających zarządzanie wymaganiami
- Rozdział 3 Metodyka zarządzania wymaganiami – rozdział zawiera opis cyklu zarządzania wymaganiami wraz z procedurami zarządzania wymaganiami w ich poszczególnych fazach życia
- Rozdział 4 Jednolita baza wymagań – rozdział zawiera najważniejsze założenia i wytyczne dla jednolitej bazy wymagań Geoportalu
- Rozdział 5 Plan wdrożenia metodyki zarządzania wymaganiami – rozdział zawiera plan wdrożenia metodyki zawartej w niniejszym dokumencie
- Rozdział 6 Korzyści z zastosowania metodyki zarządzania wymaganiami – rozdział zawiera korzyści jakie wynikają z wdrożenia niniejszej metodyki

1.3 Słownik użytych terminów

Tabela 1: Zastosowane pojęcia i skróty

Obszar organizacyjno-prawny	
Aktor	Użytkownik bądź system wchodzący w interakcje z systemem
Aplikacja EMUiA	Aplikacja do prowadzenia Ewidencji Miejscowości, Ulic i Adresów wytworzona w ramach projektu TERYT2 przez GUGiK i udostępniana nieodpłatnie gminom na mocy zawieranych przez Głównego Geodetę Kraju porozumień z gminami.
CSV	ang. Comma Separated Values - wartości rozdzielone przecinkiem, jest to format przechowywania danych w plikach tekstowych

Obszar organizacyjno-prawny	
Cykl życia wymagania	Sekwencja działań zmierzających do wprowadzenia wymagania do bazy wymagań
EA	Enterprise Architect - narzędzie do modelowania UML
GBDOT	Krajowego Systemu Zarządzania Bazą Danych Obiektów Topograficznych
GUGiK	Główny Urząd Geodezji i Kartografii
ISOK	Informatyczny System Osłony Kraju przed nadzwyczajnymi zagrożeniami
K-GESUT	Krajowa baza danych geodezyjnej ewidencji sieci uzbrojenia terenu
OPZ	Dokument z opisem przedmiotu zamówienia
Przypadek użycia	Funkcjonalność systemu
Repozytorium	Wydzielone miejsce w infrastrukturze GUGiK służące do gromadzenia, aktualizacji wymagań związanych z SIG
SIG	System informacyjny projektów inwestycyjnych prowadzonych w Głównym Urzędzie Geodezji i Kartografii, tj. projektów Geoportal2, TERYT2, GBDOT oraz ISOK.
SOPZ	Szczegółowy opis przedmiotu zamówienia
Stany wymagań	Sytuacja w jakiej może być wymaganie
SVN	Subversion, system kontroli wersji
Śledzenie wymagań	Monitorowanie zmian w wymaganiach przechodzących przez cykl życia wymagań
TFS	Team Foundation Server do kontroli wersji
UML	ang. Unified Modeling Language - zunifikowany język modelowania
Wymaganie	Własność systemu informatycznego, którą system ten musi posiadać, aby spełnić oczekiwania zamawiającego. Własność ta może być związana z określonym sposobem funkcjonowania systemu (wymaganie funkcjonalne) lub cechami jakościowymi narzuconymi na system (wymaganie pozafunkcjonalne). System informatyczny spełnia wymaganie, jeśli zostanie potwierdzone że posiada wszystkie własności, które są określone wymaganiami oraz spełnia cechy jakościowe.

Obszar organizacyjno-prawny	
Wymaganie do ponownego użycia	Wymaganie rekomendowane do kolejnego wykorzystania w ramach budowy systemów w GUGiK
Zespół Zamawiającego	Przedstawiciele Zamawiającego lub Zamawiającego i Wsparcia uczestniczący z realizacji Projektu

1.4 Dokumenty powiązane

Tabela 2: Wykaz dokumentów powiązanych

Lp.	Nazwa dokumentu	Nazwa pliku	Wersja
1	Raport z diagramów	Zestawienie_diagramow.rtf	1.0
2	Raport z przypadków użycia	Zestawienie_przypadkowuzycia.rtf	1.0
3	Raport z wymagań	Zestawienie_wymagan.rtf	1.0
4	Raport przeprowadzonych zmian w repozytorium bazy wymagań	Zestawienie_zmian.rtf	1.0

2 Analiza dostępnych narzędzi

W niniejszym rozdziale przedstawiona została analiza narzędzi, które są w dyspozycji Głównego Urzędu Geodezji i Kartografii i mogą być wykorzystywane do zarządzania wymaganiami w czasie realizacji projektu.

Analiza została przeprowadzona w celu wsparcia dokonania wyboru optymalnego pakietu na potrzeby zarządzania wymaganiami w Głównym Urzędzie Geodezji i Kartografii.

Wśród narzędzi użytkowanych przez pracowników Głównego Urzędu Geodezji i Kartografii narzędziem najlepiej odpowiadającym potrzebom związanym z zarządzaniem wymaganiami jest Enterprise Architect firmy Sparx Systems. Jest to jedno z popularniejszych narzędzi dostępnych na rynku, udostępniające środowisko do analizy i projektowania oprogramowania przy użyciu języków modelowania, m.in. języka UML.

Projekt tworzony w programie Enterprise Architect jest przechowywany w formie relacyjnej bazy danych. Zawartość modelu może być poddana kontroli wersji przy użyciu SVN lub TFS. Enterprise Architect posiada funkcjonalność Team Review służącą do wspierania pracy grupowej. Team Review pozwala na współdzielenie jednego repozytorium EA (w formie bazy danych lub współdzielonego pliku .eap) i wymianę informacji pomiędzy użytkownikami. Team Review zalecane jest dla grup projektowych gdzie z repozytorium EA korzystają aktywnie więcej niż 3 osoby, są zlokalizowani w różnych miejscach, a tworzone modele stanowią integralną część dokumentacji projektowej. Także kolejną zaletą jest dostęp do modelu w trybie tylko do odczytu (Enterprise Architect Lite). EA zapewnia łatwy import wymagań z MS Word, poprzez generację do modelu.

Istnieje dwójaka możliwość pracy na tym samym repozytorium poprzez:

- replikację modelu,
- współdzielony plik EAP w udostępnionym udziale sieciowym.

W poniższej tabeli przedstawione zostało porównanie obu modeli pracy.

Tabela 3: Porównanie metod pracy nad repozytorium

REPLIKACJE MODELU	WSPÓŁDZIELONY PLIK EAP
ZALETY	
nie jest potrzebne stworzenie i utrzymywanie infrastruktury i usług centralnych w postaci na przykład wspólnej bazy danych, serwera kontroli wersji (SVN, czy TFS)	członkowie zespołu zawsze mogą zobaczyć najnowsze zmiany i status projektu
nie ma również potrzeby wystawiania żadnych usług z sieci lokalnej do Internetu	należy przygotować środowisko
zorganizowanie pracy w oparciu o repliki nie wymaga dużego nakładu pracy i czasu	
rekomendowana przy uruchomieniu dedykowanej bazy danych jako repozytorium EA	

REPLIKACJE MODELU	WSPÓLDZIELONY PLIK EAP
WADY	
zmiany wprowadzone przez innych użytkowników nie są widoczne od razu, zmiany będą widoczne dopiero w kolejnej iteracji, gdy repliki zostaną scalone	zawieszenie pracy przy awarii sieci
potrzeba organizacyjnego przypisania właścicieli określonych pakietów, aby uniknąć wzajemnego nadpisywania tych samych elementów	użytkownicy dołączają swoje zmiany do repozytorium tylko w przypadku gdy jest to konieczne
potrzeba zaangażowania administratora modelu do zarządzania replikami	potrzeba zaangażowania administratora modelu do zarządzania plikami
edycja modelu zorganizowana w iteracje, gdzie iteracja rozpoczyna się wygenerowaniem replik a kończy scaleniem ich	
W celu skutecznego edytowania modelu konieczne jest opracowanie i wdrożenie odpowiedniej procedury projektowej	
ryzyko wzajemnego nadpisywania zmian w modelu	

Narzędzie Enterprise Architect występuje w różnych wersjach. Poniżej przedstawione zostało porównanie najpopularniejszych wersji systemu.

Ultimate Edition jest wersją rozszerzoną systemu Enterprise Architect. Wspiera takie obszary jak: modelowanie biznesowe, inżynierię oprogramowania, projektowanie i implementację systemów informatycznych. Jako istotne jego funkcjonalności wyróżnić można: generowanie dokumentacji, śledzenie zmian w projekcie, tworzenie współdzielonego repozytorium bazodanowego, zarządzanie projektami, modelowanie interfejsów użytkownika oraz generowanie kodu na podstawie zamodelowanych procesów. Pakiet Ultimate Edition skierowany jest do zaawansowanych użytkowników i osób specjalizujących się w różnorodnych dziedzinach biznesowych.

Systems Engineering jest to także wersja rozszerzona z przeznaczeniem do projektowania systemów wytwarzanych z wykorzystaniem UML, SysML, DDS i technologii pokrewnych. Pakiet Systems Engineering skierowany jest do zaawansowanych użytkowników wytwarzających nowoczesne i zaawansowane rozwiązania informatycznych.

Business & Software Engineering również jest to wersja rozszerzona systemu Enterprise Architect. Jego funkcjonalności nakierowane są na zarządzanie zakresem projektu oraz zespołem. Pakiet Business & Software Engineering skierowany jest do deweloperów, architektów, projektantów systemów i analityków biznesowych.

Corporate Edition pozwala na tworzenie i przechowywanie współdzielonego repozytorium projektu z użyciem następujących baz danych: MS SQL Server, MySQL, Oracle9i, PostgreSQL, MSDE, Adaptive Server Anywhere i MS Access, zapewnia również zarządzanie kontrolą dostępu oraz kontrolą wersji, stanowi wsparcie dla technologii MDG (Model Driven Generation). Pakiet ten skierowany jest do dużych zespołów projektowych. Wersja Corporate Edition posiada pełną funkcjonalność poniższych pakietów, czyli wersji Desktop i Professional.

Professional Edition posiada możliwość definiowania wspólnego repozytorium bazodanowego dla projektów, podpięcie projektu do wybranego narzędzia umożliwiającego kontrolę wersji, opcję budowania kodu i bazy danych, wspiera technologię MDG oraz proces debugowania, posiada pełną funkcjonalność generowania dokumentacji w formatach HTML oraz RTF, a także śledzenia wymagań projektowych.

Desktop Edition jest najwcześniejszą wersją pakietu Enterprise Architect. Zapewnia tworzenie i modyfikację diagramów UML, generowanie dokumentacji w formatach HTML oraz RTF. Pakiet ten nie zapewnia generowania kodu oraz importu/eksportu kodu i schematów DDL, brak możliwości współdzielenia projektów między wieloma użytkownikami, brak wsparcia dla technologii MDA oraz symulacji procesów. Pakiet ten skierowany jest do samodzielnych analityków oraz deweloperów.

Poniższa tabela przedstawia porównanie najważniejszych cech ww. wersji Enterprise Architect.

Tabela 4: Cechy wersji Enterprise Architect

Cecha	Ultimate Edition	Systems Engineering	Business & Software Engineering	Corporate Edition	Professional Edition	Desktop Edition
Pliki projektowe .EAP	+	+	+	+	+	+
Zaawansowane modelowanie w UML 2.4	+	+	+	+	+	+
API do automatyzacji	+	+	+	+	+	+
Modelowanie procesów biznesowych	+	+	+	+	+	+
Edycja tabelaryczna w liście elementów	+	+	+	+	+	+
Zaawansowane przeszukiwanie repozytorium	+	+	+	+	+	+
Modelowanie map myśli	+	+	+	+	+	+
Rozszerzanie za pomocą profili	+	+	+	+	+	+
Macierz relacji (do śledzenia zależności)	+	+	+	+	+	+
Generowanie raportów HTML i RTF	+	+	+	+	+	+
Zarządzanie wymaganiami	+	+	+	+	+	+
Dostosowywanie wyglądu elementów	+	+	+	+	+	+

Cecha	Ultimate Edition	Systems Engineering	Business & Software Engineering	Corporate Edition	Professional Edition	Desktop Edition
(skrypty opisu kształtów)						
Edytor stanów	+	+	+	+	+	+
Zarządzanie testami	+	+	+	+	+	+
Integracja z systemami kontroli wersji	+	+	+	+	+	+
Import i eksport XMI w wersjach 2.1, 1.2, 1.1, 1.0	+	+	+	+	+	+
Pełna inżynieria kodu źródłowego	+	+	+	+	+	
Inżynieria kodu opisu struktury baz danych	+	+	+	+	+	
Debugowanie i wizualizacja aplikacji	+	+	+	+	+	
Technologie MDG (tworzenie i używanie)	+	+	+	+	+	
Architektura sterowana modelem – MDA (z dodatkami MDA)	+	+	+	+	+	
Projektowe forum dyskusyjne	+	+	+	+	+	
Replikacja projektów .EAP	+	+	+	+	+	
Dostosowywanie raportów: edytor szablonów dla raportów RTF	+	+	+	+	+	
Inżynieria wsteczna binariów Java i .NET (jar,il)	+	+	+	+	+	
Współdzielona praca na modelach	+	+	+	+	+	
Tworzenie WSDL	+	+	+	+	+	
Tworzenie XML Schema (XSD)	+	+	+	+	+	
Audyt zmian w modelu	+	+	+	+		

Cecha	Ultimate Edition	Systems Engineering	Business & Software Engineering	Corporate Edition	Professional Edition	Desktop Edition
Porównywanie i złączanie linii bazowych	+	+	+	+		
Repozytoria w bazach danych SQL	+	+	+	+		
Reguły bezpieczeństwa dla modelu (oparte o role)	+	+	+	+		
Dostępność edycji Floating	+	+	+	+		
Tworzenie skryptów działających na EA w językach JScript, VBScript i Javascript	+	+	+	+		
Optymalizacja działań w sieciach – WAN Optimizer	+	+	+	+		
Opóźnione wczytywanie projektu	+	+	+	+		
Generowanie kodu źródłowego z modeli zachowań	+	+	+			
Operacje matematyczne w skryptach	+	+	+			
Generowanie BPEL z diagramów BPMN 1.1	+		+			
Kompozytor reguł biznesowych	+		+			
Tworzenie kodu źródłowego w ADA 2005	+	+				
Wsparcie dla SysML 1.1	+	+				
Symulowanie modelu parametrycznego SysML	+	+				

Zaproponowana w niniejszym dokumencie metodyka zarządzania wymaganiami oparta została o wersję Professional – jest to wersja, która obecnie jest wykorzystywana przez pracowników GUGiK korzystających z ww. narzędzia.

Docelowo jednak rekomendowana jest zmiana wersji oprogramowania Enterprise Architect do jednej z wersji rozszerzonych. Dla dużych grup projektowych osadzonych w różnych siedzibach, jak ma to miejsce dla projektów tworzonych na potrzeby GUGiK, najlepszym rozwiązaniem byłaby wersja **Corporate**. Jest to wersja najpełniejsza, obsługująca zabezpieczenia w postaci modułu przydzielania zróżnicowanych uprawnień użytkownikom i grupom użytkowników, blokadę elementów używanych przez danego użytkownika w danym czasie.

Wersje Ultimate Edition, Systems Engineering, oraz Business & Software Engineering wydają się zbyt rozbudowane na potrzeby projektów dla GUGiK, ich głównym przeznaczeniem jest generacja kodu z zaprojektowanych modeli. **Business & Software Engineering Edition** posiada pełną funkcjonalność edycji Corporate Edition wzbogaconej o funkcjonalność rozwiązań biznesowych (Zawiera moduły integracji MGD Integration i MGD Link z programami MS Visual Studio, MS Eclipse, UPDM, Zachman Framework, TOGAF. Edycja ta zawiera narzędzia do modelowania i planowania strategii biznesowych takich jak Value Chain, Balanced Scorecard, Strategy Map, Decision Tree, czy Organization Chart. **Systems Engineering Edition** to także pełna funkcjonalność edycji Corporate Edition poszerzonej dodatkowo o wbudowane integracje z SysML, MS Visual Studio, Eclipse, DDS, DODAF/MODAF, a także wsparcie dla ADA 2005. Edycja ta przeznaczona jest dla inżynierów, programistów korzystających z bogatej gamy dodatkowej funkcjonalności (SysML, DDS, Visual Studio). **Ultimate Edition** łączący funkcjonalność dwóch powyższych edycji, czyli Business & Software Engineering Edition i Systems Engineering Edition, jest to najbardziej zaawansowana edycja Enterprise Architect. Obejmuje pełną funkcjonalność wszystkich pakietów Enterprise Architect. Licencja ta jest przeznaczona dla zaawansowanych użytkowników, którzy wykorzystują również dodatkowe możliwości integracji (Zachman, Eclipse, TOGAF, DDS, itp).

Dla edycji: Corporate, Business & Software Engineering, Systems Engineering, Ultimate dostępne są licencje “Floating”, pozwala to na korzystanie z jedna licencja dla określonej liczby osób w ramach sieci lokalnej.

3 Metodyka zarządzania wymaganiami

Wymaganie to pewna własność produktu końcowego - system informatycznego, którą system ten musi posiadać, aby spełnić oczekiwania zamawiającego. Własność ta może być związana z określonym sposobem funkcjonowania systemu (**wymaganie funkcjonalne**) lub cechami jakościowymi narzuconymi na system (**wymaganie pozafunkcjonalne**). System informatyczny spełnia wymaganie, jeśli zostanie potwierdzone że posiada wszystkie własności, które są określone wymaganiami oraz spełnia cechy jakościowe.

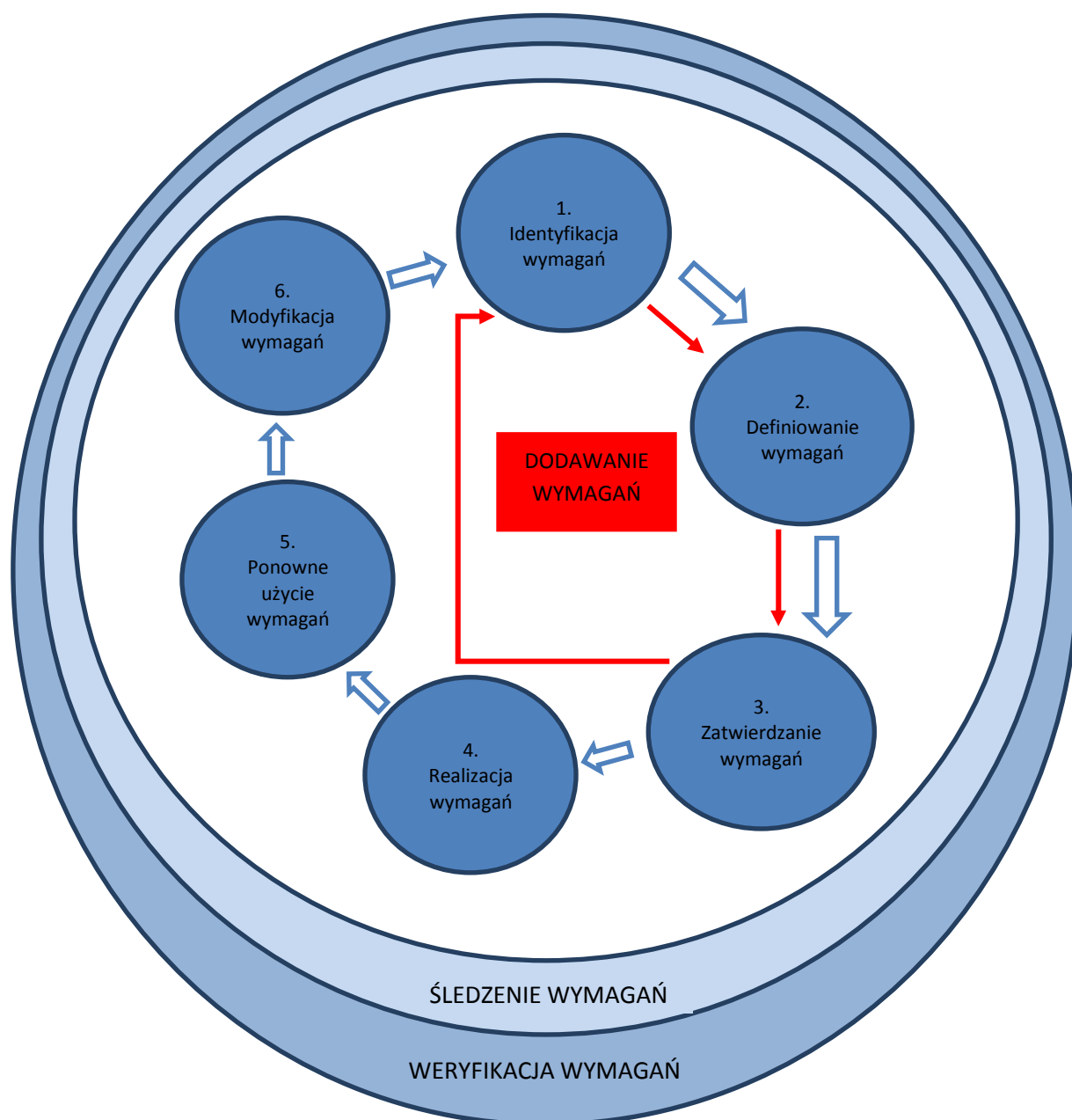
Wymagania stanowią odpowiedź na pytanie jaki system należy zbudować oraz w jaki sposób powinien on działać. Prawidłowe zarządzanie wymaganiami steruje budową systemu, a także jest narzędziem komunikacji pomiędzy Zamawiającym a Wykonawcą systemu.

Metodyka zarządzania wymaganiami opiera się o zdefiniowane fazy wymagań, które składają się na **cykl życia wymagań**. W zależności od tego, w jakiej fazie znajduje się poszczególne wymaganie podlega ono odpowiednim czynnościom zarządczym. Dla każdego z wymagań zdefiniowane zostały **stany**, które może ono przyjmować podczas cyklu życia.

Istotnym zagadnieniem związanym z zarządzaniem wymaganiami jest **śledzenie wymagań**, które ma na celu kontrolowanie powiązań pomiędzy wymaganiami, w tym w szczególności pomiędzy wymaganiami realizowanymi w ramach różnych projektów.

Wymagania powinny podlegać cyklicznej **weryfikacji**, czyli przeglądowi zapewniającym wymaganiom jakość i zgodność z założeniami.

3.1 Cykl życia wymagań



Rysunek 1: Podstawowy cykl życia wymagania

Na potrzeby niniejszej metodyki zdefiniowany został cykl życia wymagań, na który składają się następujące fazy wymagań:

- Identyfikacja wymagań,
- Definiowanie wymagań,
- Zatwierdzanie wymagań,
- Realizacja wymagań,
- Ponowne użycie wymagań,
- Modyfikacja wymagań.

Cykl życia wymagania rozpoczyna się od pierwszej fazy którą jest identyfikacja wymagań, jeśli wymaganie zostanie zidentyfikowane to drugą fazą jest definiowanie wymagań, po pomyślnym zdefiniowaniu wymagań, proces ten wchodzi w fazę zatwierdzenia wymagań, gdzie po pozytywnym zatwierdzeniu przechodzi już do implementacji tego wymagania, następną fazą jest ponowne użycie wymagania, dalej jest faza modyfikacji wymagań po której wymagania mogą być dalej identyfikowane. W trakcie całego tego cyklu mogą być prowadzona doraźna weryfikacja wymagań, a także śledzenie wymagań. Pierwsze trzy fazy wymagań, czyli identyfikacja wymagań, definiowanie wymagań oraz zatwierdzenie wymagań tworzą podcykl dodawania wymagań w niniejszym cyklu życia wymagań.

3.1.1 Identyfikacja wymagań

Na identyfikację wymagań składają się kroki niezbędne do poprawnego i prawidłowego sformułowania wymagania, które odpowiadać będzie rzeczywistej potrzebie lub stanowić będzie rozwiązanie problemu który napotkał zespół realizujący projekt. W tym celu wskazane jest przejście następującej ścieżki pozwalającej na poprawne zidentyfikowanie wymagań:

1. Określenie potrzeb i problemów

Pierwszym krokiem niezbędnym do poprawnego zidentyfikowania wymagań jest określenie potrzeb i problemów. Potrzeby i problemy mogą być definiowane na bardzo ogólnym poziomie szczegółowości i stanowić mają punkt wyjścia do szczegółowego identyfikowania wymagań.

Potencjalnym źródłem pozyskania informacji o problemach i potrzebach, pozwalającymi na kompleksowe późniejsze zidentyfikowanie wymagań są:

- Zapisy dotyczące problemów i celów zawarte w studiach wykonalności (w szczególności w rozdziałach IV.2 i V.1),
- Zidentyfikowane problemy i potrzeby interesariuszy projektu,
- Wizja systemu,
- Akty prawne i organizacyjne,
- Standardy i pryncypia architektoniczne,
- Architektury biznesowe opracowywane dla poszczególnych projektów.

Rozpoczęcie analizy wymagań od identyfikacji potrzeb i problemów, a także powiązanie konkretnych wymagań z potrzebami i problemami ułatwia późniejsze zarządzanie wymaganiami, gdyż pozwala na łatwe określenie pochodzenia danego wymagania. Łatwość identyfikacji pochodzenia danego wymagania jest istotne w sytuacji, gdy planowana jest modyfikacja wymagań i istnieje ryzyko rezygnacji lub zmiany treści wymagań, które wynikają z wymogów zewnętrznych (np. zewnętrzne akty prawne, zaakceptowane studia wykonalności itp.). Gdy potrzeba zostanie już zidentyfikowana to bibliotekarz projektu zamieszcza ją w bazie wymagań.

2. Zbudowanie przypadków użycia

Przypadki użycia opisują zachowanie systemu z punktu widzenia zewnętrznego obserwatora. Zastosowanie przypadków użycia pozwala na określenie granic systemu, zdefiniowanie użytkowników i innych systemów współpracujących z systemem, a także zdefiniowanie najważniejszych funkcjonalności realizowanych przez system.

Potencjalnym źródłem informacji dla budowy przypadków użycia powinny być zdefiniowane w kroku 1 potrzeby i problemy, a także generyczne przypadki użycia z pozostałych projektów (m.in. poprzez ich ponowne użycie). Na model przypadków użycia składają się diagramy prezentujące powiązania pomiędzy aktorami i przypadkami użycia, a także opisy poszczególnych przypadków użycia.

Budowa modelu przypadków użycia powinna przebiegać następująco:

- identyfikacja aktorów,
- identyfikacja przypadków użycia,
- identyfikacja powiązań i związków między przypadkami użycia,
- naniesienie aktorów, przypadków użycia i powiązań między nimi na diagramy przypadków użycia,
- opisanie przypadków użycia.

Szczegółowe czynności związane z budową diagramów przedstawia Tabela 5.

Tabela 5: Procedura tworzenia modelu przypadków użycia

L.p.	Krok	Odpowiada na pytanie
1	Identyfikacja aktorów	Kto komunikuje się z systemem? Kto będzie korzystał z funkcji systemu? Kto będzie system pielęgnował? Jakie urządzenia system obsługuje ? Z jakimi innymi systemami system się komunikuje ? Kto lub co jest zainteresowane wynikami pracy systemu ?
2	Identyfikacja przypadków użycia	Jakiej funkcji aktor wymaga od systemu? Czy aktor ma być powiadamiany o zdarzeniach w systemie? Jakie są wejścia i wyjścia systemu? Czy można powiązać w jeden przypadek użycia zespół funkcji wspólnie realizujących ten sam cel? Czy istnieją pewne działania i czynności realizowane w ramach wielu przypadków użycia? Czy przypadki użycia mogą podlegać rozszerzeniom?
3	Identyfikacja powiązań	Jakie mogą wystąpić powiązania pomiędzy aktorami ? Jakie mogą wystąpić powiązania pomiędzy przypadkami użycia ? Jakie mogą wystąpić powiązania pomiędzy aktorami i przypadkami użycia ?
4	Naniesienie aktorów, przypadków użycia i powiązań między nimi na diagram przypadków użycia	
5.	Opisanie przypadków użycia	Jaki jest cel przypadku użycia? Jakie warunki początkowe muszą zostać spełnione aby przypadek mógł zostać wykonany? Jaki jest oczekiwany rezultat? Jakie są przewidywane alternatywne rezultaty (błędy, wyjątki itp.)? Jaki jest scenariusz główny przypadku użycia?

	Jakie są scenariusze alternatywne przypadku użycia?
--	---

Ze względu na wykorzystanie modelu przypadków użycia, a w szczególności scenariuszy do szacowania rozmiaru oprogramowania (Patrz dokument: Metoda szacowania rozmiaru oprogramowania) niezbędne jest staranne opisanie przypadków użycia zgodnie z następującymi zasadami:

- opis przypadku użycia musi zawierać jego definicję oraz opis celu,
- należy wskazać oczekiwany rezultat przypadku użycia, wskazujący rezultat osiągnięty w wyniku wykonania scenariusza głównego przypadku użycia, a także rezultaty alternatywne, które mogą zostać osiągnięte podczas przebiegów scenariuszy alternatywnych,
- scenariusze przedstawiają poszczególne kroki wykonywane przez system lub aktora wykonywane w ramach realizacji przypadku użycia. Scenariusze powinny być pisane przy użyciu notacji SVDPI (Subject – Verb – Direct-Object – Preposition – Indirect-Object), czyli:
 - o Podmiot – wskazujący na nazwę aktora (lub słowo „system”) wykonującego krok w przypadku użycia,
 - o Orzeczenie – nazwa czynności, która jest wykonywana przez podmiot,
 - o dopełnienie – nazwa obiektu, który podlega czynności,
 - o Przyimek i dopełnienie dalsze – przyimek „na”, „w” itp. oraz opcjonalna nazwa innego obiektu.

Na potrzeby późniejszego szacowania rozmiaru wskazane jest zachowanie odpowiedniego poziomu szczegółowości w opisywaniu przypadku użycia, a także oparcie przygotowywania opisów scenariuszy o model dziedziny oraz model danych (w szczególności w zakresie używanych pojęć słownikowych oraz nazw grup danych, których przesunięcia będą później podlegały identyfikacji w ramach szacowania). Właściwy poziom szczegółowości wskazuje jednoznacznie na czynności wykonywane przez użytkownika i system, uwzględniające kontekst przesunięć poszczególnych grup danych.

Podczas opisywania przypadków użycia należy również zweryfikować poprawność wykonanej wcześniej w ramach identyfikowania przypadków użycia dekompozycji funkcjonalnej systemu na przypadki użycia. Podczas definiowania scenariuszy należy dążyć do niepowielania wykonywania takich samych sekwencji kroków w wielu przypadkach użycia, zamiast tego odwołując się do innego przypadku użycia w ramach którego ta sekwencja kroków jest wykonywana i wskazując relację pomiędzy tymi przypadkami (relacje typu include lub extend). W szczególności, po opisanie przebiegów scenariuszy przypadków użycia należy zweryfikować poprawność zidentyfikowanych wcześniej powiązań pomiędzy przypadkami użycia.

Model przypadków użycia należy przenieść do bazy wymagań, osobą za to odpowiedzialną jest bibliotekarz projektu.

3. Zidentyfikowanie wymagań funkcjonalnych i pozafunkcjonalnych systemu

Kolejnym krokiem jest właściwe identyfikowanie wymagań. W pierwszej kolejności identyfikowane są wymagania funkcjonalne (zestaw funkcji, które ma wykonywać system oraz wymagania związane z bezpieczeństwem), a następnie wymagania pozafunkcjonalne, związane z:

- użytecznością, ergonomią i wizualnymi aspektami oprogramowania,
- niezawodnością (np. częstością występowania błędów),
- wydajnością systemu,
- łatwością użytkowania, testowania i instalowania na różnych środowiskach.¹

Na podstawie zbudowanego modelu przypadków użycia należy przystąpić do identyfikowania wymagań funkcjonalnych systemu. Wymagania funkcjonalne opisują funkcjonalności, które system ma realizować.

Poprawnie określone wymaganie powinno odnosić się tylko do jednej kwestii, powinno być kompletne, sformułowane precyzyjnie i jednoznacznie, nie może być sprzeczne z innymi wymaganiami. Wymaganie spełnia część potrzeb biznesowych interesariuszy lub w całości. Wymaganie musi być realne do zrealizowania w ramach ograniczeń projektu, a także możliwe do przetestowania. Bibliotekarz projektu powinien wprowadzić do bazy wymagań opracowane przez zespół projektowy potrzeby oraz przypadki użycia, a status wymagania ustawić na „propozycja” oraz fazę projektu na „1.0”.

3.1.2 Definiowanie wymagań

W fazie Definiowania wymagań wymagania, które zostały wcześniej zidentyfikowane podlegają szczegółowemu definiowaniu – uzupełnianiu ich opisu o wymagane elementy.

Procedura definiowania wymagań składa się z poniższych kroków:

1. Sformułowanie wymagania i jego opisu

Wymaganie w tym punkcie powinno zostać jednoznacznie zdefiniowane, opis ten powinien być zwięzły, konkretny i jednoznacznie identyfikować wymaganie.

2. Wprowadzenie wymagania do repozytorium przez bibliotekarza projektu

Kolejnym etapem jest wprowadzenie zdefiniowanej nazwy i opisu wymagania do bazy wymagań, tą czynność wykonuje bibliotekarz projektu poprzez dodanie odpowiedniego elementu i uzupełnieniu go tekstem. W tym etapie faza wymagania otrzymuje numer „2.0”.

3. Wprowadzenie wartości atrybutów wymagania do repozytorium

Opis wymagania jest uzupełniany przez bibliotekarza projektu o takie dane jak: status wymagania, stopień powinności, wersje oraz o wartości pozostałych pól, które są istotne dla danego projektu.

Wymagania posiadają poniższe atrybuty:

- Obowiązkowe:
 - o Identyfikator wymagania

¹ Na podstawie klasyfikacji wymagań FURPS (Functionality, Usability, Reliability, Performance, Supportability)

- ☐ Nazwa wymagania
 - ☐ Treść wymagania
 - ☐ Status wymagania
 - ☐ Stopień powinności
 - ☐ Wersja
- Opcjonalne:
 - ☐ Powiązanie z usługą biznesową, aplikacyjną, danych lub techn.
 - ☐ Klasyfikację kontraktu architektonicznego (OPZ)
 - ☐ Priorytet
 - ☐ Trudność
 - ☐ Opis
 - ☐ Źródło
 - ☐ Kryterium spełnienia
 - ☐ Zależności
 - ☐ Data planowanej realizacji wymagania

Identyfikator wymagania (np. *G.NF.032*) zapisany jest za pomocą liter i cyfr, pozwalający na powiązanie funkcjonalności z innymi obiektami, unikalny w ramach całego modelu wymagań. Identyfikator zapisany jest w postaci Z.Y.XXX, gdzie:

- Z – oznacza identyfikator systemu (odpowiednio G2- System Geoportal, SDI- Moduł SDI, UMM – Uniwersalny Moduł Mapowy, HARM – narzędzia do harmonizacji, KSZBDOT – Krajowy System Zarządzania BDOT, SZNMT – System Zarządzania NMT, KGESUT – System K-GESUT, PRG – System Zarządzania PRG, EMUiA – Aplikacja EMUiA).,
- Y – oznaczenie rodzaju wymagania (funkcjonalne / pozafunkcjonalne). Literą „F” powinny być oznaczane wymagania funkcjonalne, a literami „NF” niefunkcjonalne wymagania.
- XXX – trzycyfrowy kolejny numer wymagania w ramach projektu.

Nazwa wymagania (np. *Przeglądanie rejestru zdarzeń systemowych*) powinna być jasna, precyzyjna i zwięzła odnosząca się do opisu wymagania.

Treść wymagania jest jego słownym opisem wyczerpująca jego kwestię. Treść wymagania powinna pochodzić z fazy Identyfikacja wymagań i spełniać wymagania przedstawione w Rozdziale 3.1.1. Przykładowa treść wymagania: *Korzystanie z funkcji systemu i baz danych MUSI być możliwe tylko po zalogowaniu użytkownika do systemu i zależeć od przyznanych temu użytkownikowi praw.*

Status wymagania określa jego stan. Dopuszczalne statusy to: propozycja, zatwierdzony, zweryfikowany, zaimplementowany, odrzucony oraz wstrzymany. Szczegółowe wyjaśnienia odnośnie statusów wymagań znajdują się w rozdziale 3.2.

Stopień powinności wymagania rozumiany jest zgodnie z definicją jego angielskich odpowiedników określonych w RFC2119 "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, S. Bradner, March 1997:

- „MUSI” oznacza bezwzględny wymóg implementacji wymagania w rozwiązaniu przez Wykonawcę,
- „ZABRANIA SIĘ” oznacza bezwzględny zakaz implementacji wymagania w rozwiązaniu przez Wykonawcę,
- „POWINEN” należy rozumieć w taki sposób, że niespełnienie warunku opatrzonego tą klauzulą jest dopuszczalne tylko w szczególnie uzasadnionych przypadkach po zgodzie Zamawiającego,
- „NIEZALECANE” który oznacza iż mogą istnieć różne powody, których określone postępowanie jest dopuszczalne lub nawet pożyteczne, ale wszystkie konsekwencje tego wymagania powinny być zrozumiałe i dokładnie rozważone przed realizacją tak opisanego wymagania z tą etykietą. W szczególnie uzasadnionych przypadkach Zamawiający może wydać zgodę na implementację wymagania w rozwiązaniu przez Wykonawcę,
- „MOŻE” oznacza wymaganie opcjonalne. Wykonawca może potraktować to wymaganie jako opcjonalne, które należy rozważyć, czy w danym przypadku ma zastosowanie.

Stopnie powinności należy w treści wymagania wyróżniać wielkimi literami.

Wersja wymagania jest zapisem cyfrowym, pozwala na określenie poziomu ewolucji jakie dane wymaganie przeszło. Wersja wymagania zapisywana jest w postaci N.N – gdzie N oznacza liczbę naturalną lub zero (np. 0.9 lub 1.2). Każda zmiana wymagania musi wiązać się z podniesieniem wersji wymagania, przy czym zmiany kosmetyczne powinny wiązać się z podniesieniem części dziesiątej wersji, natomiast duże zmiany powinny powodować podniesienie wersji to kolejnej wersji N.0 (np. podniesienie z wersji 1.4 do wersji 2.0).

Powiązanie z usługą biznesową, aplikacyjną, danych lub technologiczną w ramach projektu lub budowanego systemu, w szczególności wymaganie funkcjonalne MUSI być powiązane z usługą aplikacyjną, która wymagania to realizuje.

Trudność oznacza poziom skomplikowania i trudności w implementacji wymagania przez Wykonawcę. Trudność przyjmuje skalę: wysoka, średnia, niska.

Opis jest dodatkowym polem zawierającym informacje uzupełniające, nie ujęte w poprzednich polach. W szczególności w polu opis powinny zostać zawarte informacje takie jak odstępstwa od wcześniej przyjętych standardów.

Źródło określa dokument lub inne źródło, z którego wynika potrzeba wymagania (np. akt prawny, studium wykonalności, potrzeba interesariusza).

Kryterium spełnienia wymagania jest dodatkowym opcjonalnym polem zawierającym informacje o przestanie, która musi zostać spełniona przez wymagania aby mogło ono być ono uznane za

poprawnie zaimplementowane. Kryterium spełnienia wymagania będzie przydatne do późniejszego tworzenia scenariuszy testowych dla systemu.

Podczas definiowania wymagań istotne jest również określenie **zależności pomiędzy wymaganiami**. Dla wymagań istnieć mogą następujące typy zależności wymagań:

- pojedyncze wymaganie (lub grupa wymagań) zależne od innego pojedynczego wymagania,
- pojedyncze wymaganie (lub grupa wymagań) zależne od modułu / obszaru funkcjonalnego,
- pojedyncze wymaganie (lub grupa wymagań) zależne od całego systemu.

Zależności pomiędzy wymaganiami jest dodawana w sytuacji, gdy wymaganie korzysta z:

- innych usług,
- baz danych,
- innych systemów,
- danych.

Zależności pomiędzy wymaganiami powinny być wprowadzane poprzez pokazanie relacji pomiędzy obiektami prezentującymi wymagania. W opisie relacji należy wprowadzić nazwę tejże relacji zależności.

Data planowanej realizacji wymagania jest datą na jaką wykonawca ustalił z Zamawiającym zaimplementowanie wymagania. Data planowanej realizacji wymagania może okazać się że nie jest rzeczywistą datą faktycznego zaimplementowania tego wymagania.

3.1.3 Zatwierdzenie wymagań

Każde zidentyfikowane wymaganie przed przejściem do kolejnej fazy zarządzania wymaganiami musi być zatwierdzone.

Do zatwierdzenia wymagania jest konieczne oszacowanie pracochłonności i wpływu na system. Jeśli zatwierdzone jest modyfikowane wymaganie należy wycenić zmianę jaka jest dokonywana. Przeprowadzana zmiana musi być zgodną z procedurą zarządzania zmianą.

Zatwierdzanie wymagań w przypadku grup wymagań związanych z nowym systemem lub modułem może być realizowane łącznie dla całej grupy wymagań.

Wymagania:

- odbiegające od standardów, a także wymagania obejmujące więcej niż jeden system informatyczny (np. wymagania dotyczące integracji systemów) zatwierdza Rada Architektury
- pozostałe zatwierdza kierownik projektu po stronie GUGiK.

Procedura zatwierdzania wymagań składa się z następujących etapów:

1. Oszacowanie wpływu na system

Oszacowanie wpływu odbywa się poprzez zidentyfikowanie wpływu wymagań na istniejące wymagania systemu oraz na inne wymagania. Służą do tego mechanizmy śledzenia wymagań pozwalające na analizę wpływu. Zidentyfikowany wpływ podlega kategoryzacji jego zakresu.

Kategoryzowanie wpływu:

- brak wpływu lub niewielki wpływ – ocena dla wymagania nie narzuca konieczności przeprowadzania modyfikacji istniejących wymagań bądź wymaga jedynie zmian kosmetycznych, a osobą odpowiedzialną za zatwierdzenie jest Kierownik Projektu, z tej fazy nie ma obowiązku przejścia do podprocesu dodawania wymagań
- średni wpływ – ocena dla wymagania powiązanego z innymi wymaganiami, w tym przypadku obowiązkowe jest przeprowadzanie procedury modyfikacji dla wymagań powiązanych, w przypadku gdy powiązane wymagania są w ramach jednego systemu zatwierdza je Kierownik Projektu, gdy dotyczą wielu systemów to zatwierdza Rada Architektury
- duży wpływ – ocena dla wymagania, które skutkuje koniecznością zmiany krytycznych funkcjonalności kluczowych systemów lub koniecznością zmian w wielu innych systemach, wówczas obowiązkowo należy przejść do procesu modyfikacji wymagań, dla tego wpływu odpowiedzialność za zatwierdzenie ponosi Rada Architektury

2. Wycena implementacji wymagań

Wymagania powinny być objęte wyceną ich implementacji, dla wymagań obejmujących cały system bądź moduł to powinna być wycena całościowa rozwiązania, natomiast dla pojedynczych wymagań szacowany jest nakład na tą pojedynczą realizację. Dla wszystkich wymagań bezwzględnie koszt powinien być powiększony o koszty modyfikacji innych wymagań i rozwiązań, jeśli realizacja danego wymagania wiąże się z tym. Wymaganie powinno także zostać wycenione pod względem jego czasu realizacji. Wycena czasowa realizacji zmienianych wymagań powinna zostać potwierdzona przez jego Wykonawcę.

3. Zatwierdzenie wymagania

Po oszacowaniu wpływu i wycenie wymagania podejmowana jest decyzja o zatwierdzeniu wymagania bądź odrzuceniu czy wstrzymaniu. Dla wymagań z określonym niewielkim wpływem osobą zatwierdzającą jest Kierownik Projektu, dla pozostałych Rada Architektury.

4. Zmiana statusu

Przy pomyślnym zaakceptowaniu bibliotekarz projektu zmienia status wymagania na zatwierdzony. W przypadku gdy wymaganie jest odrzucone to status zmieniany jest na odrzucony, a jeśli realizacja wymagania jest wstrzymana to status zmieniany jest na wartość wstrzymane. Wraz ze zmianą statusu wprowadza do bazy wymagań ustaloną dla danego wymagania bądź grupy wymagań datę planowanej realizacji wymagania oraz zmienia fazę wymagania na numer „3.0”.

Zatwierdzenie wymagań ma na celu zrozumienie potrzeby i sprawdzenia zgodności z rzeczywistymi pragnieniami oraz intencjami interesariuszy.

Osoby uczestniczące w procesie zatwierdzania wymagań:

- bibliotekarz projektu
- Kierownik Projektu
- Rada Architektury
- właściciel wymagania

- administrator wymagań
- zespół ds. monitorowania wymagań

3.1.4 Realizacja wymagania

Warunkiem koniecznym do rozpoczęcia realizacji wymagania jest zakończenie procesu współpracy z wykonawcą odnośnie wprowadzenia wymagań (Rysunek 17) znajdujący się w rozdz. „5.3 Etapy wdrażania metodyki”.

Procedura realizacji wymagania przedstawiona została poniżej w punktach:

1. Przekazanie wykonawcom wymagań do implementacji

Zamawiający przekazuje Wykonawcy wymagania, które są wycinkiem bazy wymagań zawierające jedynie wymagania zatwierdzone do realizacji. Bibliotekarz projektu zmienia status wymagania na „w trakcie implementacji”.

2. Implementacja wymagań przez wykonawców

W tym etapie Wykonawca realizuje wymaganie przekładając je na działającą funkcjonalność systemu. Zaimplementowane wymaganie musi spełniać wszelkie założenia i ograniczenia zdefiniowane przez Zamawiającego. W przypadku wymagań modyfikowanych Wykonawca wprowadza zmiany w danej zaimplementowanej funkcjonalności systemu.

3. Zmiana statusu wymagań

Po zaimplementowaniu wymagania bądź grupy wymagań bibliotekarz projektu zmienia status wymagania na „zaimplementowane” oraz zmienia numer fazy wymagania.

Wraz ze zmianą stanu wymagania, w trakcie jego realizacji (implementacji w systemie) należy zmieniać jego status aby odpowiadał rzeczywistemu stanowi. Każdy status wymagania ma przypisany odrębny kolor, dostępne statusy:

- propozycja/zidentyfikowany
- zatwierdzony
- zweryfikowany
- w trakcie implementacji
- zaimplementowany
- odrzucony
- wstrzymany

Istotne jest utrzymanie aktualnego statusu za co odpowiedzialny jest bibliotekarz. Aktualny status pozwala na śledzenie realizacji wymagania na wszystkich jego etapach.

Przekazywanie wymagań wykonawcom odbywa się poprzez wykonywanie replik bazy wymagań.

3.1.5 Ponowne użycie wymagań

Zgodnie z założeniami przyjętymi w niniejszej metodyce w miarę możliwości wymagania powinny być ponownie używane. Ponowne użycie wymagań wskazane jest m.in. w następujących przypadkach:

- wymagania określające sposób wykorzystania usług infrastrukturalnych SIG,

- wymagania określające sposób wykorzystania usług udostępnianych przez poszczególne systemy,
- wymagania związane ze sposobem implementacji standardów SIG.

Ponowne użycie wymagań oznacza zastosowanie w danym projekcie wymagań, które zostały opracowane w innym projekcie.

3.1.5.1 Przygotowywanie wymagań do ponownego użycia

Wymagania, które zostały już zaimplementowane mogą być ponownie używane w kolejnych projektach. Identyfikacja możliwości ponownego użycia wymagań możliwa jest w dwóch sytuacjach:

- przegląd wymagań przez zespół ds. monitorowania wymagań po zakończeniu implementacji wymagań i wskazanie wymagań jako wymagania do ponownego użycia,
- przegląd wymagań przez zespoły projektowe innych systemów i zidentyfikowanie ich jako wymagania możliwe do ponownego użycia w danym systemie.

3.1.5.2 Ponowne używanie wymagań

W celu ponownego użycia wymagań niezbędne jest wykonanie następujących kroków:

1. Zidentyfikowanie przez zespół projektowy wymagań, które mogłyby zostać ponownie użyte
2. Uzyskanie od właściciela wymagań zgody na ponowne użycie wymagań
3. Uzgodnienie z właścicielem wymagań zasad wykorzystania wymagań, w tym ewentualnych uszczegółowień i modyfikacji

3.1.6 Modyfikacja wymagań

W ramach realizacji prac przez Wykonawcę:

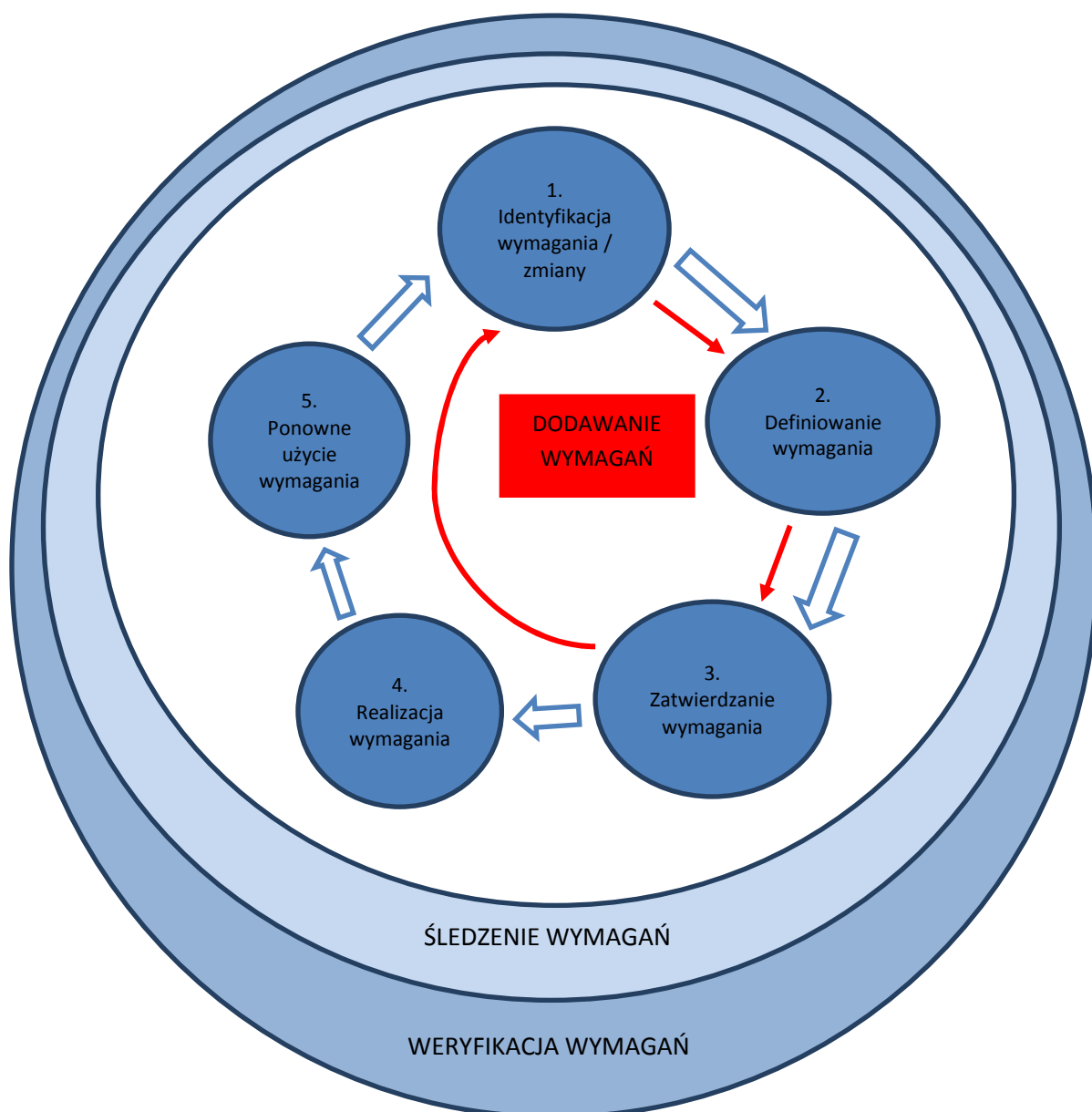
- wymagania zawarte w zamówieniu będą ulegały uszczegóławianiu,
- będą pojawiały się sytuacje wymagające wprowadzenia zmian.

W celu zachowania spójności i aktualności modelu wymagań niezbędne będzie (po uzyskaniu akceptacji uprawnionej osoby) zaktualizowanie modelu wymagań, oraz przypisanie numeru obecnej wersji wymagania. Numer wersji wymagania pozwala na weryfikację zmian przez pozostałych użytkowników. Historia zmian wymagań powinna być przechowywana. Faza wymagania otrzymuje numer „6.0”.

Sytuacje w których będą pojawiały się potencjalne zmiany wynikać będą z:

- Identyfikacji żądań zmian
- Zapisania nowych priorytetów
- Identyfikacji i zarządzania konfliktami

Powyższe sytuacje objęte są procedurą zmian. Modyfikacja wymagania przechodzi skrócony cykl



Rysunek 2: Skrócona ścieżka cyklu życia wymagania

życia wymagania zaprezentowany poniżej.

Procedura modyfikacji wymagań składa się z następujących punktów:

1. Identyfikacja wymagania bądź identyfikacja zmiany wymagania

Polega na zdiagnozowaniu powodu modyfikacji istniejącego wymagania. Zidentyfikowana potrzeba lub problem zostaje wpisany do repozytorium wymagań przez bibliotekarza projektu. Bibliotekarz projektu również przebudowuje model przypadków użycia jeśli uzasadnione jest to poziomem zmiany wymagania.

2. Zdefiniowanie wymagania bądź zmiany treści wymagania

Po identyfikacji powodów zmiany należy sformułować definicję wymagania, wprowadzić ją do repozytorium wymagań, a także zmienić jego status, co powinien uczynić bibliotekarz projektu.

3. Zatwierdzenie wymagania bądź odrzucenie lub wstrzymanie realizacji

Należy ocenić wpływ zmiany wymagania jaki może mieć na system bądź moduł oraz na swoje otoczenie. Następnie szacowany jest koszt wprowadzenia zmian w wymaganiu oraz koszt jego nowej implementacji bądź aktualizacji istniejących systemów. Po zatwierdzeniu bibliotekarz projektu zmienia jego status. Jeśli właścicielem modyfikowanego wymagania jest inna osoba niż zgłaszająca potrzebę zmiany należy uzyskać pozwolenie właściciela wymagania na przebudowę wymagania. W szczególnych przypadkach gdy zmiana dotyczy się krytycznych modułów to należy uzyskać również zgodę Rady Architektury.

4. Realizacja wymagań zatwierdzonych

Po zatwierdzeniu modyfikacji wymagania należy przejść do wprowadzania zmian w zaimplementowanym systemie bądź do realizacji nowej implementacji, która również kończy się zmianą statusu wymagania. Zamawiający przekazuje wykonawcy zakres zmian wraz z wycinkiem bazy wymagań gdzie zostały wprowadzone zmiany.

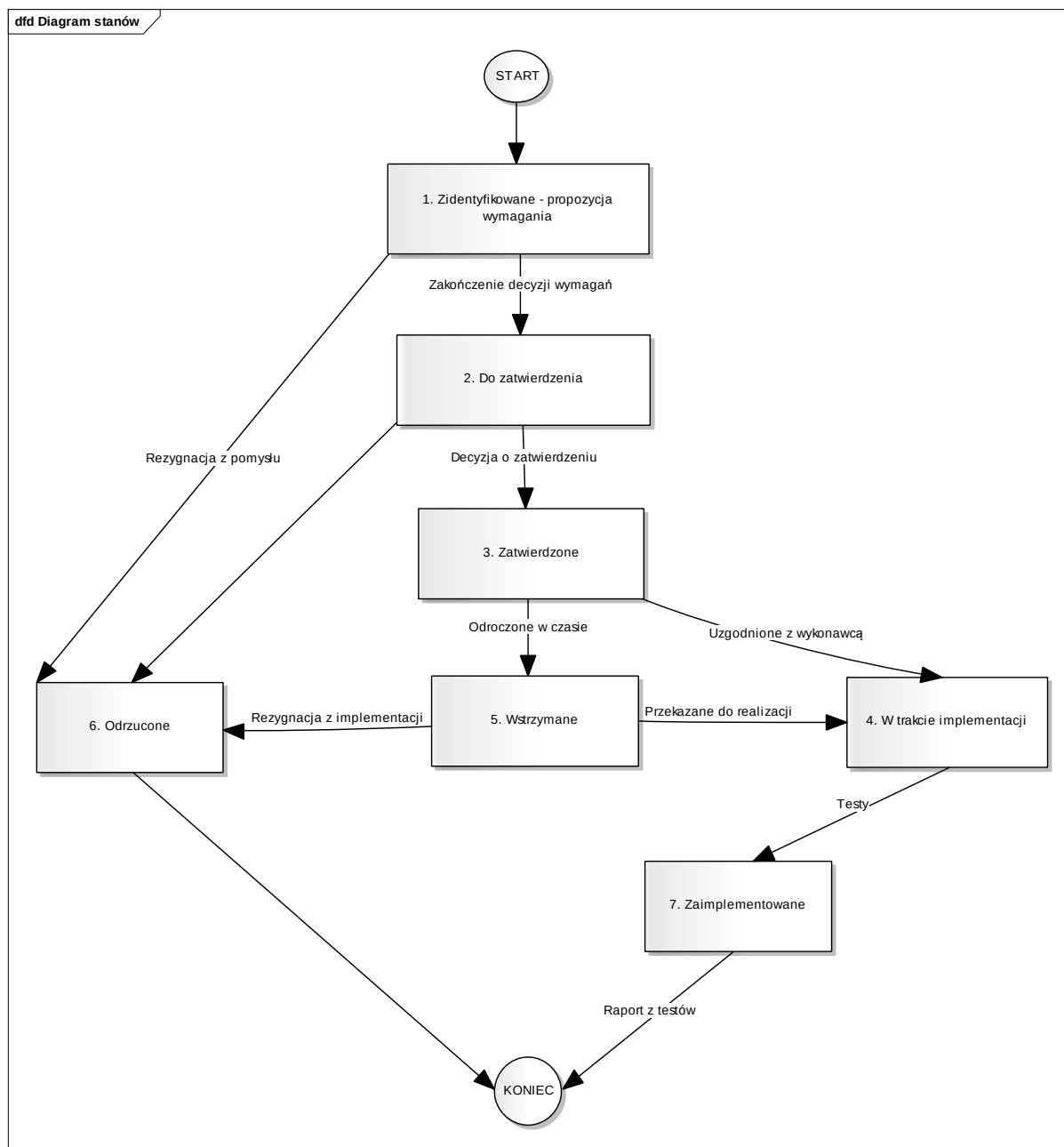
5. Sklasyfikowanie rozwiązania wymagania jako wymaganie możliwe do ponownego wykorzystania

W celu sklasyfikowania wymagania powinien odbyć się przegląd wymagań dokonany przez zespół ds. monitorowania wymagań po zakończeniu implementacji wymagań.

Przy wprowadzaniu modyfikacji wymagania istotne jest podanie przyczyny zmiany oraz datę zmiany.

3.2 Statusy wymagań

Statusy wymagania mogą zmieniać się adekwatnie do przechodzenia przez poniższą ścieżkę.



Rysunek 3: Diagram stanów wymagań

Stan wymagania określa położenie w jakim może znaleźć się wymaganie i prezentowany jest przez otrzymany status. Wymaganie może przyjmować poniższe stany:

- zidentyfikowane,
- do zatwierdzenia,
- zatwierdzone
- odrzucone,
- wstrzymane,
- w trakcie implementacji,
- zaimplementowane.

Przejścia pomiędzy stanami jest procesem przebiegającym sekwencyjnie, którego pierwszym krokiem jest zidentyfikowanie wymagania. Proces kończy się wraz z zaimplementowaniem wymagania bądź odrzuceniem.

Tabela 6: Przejścia pomiędzy stanami wymagań

Lp.	Status	Następny stan	Przejście
1	Start	Zidentyfikowane	Zaobserwowanie rzeczywistej potrzeby lub rozwiązania problemu który napotkał zespół realizujący projekt.
2	Zidentyfikowane	Do zatwierdzenia	Wymaganie zmienia swój stan gdy zostanie zidentyfikowana potrzeba bądź problem z którego zostanie zdefiniowane wymaganie.
3	Zidentyfikowane	Odrzucone	Wymaganie zmienia swój stan gdy zespół ds. monitorowania wymagań zrezygnuje z pomysłu.
4	Do zatwierdzenia	Zatwierdzone	Wymaganie zmienia swój stan gdy pomyślnie zostanie zaakceptowane przez zespół ds. monitorowania wymagań, składający się z osób odpowiednio umocowanych w hierarchii projektu i posiadających wiedzę o projekcie w zakresie zarządczym i merytorycznym.
5	Do zatwierdzenia	Odrzucone	Wymaganie zmieni swój stan gdy zespół ds. monitorowania wymagań zrezygnuje z jego implementacji w momencie gdy jest on zaklasyfikowany do weryfikacji.
6	Zatwierdzone	W trakcie implementacji	Wymaganie zostało ocenione pozytywnie. Zatwierdzenie wymagań odbywa się przez zespół ds. monitorowania wymagań lub upoważnione osoby po stronie zamawiającego
7	Zatwierdzone	Wstrzymane	Wymaganie zostało zatwierdzone do realizacji ale odroczone w czasie.
8	Zatwierdzone	W trakcie implementacji	Wymaganie zostało pozytywnie zatwierdzone, zakończono wszelkie uzgodnienia z Wykonawcą i podjęto decyzje o jego realizacji i wymaganie zostało przekazane do Wykonawcy w celu realizacji.
9	Zatwierdzone	Wstrzymane	Zespół ds. monitorowania wymagań podjął decyzje o tymczasowym wstrzymaniu realizacji wymagania.

10	W trakcie implementacji	Zaimplementowane	Wymaganie zostało pomyślnie zrealizowane i zostały zrealizowane jego testy z wynikiem pozytywnym.
11	Wstrzymane	W trakcie implementacji	Zespół do ds. monitorowania wymagań podjął decyzje o realizacji wymagania i wymaganie zostało przekazane do implementacji do Wykonawcy.
12	Wstrzymane	Odrzucone	Zespół do ds. monitorowania wymagań podjął decyzje o nierealizowaniu wymagania.
13	Zaimplementowane	Koniec	Wymaganie zostało zrealizowane, przeszło pozytywnie testy, powstał raport z testów, została zakończona wszelka dokumentacja powykonawcza oraz Zamawiający dokonał formalnego odbioru wymagania.

3.3 Śledzenie wymagań

Śledzenie wymagań ma na celu kontrolowanie wpływu zmian jednych wymagań na pozostałe z nim powiązane oraz zmian statusów tych wymagań.

Narzędziami do śledzenia wymagań są raporty. Szczegółowy opis raportów znajduje się w rozdziale 5.2 Szablony raportów. Raporty umożliwiają generowanie informacji o wymaganiach, komponentach i systemach powiązanych z danym wymaganiem lub grupą wymagań.

Do poprawnego śledzenia wymagań konieczne jest utrzymywanie aktualności statusów wymagań po zmianach stanów wymagań. Śledzenie może odbywać się w całym przebiegu cyklu życia wymagania.

Wszelkie zmiany w wymaganiach są zapisywane w narzędziu z podaniem następujących informacji:

- Użytkownika dokonującego zmiany
- Daty
- Nazwy obiektu modyfikowanego
- Atrybut który uległ zmianie
- Oryginalna wartość atrybutu
- Zmodyfikowana wartość atrybutu

Procedura śledzenia wymagań składa się z następujących kroków:

1. Określenie wymagania lub grupy wymagań, dla których należy zidentyfikować wymagania powiązane
2. Określenie typu relacji pomiędzy wymaganiami, które mają zostać zidentyfikowane (wymaganie, z których korzysta wskazane wymaganie (grupa wymagań) lub/i relacja przeciwna – wymagania, które korzystają ze wskazanego wymagania (grupy wymagań))
3. Zidentyfikowanie wymagań będących w wybranej relacji ze wskazanym wymaganiem (poprzez raporty)
4. Analiza wymagań i ich zmian

3.4 Weryfikacja wymagań

Weryfikacja wymagań ma na celu zapewnienie jakości i zgodności zdefiniowanym wymaganiom. Wymagania, które nie zapewniają wystarczającej ilości informacji lub nie można ich przetestować, w celu oceny i zatwierdzenia przez zainteresowane strony nie będą dalej weryfikowane ze względu na brak jakości. Wymaganie takie nie będzie dalej wykorzystywane, należy je uszczegółowić jeśli jest to możliwe bądź usunąć. Weryfikacja wymagań odbywa się na przeglądzie zgodności wymagań prowadzonego przez specjalnie powołane w tym celu ciało lub upoważnione osoby po stronie zamawiającego. Metoda badania zgodności obejmuje dokumenty projektowe na podstawie których po analizie powstanie raport z badania.

Przedmiotem badania zgodności są następujące dokumenty projektowe:

- Architektura biznesowa;
- Architektura systemów informatycznych;
- Architektura technologiczna analiza dostępnej infrastruktury IT, Architektura technologiczna;
- Opis Przedmiotu Zamówienia na system informatyczny;
- Projekty funkcjonalne i techniczne;
- Plan testów;
- Raporty ze śledzenia wymagań;

oraz inne dokumenty, które swym zakresem merytorycznym obejmują przynajmniej część wyżej wymienionych.

Zgodność wymagań oznacza, że zapisane bądź zaimplementowane wymagania:

1. Wspierają założoną strategię i kierunki rozwoju zawarte w dokumentach:
 - a. Studium Wykonalności danego projektu
 - b. Architektura biznesowa danego projektu
2. Spełniają wytyczne SIG dotyczące:
 - a. Architektury biznesowej
 - b. Architektury systemów informatycznych
 - c. Architektury technologicznej
 - d. Standardów SIG
 - e. Architektury SIG
3. Spełniają Pryncypia Architektoniczne
4. Spełniają potrzebę interesariuszy
5. Zapisane statusy są zgodne ze stanem faktycznym wymagania

Wynikiem przeglądu jest określenie jednego z następujących poziomów zgodności:

1. Niezgodny – wykonanie nie ma części wspólnych ze specyfikacją i dokumentacją.
2. Zgodny w części – wykonanie ma pewną część wspólną ze specyfikacją i te wspólne cechy są wykonane zgodnie ze specyfikacją. Aczkolwiek niektóre cechy specyfikacji nie są wykonane, a także wykonanie posiada zbiór cech, które nie są częścią specyfikacji i dokumentacji.
3. Zgodny niepełny – niektóre cechy specyfikacji nie są wykonane, ale wszystkie wykonane są zgodne ze specyfikacją.

4. Zgodny nadmiarowy – wszystkie cechy specyfikacji zostały uwzględnione i wykonane, przy czym wykonanie zawiera inne cechy, które nie są częścią specyfikacji.
5. Zgodny w pełni – istnieje idealna zbieżność cech, na które wskazuje specyfikacja oraz wykonanie. Wszystkie cechy wskazane w specyfikacji są zrealizowane, a także nie istnieją inne cechy, które nie są uwzględnione w specyfikacji.
6. Niezgodny w części – którykolwiek stan z poprzednich, w którym to niektóre cechy specyfikacji są uwzględnione, ale w sposób niezgodny ze specyfikacją.

Weryfikacja wymagań może odbywać się podczas całego przebiegu cyklu życia wymagania.

Przegląd dotyczący wymagań zostanie uruchomiony w przypadku :

- Przy zaistnieniu konieczności zmian wymagań
- Przy implementacji systemów
- Przy wystąpieniu zdarzeń:
 - o Inicjacja projektu
 - o Projekt wstępny
 - o Kluczowe zmiany/decyzje projektowe
 - o Ad hoc

4 Jednolita baza wymagań

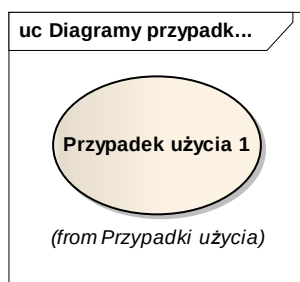
4.1 Metamodel

Podejście do metamodelu:

1. Identyfikacja potrzeby / problemu
2. Identyfikacja przypadków użycia
3. Identyfikacja przypadków użycia współdzielonych
4. Identyfikacja wymagań
5. Identyfikacja wymagań współdzielonych
6. Identyfikacja modelu przypadków użycia
7. Identyfikacja diagramów wymagań
8. Identyfikacja diagramów przeglądowych
9. Identyfikacja widoków przeglądowych

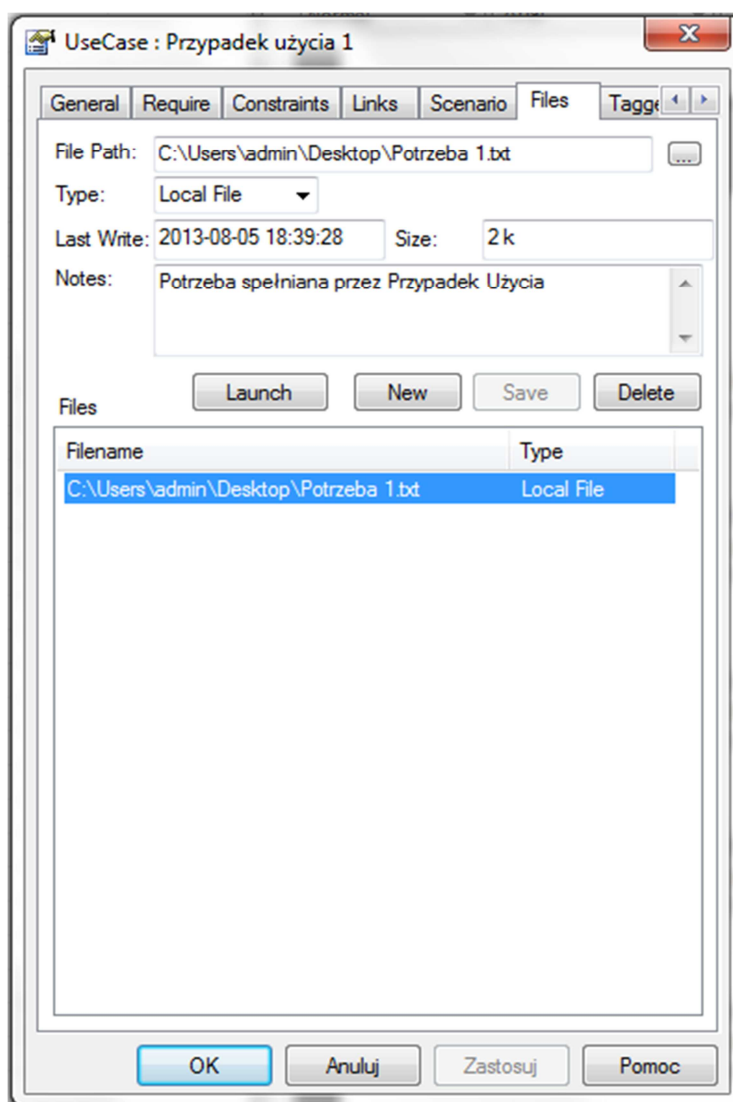
Potrzeby interesariuszy są zaspokajana poprzez wprowadzenie określonych przypadków użycia. Potrzeby zapisywane są językiem naturalnym w formie opisu do pliku tekstowego.

Przypadek użycia określa funkcjonalności systemu i ma postać jak na poniższym rysunku.



Rysunek 4: Przypadek użycia

Do przypadku użycia przypisane są linki ze ścieżką adresową pliku bądź plików z zapisaną potrzebą tak jak pokazano to na poniższym rysunku.

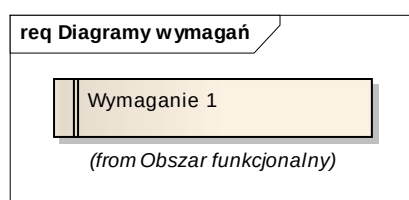


Rysunek 5: Przypisanie potrzeby do przypadku użycia

W przypadku użycia jest także zapisywane wymaganie jakie z niego wynika tak jak na poniższym rysunku.

Rysunek 6: Przypisanie wymagania do przypadku użycia

Wymaganie określa jakie funkcji ma realizować system, zaprezentowane jest na poniższym rysunku.



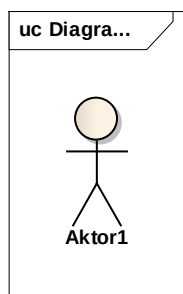
Rysunek 7: Wymaganie

Podstawowym elementem diagramu przypadków użycia jest:

- aktor,
- przypadek użycia,

- relacja.

Aktor odzwierciedla użytkownika systemu, bądź inny system wchodzący w interakcje z systemem, jego symbol został zaprezentowany poniżej.

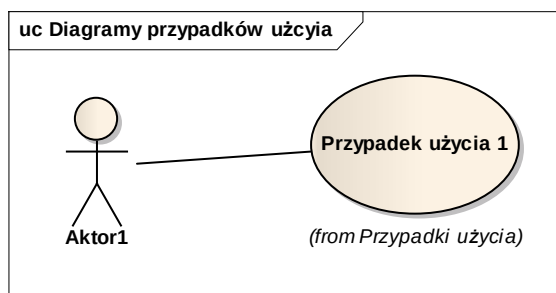


Rysunek 8: Aktor

Każdy aktor musi być przyporządkowany minimum jednemu przypadkowi użycia, a każdy przypadek użycia musi być przyporządkowany minimum jednemu aktorowi, odstępstwem jest gdy przypadki użycia są połączone związkami zawierania lub rozszerzenia, to bazowy przypadek użycia staje się punktem łączącym aktora z danym przypadkiem użycia.

Aktor połączony może być z przypadkiem użycia poprzez:

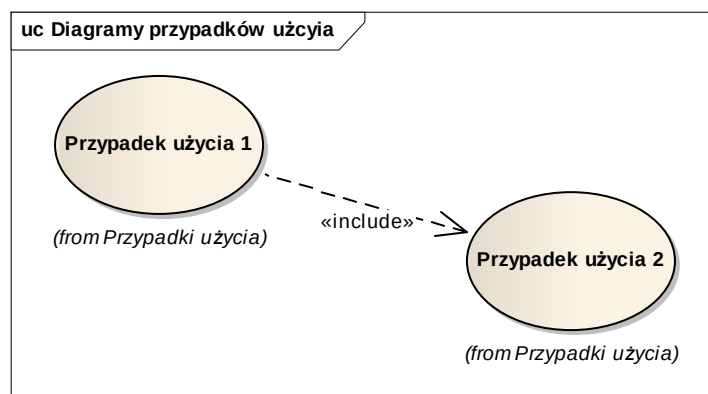
- asocjację



Rysunek 9: Asocjacja pomiędzy aktorem a przypadkiem użycia

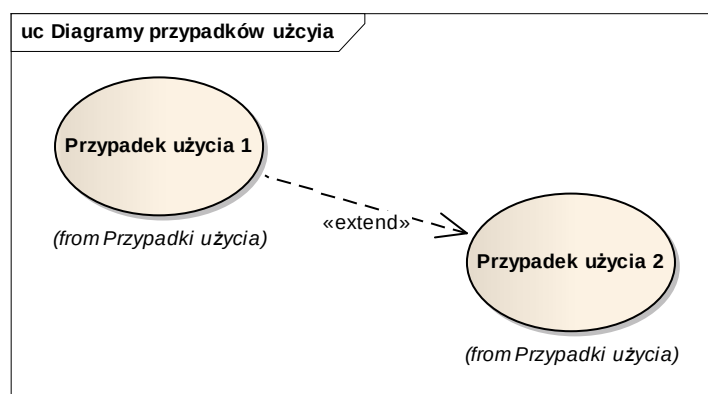
Związki między przypadkami użycia jakie mogą wystąpić przedstawione zostały poniżej:

1. Includes (zawieranie) – w czasie wykonywania Przypadku użycia 1, wykonywana jest również funkcjonalność Przypadku użycia 2, Przypadek użycia 1 wykorzystuje do swojego wykonania Przypadku użycia 2.



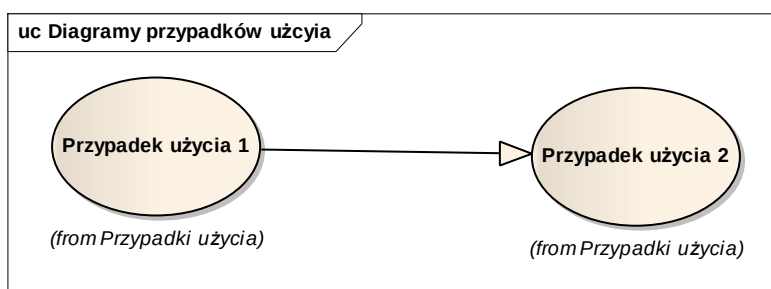
Rysunek 10: Związek includes między przypadkami użycia

2. Extends (rozszerzenie) - szczególnym przypadkiem Przypadku użycia 2 jest Przypadek użycia 1, Przypadek użycia 1 uwzględnia nową sytuację wyjątkową nie obsługiwaną przez Przypadek użycia 2, strzałka skierowana jest do operacji bardziej ogólnej.



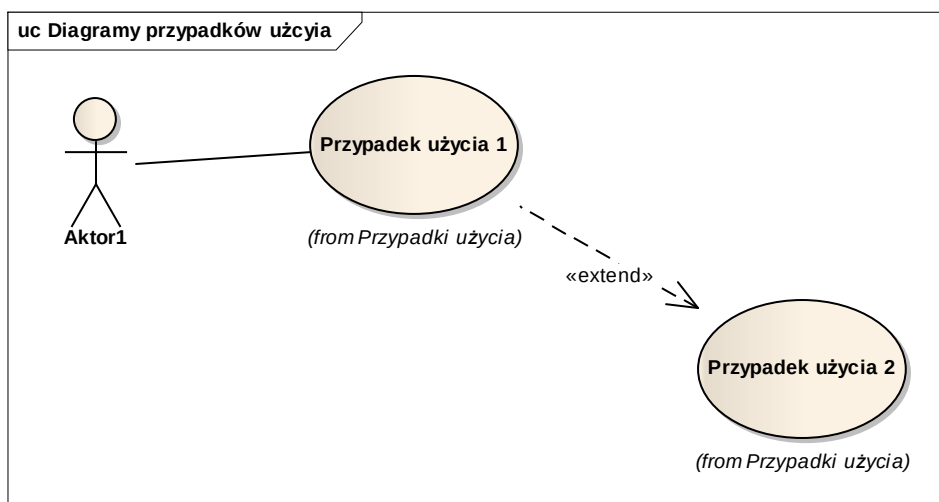
Rysunek 11: Związek extends między przypadkami użycia

3. Generalization (uogólnienie) – Przypadek użycia 2 jest bytem ogólnym, a Przypadek użycia 1 jest bytem szczegółowym, Przypadek użycia 1 może zastąpić Przypadek użycia 2.



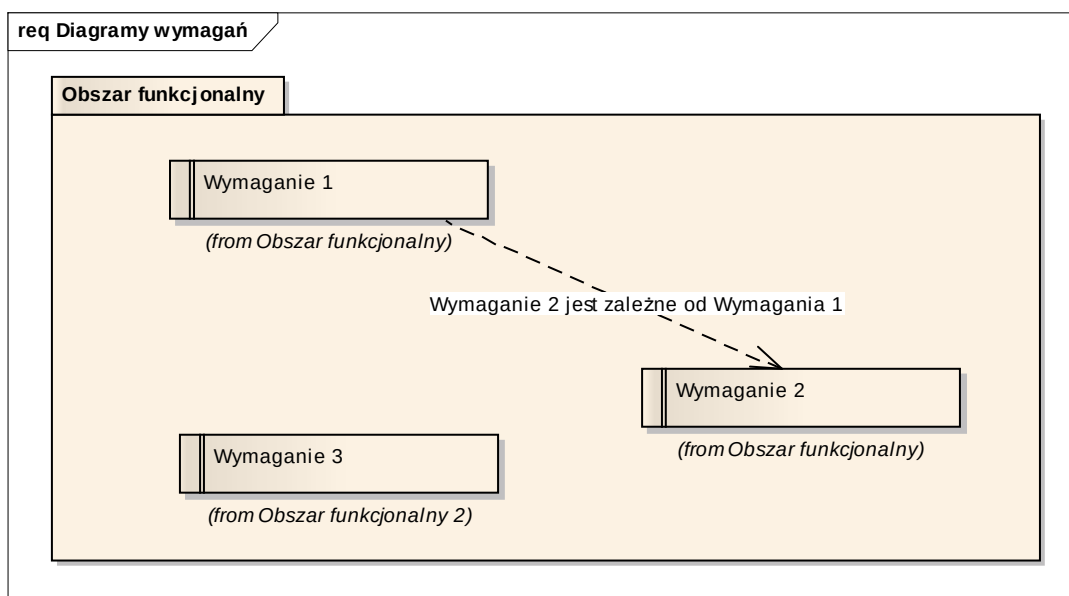
Rysunek 12: Związek generalization między przypadkami użycia

Poniżej zaprezentowano diagram przypadków użycia.



Rysunek 13: Diagram przypadków użycia

Diagram wymagań zawiera hierarchie wymagań połączonych w obszary, przykład został zaprezentowany poniżej. Poniższy rysunek zawiera jeden obszar funkcjonalnych w ramach którego znajdują się trzy wymagania, gdzie Wymaganie 2 jest zależne od Wymagania 1, Wymaganie 3 pochodzi z innego obszaru.



Rysunek 14: Diagram wymagań

4.2 Organizacja ról w zarządzaniu wymaganiami

Rola to zestaw odpowiedzialności, aktywności i uprawnień, które są nadane konkretnej osobie lub grupie osób. Jedna osoba lub zespół może pełnić więcej niż jedną rolę, o ile nie zostało to określone inaczej. Niniejsza metodyka opiera się o role, które uczestniczą w zarządzaniu projektami, a także powołuje nowe role wynikające wprost z zarządzania wymaganiami.

W procesie zarządzania wymaganiami uczestniczą następujące role :

- Administrator wymagań,

- Bibliotekarz projektu,
- Właściciel wymagania,
- Zespół ds. monitorowania wymagań,
- Kierownik projektu po stronie GUGiK,
- Kierownik projektu po stronie Wykonawcy,
- Rada Architektury.

Administrator wymagań odpowiada za zarządzanie bazą wymagań.

Do obowiązków administratora wymagań należą:

- utrzymanie bazy wymagań, za zapewnienie jej integralności, spójności i kompletności,
- zarządzanie strukturą bazy wymagań,
- utrzymywanie w aktualności procedur związanych z zarządzaniem bazą wymagań,
- zarządzanie dostępnymi szablonami raportów z bazy wymagań,
- raportowanie o stanie bazy wymagań do Rady Architektury i Kierownictwa GUGiK oraz Kierowników projektów po stronie GUGiK,
- identyfikowanie nieprawidłowości i eskalowanie informacji o zidentyfikowanych nieprawidłowościach,
- zapewnienie należytej współpracy pomiędzy poszczególnymi bibliotekarzami wymagań.

Wskazane jest, aby rolę administratora wymagań pełniła osoba po stronie GUGiK. Dopuszcza się powołanie ról zastępców Administratora, którzy wspierają Administratora w pracach operacyjnych. Role zastępcy administratora mogą być pełnione po stronie Wsparcia.

Bibliotekarz wymagań to rola odpowiadająca za administrowanie wymaganiami po stronie projektu. Dla każdego projektu powinien być powołany bibliotekarz wymagań.

Do obowiązków bibliotekarza wymagań należą:

- utrzymywanie aktualności bazy wymagań w zakresie wymagań dotyczących projektu, wprowadzanie do bazy wymagań wszelkich zmian związanych ze statusami wymagań lub jego atrybutami,
- weryfikowanie wpisów w bazie wymagań dotyczących wymagań powiązanych z projektem,
- udostępnianie informacji o wymaganiach zapisanych w bazie wymagań, w tym przygotowywanie wycinków z bazy wymagań dla wykonawcy,
- przeprowadzanie inwentaryzacji wymagań dotyczących danego projektu,
- wyjaśnianie zidentyfikowanych niezgodności w zakresie wymagań dotyczących danego projektu.

Wskazane jest, aby rolę bibliotekarza wymagań pełniła jedna osoba dla poszczególnego projektu. Rolę Bibliotekarza wymagań może pełnić osoba po stronie GUGiK lub Wsparcia.

Właściciel wymagania odpowiada za pojedyncze wymaganie bądź całą grupę wymagań dotyczących podsystemu. Każde wymaganie bez względu na fazę w jakiej przebywa w cyklu życia posiada swojego właściciela i tylko za jego zgodą można wprowadzić zmiany w wymaganiu. Do obowiązków właściciela wymagania należą:

- uzgadnianie możliwości ponownego użycia w innych projektach wymagań, które mu podlegają,
- uzgadnianie z innymi właścicielami wymagań wpływu wymagań, które mu podlegają,
- uzgadnianie modyfikacji wymagań wynikającej z potrzeb innych projektów,
- podejmowanie decyzji o uszczegóławianiu wymagania.

Rolę właściciela wymagań pełni osoba wskazana przez Kierownika Projektu, w szczególności rolę tę może pełnić sam kierownik lub inny członek zespołu projektowego (np. architekt projektu). Rola właściciela wymagania powinna być pełniona po stronie Zamawiającego. W projekcie może być wielu właścicieli wymagań, natomiast jedno wymaganie może mieć tylko jednego właściciela.

Zespół ds. monitorowania wymagań to zespół powołany z członków GUGiK, który jest zobowiązany do monitorowania stanu realizacji wymagań. Skład zespołu jest zmienny i zależy od zakresu monitorowania. Stałym członkiem zespołu jest administrator wymagań, a w zależności od potrzeby do zespołu włączani są Kierownicy projektu po stronie GUGiK, właściciele wymagań, oraz bibliotekarze wymagań. Do zespołu ds. monitorowania wymagań mogą należeć także osoby ze Wsparcia projektu, szczególnie gdy pełnią jedną z ról w procesie zarządzania wymaganiami.

Do obowiązków zespołu ds. monitorowania wymagań należy m.in.:

- odpowiedzialność za śledzenie wymagań,
- odpowiedzialność za weryfikacje wymagań,
- odpowiedzialność za tymczasowe wstrzymanie realizacji wymagania,
- odpowiedzialność za odrzucenie realizacji wymagania.

Kierownik Projektu po stronie GUGiK odpowiada za wykonywanie w Projekcie działań związanych z zarządzaniem wymaganiami wynikających z metodyki. Do obowiązków Kierownika Projektu po stronie GUGiK należy:

- powołanie bibliotekarza wymagań dla projektu,
- powołanie właścicieli wymagań,
- umieszczenie zapisów dot. zgodności z metodyką z dokumentacji przetargowej i projektowej oraz stosowanie jej zapisów w trakcie współpracy z Wykonawcą,
- zatwierdzanie wymagań.

Kierownik Projektu po stronie Wykonawcy odpowiada za wykonywanie w Projekcie działań związanych z zarządzaniem wymaganiami wynikających z metodyki. Do obowiązków Kierownika Projektu po stronie Wykonawcy w zakresie zarządzania wymaganiami należy m.in.:

- przekazywanie wymagań w narzędziu zgodnym z zaleceniami metodyki,
- udział w ujednolicaniu wymagań z Zamawiającym,
- udział w uszczegóławianiu wymagań.

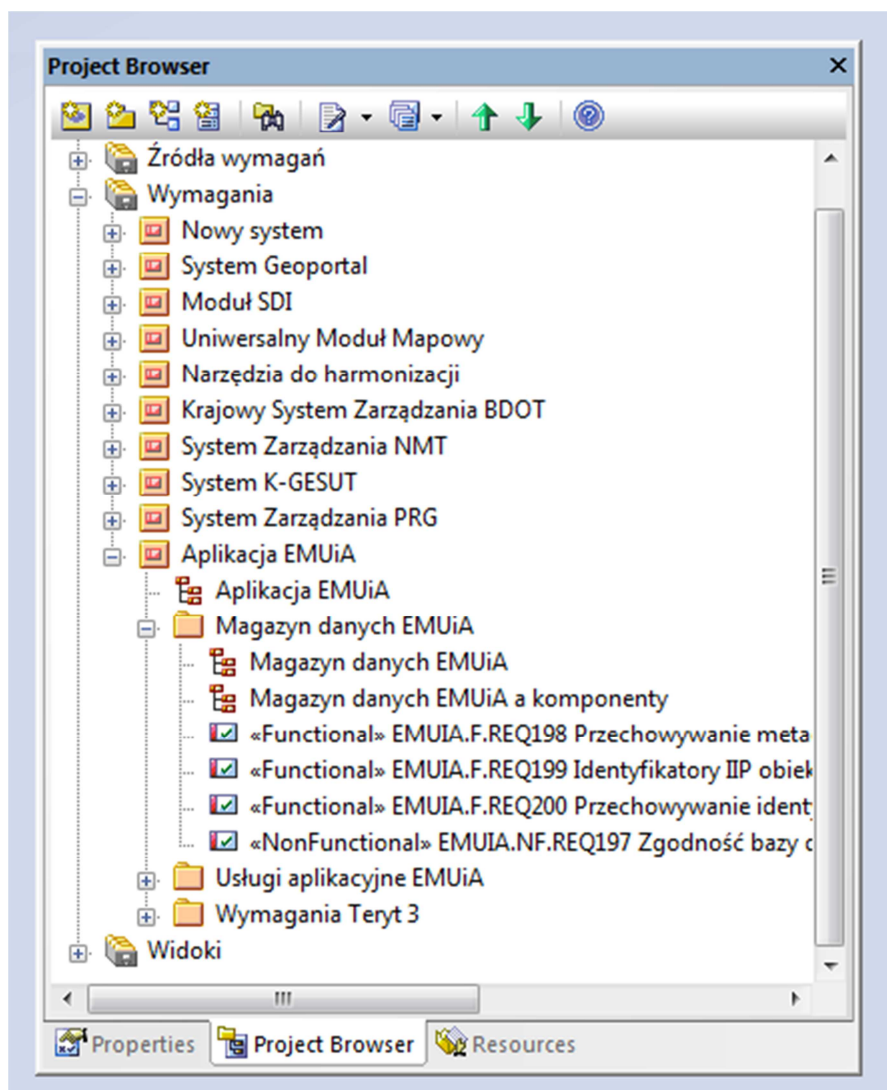
Rada Architektury uczestniczy w zarządzaniu wymaganiami jako ciało decyzyjne, nadzorujące realizację metodyki, a w szczególności podejmujące decyzje dotyczące wymagań wpływających na więcej niż jeden system.

4.3 Struktura bazy wymagań

Struktura bazy wymagań została utworzona w oparciu o dostępne elementy i relacje właściwe dla notacji UML, rozszerzona na potrzeby niniejszej metodyki.

Struktura bazy wymagań utworzona została przy wykorzystaniu następujących elementów:

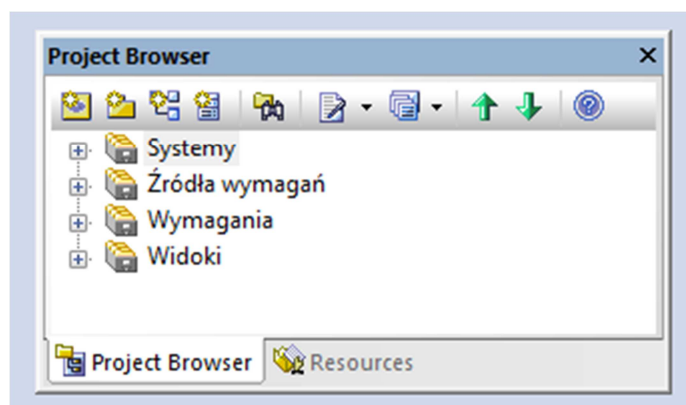
- Węzeł główny (ang. *Root node*),
- Model,
- Pakiet (ang. *Package*),
- Diagramy i poszczególne obiekty.



Rysunek 15: Elementy wykorzystane do tworzenia struktury bazy wymagań

Węzły główne oznaczane są ikoną . Za ich pomocą baza wymagań podzielona została logicznie na części pokazujące najważniejsze aspekty opisywania systemów informatycznych, które zostały objęte metodyką zarządzania wymaganiami.

Cztery węzły główne, tj. Systemy, Źródła wymagań, Wymagania, Widoki stanowi ogólną strukturę bazy wymagań.



Rysunek 16: Ogólna struktura bazy wymagań

W każdym z węzłów znajdują się modele odpowiadające poszczególnym systemom objętym metodyką zarządzania wymaganiami. Modele grupują diagramy i elementy stanowiące artefakty powiązane z zarządzaniem wymaganiami dla danego systemu. W zależności od rodzaju zawartych w danym modelu diagramów i elementów modele mogą być oznaczane różnymi ikonami:

-  dla modelu grupującego diagramy komponentów (*ang. Component diagram*),
-  dla modelu grupującego diagramy wymagań (*ang. Requirement diagram*),
-  dla modelu grupującego diagramy przypadków użycia (*ang. Use case diagram*),
-  dla modelu grupującego diagramy klas (*ang. Class diagram*).

Diagramy i poszczególne elementy modeli mogą być grupowane w pakiety, oznaczona ikoną . Dopuszczalne jest również zagnieżdżanie pakietów, czyli umieszczanie pakietów w pakietach.

Diagramy i poszczególne elementy mogą być umieszczane bezpośrednio w modelach lub w pakietach. Szczegółowy opis poszczególnych diagramów, elementów oraz dopuszczalnych relacji pomiędzy elementami przedstawiony został w rozdziałach odpowiadających poszczególnym węzłom głównym.

4.4 Procedury bezpieczeństwa

Poprzez zapewnienie bezpieczeństwa wymagań rozumiane jest zapewnienie ich dostępności, integralności i poufności. Rolą odpowiedzialną na zapewnianie bezpieczeństwa wymagań jest Administrator wymagań. Administrator jest zobowiązany m.in. do:

1. ochrony i bezpieczeństwa danych zawartych w zbiorach wymagań systemów informatycznych GUGiKu,
2. podejmowania stosownych działań w przypadku wykrycia nieuprawnionego dostępu do wymagań lub naruszenia zabezpieczenia danych wymagań,
3. nadzoru i kontroli narzędzi informatycznych służących do przetwarzania wymagań,

4. fizycznego zabezpieczenia wymagań oraz obiektów, w których są gromadzone i przetwarzane,
5. cykliczne przygotowywanie kopii bezpieczeństwa bazy danych.

Procedura bezpieczeństwa dla zarządzania wymaganiami:

1. Przypisanie odpowiedzialności w zakresie bezpieczeństwa wymagań.
2. Przypisanie wymagania do właściciela wymagań.
3. Zaangażowanie osób objętych rolami w zarządzaniu wymaganiami w bezpieczeństwo wymagań.
4. Przegląd bezpieczeństwa wymagań w trakcie weryfikacji wymagań.
5. Monitorowanie wymagań poprzez raporty zmian.
6. Zapewnienie bezpieczeństwa bazy wymagań poprzez wydawanie Wykonawcą jedynie wycinka bazy wymagań obejmującego niezbędne wymagania do realizacji.
7. Zapewnienie dostępu do bazy wymagań jedynie wyznaczonym rolom.
8. Systematyczne dokonywanie kopii zapasowych bazy wymagań (back up bazy wymagań).

Zasady dokonywania kopii zapasowych bazy wymagań są następujące:

- kopia zapasowa powinna być przechowywana poza lokalizacją w której znajdują się oryginalne pliki,
- po wykonaniu kopii należy zweryfikować jej poprawność,
- kopia zapasowa powinna być całkowitym obrazem bazy wymagań,
- kopia zapasowa powinna być w każdej chwili zdalna do użycia,
- kopia zapasowa powinna być wykonywana po każdej wprowadzonej zmianie lub grupie zmian.

4.4.1 Procedura umieszczania wymagań

Umieszczanie wymagań powinno składać się z poniższych kroków:

1. Zdefiniowane wymaganie zamieszczane jest przez bibliotekarza projektu w systemie ze statusem propozycja.
2. Wymaganie ma ustalone atrybuty takie jak: wersja, powiązanie, priorytet i źródło.
3. Zdefiniowane wymaganie jest zatwierdzane przez Radę Architektury bądź kierownika projektu. Bibliotekarz projektu zmienia status wymagania na zatwierdzony.
4. W sytuacji gdy wymaganie zostaje zrealizowane bibliotekarz zmienia status wymagania na zaimplementowane. W innej sytuacji status wymagania pozostaje na wartości zatwierdzony i oczekuje do zrealizowania w przyszłości.
5. Bibliotekarz ocenia zaimplementowane wymaganie pod względem ponownego wykorzystania, jeśli wymaganie spełnia wymogi do ponownego wykorzystania to zamieszcza je w katalogu z wymaganiami współdzielonymi.
6. Wymaganie które zostało zweryfikowane w czasie cyklu życia wymagania otrzymuje status zweryfikowane, wprowadzone do systemu przez bibliotekarza projektu.
7. Wykonanie kopii zapasowej bazy wymagań.

5 Plan wdrożenia metodyki zarządzania wymaganiami

5.1 Źródła wymagań

Metodyką zarządzania wymaganiami będą objęte projekty:

- GBDOT
- Geoportal 2
- ISOK
- K-GESUT
- TERYT 2
- TERYT 3

Źródłem wymagań dla poszczególnych systemów są dokumenty, dla projektów zakończonych jest to dokumentacja powykonawcza, a dla projektów w trakcie realizacji są to dokumenty projektów funkcjonalnych.

Wykaz poszczególnych dokumentów dla poszczególnych projektów:

- GBDOT
 - o GBDOT Projekt funkcjonalny 1.07
 - o Załącznik nr 1 do SOPZ – Architektura Krajowego Systemu
- Geoportal 2
 - o Dokumentacja Funkcjonalna systemu Geoportal 1.02
 - o Projekt Funkcjonalny systemu Geoportal – Pakiet 3 – Szyna Usług SOA 1.05
 - o Projekt Funkcjonalny systemu Geoportal – Pakiet 4 – Narzędzia do zarządzania publikacją danych i usług systemu Geoportal 1.03
 - o Projekt Funkcjonalny systemu Geoportal – Pakiet 5 -Aplikacja zarządzania treścią witryny Geoportal 1.01
 - o Projekt Funkcjonalny systemu Geoportal – Pakiet 6 – Narzędzia do zarządzania warstwą prezentacji systemu Geoportal 1.04
 - o Projekt Funkcjonalny systemu Geoportal – Pakiet 7 – Narzędzia do gromadzenia, zarządzania i publikowania zdjęć lotniczych 1.04
 - o Projekt Funkcjonalny systemu Geoportal – Pakiet 8 – Narzędzia do gromadzenia, zarządzania i publikowania ortofotomapy 1.05
 - o Projekt Funkcjonalny systemu Geoportal – Pakiet 9 – Narzędzia do gromadzenia, zarządzania i publikowania skorowidzów 1.04
 - o Projekt Funkcjonalny systemu Geoportal – Pakiet 10 – Narzędzia do gromadzenia, zarządzania i publikowania NMT 1.02
 - o Projekt Funkcjonalny systemu Geoportal – Pakiet 11 – Narzędzia do gromadzenia, zarządzania i publikowania Osnowy 1.03
 - o Projekt Funkcjonalny systemu Geoportal – Pakiet 12 – Narzędzia do gromadzenia, zarządzania i publikowania TBD 1.03
 - o Projekt Funkcjonalny systemu Geoportal – Pakiet 13 – Narzędzia do gromadzenia, zarządzania i publikowania PRNG 1.05
 - o Projekt Funkcjonalny systemu Geoportal – Pakiet 15 – Narzędzia do gromadzenia i publikowania BDO 1.03
 - o Projekt Funkcjonalny systemu Geoportal – Pakiet 16 – Narzędzia do gromadzenia i publikowania Vmap2 1.02

- o Projekt Funkcjonalny systemu Geoportal – Pakiet 17 – Narzędzia do gromadzenia, zarządzania i publikowania danych INSPIRE 1.04
 - o Projekt Funkcjonalny systemu Geoportal – Pakiet 18 – Aplikacja zarządzania zasobem CODGiK 1.05
 - o Projekt Funkcjonalny systemu Geoportal – Pakiet 19 – Aplikacja monitorowania i sprawozdawczości INSPIRE (monitoring wewnętrzny) 1.04
 - o Dokumentacja Techniczna systemu Geoportal – Pakiet 20 – Aplikacja monitorowania i sprawozdawczości INSPIRE (monitoring zewnętrzny) 1.01
- ISOK
 - o Załącznik nr 2 do Szczegółowego Opisu Przedmiotu Zamówienia – Architektura systemów informacyjnych GUGiK
 - o Załącznik nr 1 do SOPZ – Architektura Systemu Zarządzania NMT
 - o Projekt funkcjonalny SZNMT 1.05
- K-GESUT
 - o Zaktualizowany projekt funkcjonalny 1.05
- TERYT 2
 - o Specyfikacja wymagań funkcjonalnych i нефункциональных oraz Projekt funkcjonalny Aplikacji EMUiA
 - o Specyfikacja wymagań funkcjonalnych i нефункциональных oraz Projekt funkcjonalny Systemu Zarządzania PRG 6.03
- TERYT 3
 - o Opis przedmiotu zamówienia

Lista projektów, które są objęte niniejszą metodyką może zostać rozszerzona przez Radę Architektury.

5.2 Szablony raportów

Niniejszy rozdział zawiera zestawienie raportów z jednolitej bazy wymagań, które zostaną przygotowane jako predefiniowane. Poniższy opis szablonów prezentuje merytoryczny zakres informacji jaką będą prezentować, natomiast docelowe szablony zgodne z wytycznymi GUGiK zostaną zaimplementowane już w narzędziu.

- Zestawienie wymagań projektu wraz z ich cechami z modelu Wymagania
 - o Szablon zawiera wykaz wszystkich wymagań z nazwami, identyfikatorami, z opisem, statusem, fazą, wersją, źródłem, stopniem powinności, datą utworzenia oraz datą ostatniej modyfikacji
- Zestawienie wymagań współdzielonych projektu wraz z ich cechami z pakietu Wymagania współdzielone
 - o Szablon zawiera wykaz współdzielonych wymagań z nazwami, identyfikatorami, z opisem, statusem, fazą, wersją, źródłem, stopniem powinności, datą utworzenia oraz datą ostatniej modyfikacji
- Zestawienie przypadków użycia wraz z ich cechami z modelu Przypadki użycia
 - o Szablon zawiera wykaz wszystkich przypadków użycia z nazwami, identyfikatorami, z opisem, autorem, statusem, wersją, fazą, datą utworzenia oraz datą ostatniej modyfikacji, a także zawarta jest informacja o powiązaniu z wymaganiami

- Zestawienie przypadków użycia współdzielonych wraz z ich cechami z pakietu Przypadki użycia współdzielone
 - o Szablon zawiera wykaz wszystkich przypadków użycia z nazwami, identyfikatorami, z opisem, autorem, statusem, wersją, fazą, datą utworzenia oraz datą ostatniej modyfikacji, a także zawarta jest informacja o powiązaniu z wymaganiami
- Zestawienie diagramów wymagań z modelu Systemu
 - o Szablon zawiera wykaz wszystkich diagramów z nazwami, typami diagramów oraz zdjęciem diagramu
- Zestawienie wszystkich przeprowadzonych zmian
 - o Szablon zawiera wykaz wszystkich zmodyfikowanych elementów wraz z informacjami o autorze zmian, dacie zmiany, oryginalnej wartości pola, wartości docelowej pola

Przykładowe szablony obejmujące zakresem wymienione obszary informacji zostały zamieszczone w załącznikach.

5.3 Etapy wdrażania metodyki

Wdrożenie metodyki zarządzania wymaganiami zaprezentowanej w niniejszym dokumencie odbywać się będzie w trzech głównych etapach, a każdy z tych etapów posiadać będzie zadania jakie w ramach tego etapu powinny zostać zrealizowane. Etapy wraz zadaniami wdrożenia metodyki przedstawione są poniżej:

1. Utworzenie inicjalnej wersji bazy wymagań
 - a. Ustalenie miejsca przechowywania bazy wymagań
 - b. Utworzenie struktury bazy wymagań (m.in. podzielenie na systemy)
 - c. Dostosowanie repozytorium do wymagań metodyki (m.in. elementy opisu wymagań, statusy, atrybuty itp.)
 - d. Zdefiniowanie raportów
2. Wdrożenie metodyki w organizacji
 - a. Powołanie ról (dokładne zdefiniowanie obowiązków)
 - b. Przygotowanie szczegółowych procedur związanych z zarządzaniem wymaganiami i korzystaniem z bazy wraz z podręcznikami użytkownika dla administratora wymagań i bibliotekarza projektu
 - c. Warsztat dla zespołów projektowych prezentujące metodykę oraz wykonane prace
 - d. Warsztat dla uczestników procesu wymagań, obejmujący m.in. podstawy UML, podstawy korzystania z narzędzia EA, przedstawienie procedur związanych z zarządzaniem wymaganiami
3. Inicjalne zasilenie bazy wymagań
 - a. Ustalenie kolejności zasilania bazy poszczególnymi projektami (rozpoczęcie od zrealizowanych systemów przechodząc do systemów z których korzystają inne systemy)
 - b. Weryfikacja aktualności dokumentacji
 - c. Ustalenie sposobu zasilania (wprowadzanie wymagań poprzez pliki, bądź ręcznie itp.)
 - d. Zasilenie bazy wymagań
 - e. Identyfikacja powiązań i wprowadzenie ich do systemu

- f. Przegląd poprawności zasilenia (przegląd pod względem: kompletności – zasilenie wszystkimi wymaganiami i wprowadzenie wszystkich powiązań, spójności – identyfikacja sprzecznych wymagań, jakości – sprawdzenie wypełnienia wszystkich obowiązkowych pól)
- g. Próbne raportowanie

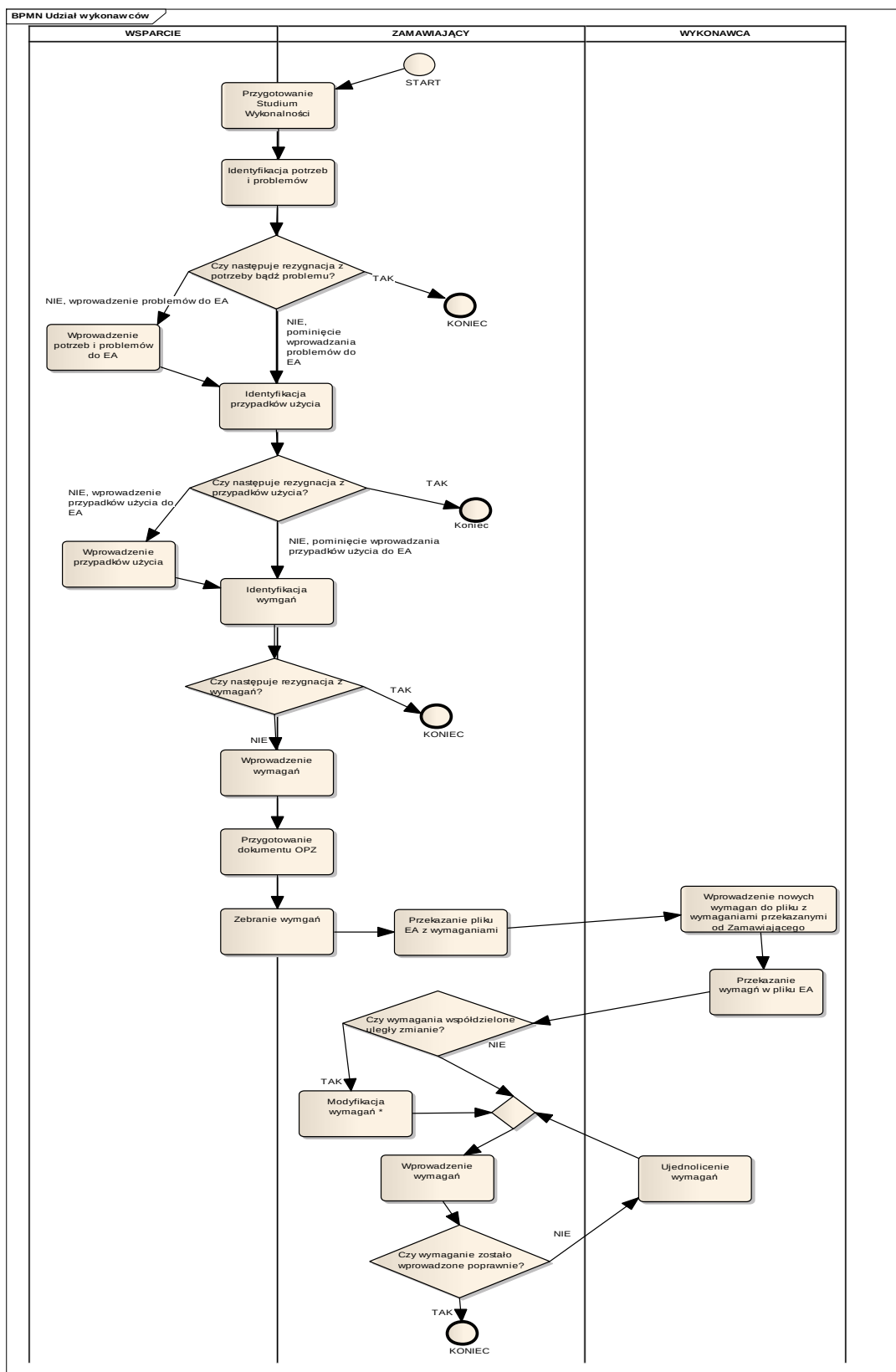
Rekomendowany jest następujący podział odpowiedzialności za realizację poszczególnych etapów wdrożenia metodyki:

Krok	Odpowiedzialność
1. Utworzenie inicjalnej wersji bazy wymagań	Projekt Geoportal 2
2. Wdrożenie metodyki w organizacji	
a. Powołanie ról (dokładne zdefiniowanie obowiązków)	Projekt Geoportal 2 oraz poszczególne projekty objęte metodyką (w zakresie powołania roli Bibliotekarza projektu)
b. Przygotowanie szczegółowych procedur związanych z zarządzaniem wymaganiami i korzystaniem z bazy wraz z podręcznikami użytkownika dla administratora wymagań i bibliotekarza projektu	Projekt Geoportal 2
c. Warsztat dla zespołów projektowych prezentujące metodykę oraz wykonane prace	Projekt Geoportal 2
d. Warsztat dla uczestników procesu wymagań, obejmujący m.in. podstawy UML, podstawy korzystania z narzędzia EA, przedstawienie procedur związanych z zarządzaniem wymaganiami	Projekt Geoportal 2
3. Inicjalne zasilenie bazy wymagań	
a. Ustalenie kolejności zasilania bazy poszczególnymi projektami (rozpoczęcie od zrealizowanych systemów przechodząc do systemów z których korzystają inne systemy)	Projekt Geoportal 2
b. Weryfikacja aktualności dokumentacji	Projekty objęte metodyką
c. Ustalenie sposobu zasilania (wprowadzanie wymagań poprzez pliki, bądź ręcznie itp.)	Projekt Geoportal 2 w uzgodnieniu z projektami objętymi metodyką
d. Zasilenie bazy wymagań	Projekt Geoportal 2 w uzgodnieniu z projektami objętymi metodyką
e. Identyfikacja powiązań i wprowadzenie ich do systemu	Projekt Geoportal 2 w uzgodnieniu z projektami objętymi metodyką
f. Przegląd poprawności zasilenia (przegląd pod względem: kompletności – zasilenie wszystkimi wymaganiami i wprowadzenie wszystkich powiązań, spójności – identyfikacja sprzecznych wymagań, jakości – sprawdzenie wypełnienia wszystkich obowiązkowych pól)	Projekt Geoportal 2
g. Próbne raportowanie	Projekt Geoportal 2

Realizacja ww. zadań może zostać zlecona przez poszczególne projekty podmiotom świadczącym usługi wsparcia. Nadzór nad realizacją poszczególnych zadań powinien być pełniony przez podmiot świadczący wsparcie dla Projektu Geoportal 2. Szczegółowy harmonogram realizacji poszczególnych etapów powinien zostać uzgodniony pomiędzy Projektem Geoportal 2 a pozostałymi projektami w momencie podjęcia decyzji o uruchomieniu prac nad wdrażaniem niniejszej metodyki.

5.4 Udział Wsparcia i Wykonawców w zarządzaniu wymaganiami

Proces współpracy ze Wsparciem i Wykonawcą odnośnie wprowadzenia wymagań rozpoczyna się od przygotowania Studium Wykonalności, które może wykonać Zespół Zamawiającego (Zamawiający bądź Wsparcie). Kolejnym krokiem jest identyfikacja potrzeb i problemów, które wykonuje Zespół Zamawiającego, a następnie zidentyfikowane problemy i potrzeby wprowadza do systemu Enterprise Architect. Następnym krokiem jest identyfikacja przypadków użycia wykonywana przez Zespół Zamawiającego, a po identyfikacji wprowadzenie tych przypadków użycia do systemu Enterprise Architect. Później identyfikowane są wymagania przez Zespół Zamawiającego. I także po tym zadaniu wprowadzane są wymagania do systemu Enterprise Architect. Kolejnym krokiem jest przygotowanie dokumentu Opisu Przedmiotu Zamówienia przez Zespół Zamawiającego i zebranie wymagań dla Wykonawcy. Zamawiający przekazuje formalnie do wykonawcy plik Enterprise Architecta z zebranymi wymaganiami. Wykonawca uszczegóławia wymagania lub wprowadza nowe i przekazuje do Zamawiającego w pliku Enterprise Architect. W przypadku gdy Wykonawca zmodyfikował wymaganie ze zbioru wymagań współdzielonych wymagane jest rozpoczęcie procesu modyfikacji wymagania, gdzie identyfikowany jest zakres zmiany, na nowo zdefiniowane jest wymaganie bądź zmiana treści wymagania za zgodą właściciela wymagania, dalej wymaganie jest zatwierdzane, jeśli pomyślnie zostanie zatwierdzone to przechodzi do fazy realizacji czyli zmiany, a potem do ponownego użycia. Po skończeniu procesu modyfikacji wymagania bądź braku zmian wymagań ze zbioru wymagań do ponownego użycia wprowadzane są wymagania przez bibliotekarza projektu do repozytorium wymagań. Bibliotekarz również weryfikuje poprawność techniczną i jakościową przekazanego pliku przez Wykonawcę, jeśli odbiega ona od założeń to Bibliotekarz ma obowiązek ujednolicić przekazany plik z Wykonawcą aż do momentu jak będzie zgodny z przyjętymi standardami. Pomyślne wprowadzenie wymagań do repozytorium Enterprise Architect kończy proces. Przebieg procesu został przedstawiony na poniższym rysunku.



Rysunek 17: Przebieg współpracy z Wykonawcą

Zadanie „Modyfikacja wymagań *” na powyższym rysunku odwołuje się do podprocesu z rozdziału „3.1.6 Modyfikacja wymagań” zamieszczonego na „Rysunku 2: Skrócona ścieżka cyklu życia wymagania”.

1. Zobowiązania Wykonawcy:

- Przekazanie wymagań w pliku z Enterprise Architect
- Zarządzanie wymaganiami zgodnie z niniejszą metodyką
- Dostosowanie się do standardów projektowych i zarządczych Zamawiającego
- Uzgodnienie bądź ujednolicenie pliku wymagań z bibliotekarzem projektu
- Identyfikacja problemów i potrzeb, przypadków użycia oraz wymagań na zlecenie Zamawiającego
- Wprowadzanie problemów i potrzeb, przypadków użycia oraz wymagań do repozytorium Enterprise Architect
- Przygotowywanie dokumentacji

2. Zobowiązania Zamawiającego

- Przygotowywanie dokumentacji
- Identyfikacja problemów i potrzeb, przypadków użycia oraz wymagań
- Przekazanie wymagań do Wykonawcy
- Odbiór wymagań od wykonawcy
- Uzgodnienie bądź ujednolicenie pliku wymagań z wykonawcą

W szczególności, gdy w projekcie jest podmiot świadczący wsparcie merytoryczno- organizacyjne to Zamawiający może przekazać część swoich obowiązków przedstawicielom Wsparcia.

6 Korzyści z zastosowania metodyki zarządzania wymaganiami

Wdrożenie jednolitej metodyki zarządzania wymaganiami dla systemu Geoportal oraz powiązanych z nim systemów dziedzinowych jest jednym z działań standaryzujących obszar zarządzania realizacją projektów, podejmowanym w ramach inicjatywy SIG.

Jednolite zarządzanie wymaganiami podczas wytwarzania systemów dziedzinowych umożliwia optymalizację działań zarządczych w projektach poprzez przyjęcie jednolitego sposobu współpracy z wykonawcami w tym zakresie, a także umożliwia korzystanie z doświadczeń z innych projektów.

Zastosowanie metodyki zarządzania wymaganiami uprości przygotowywanie dokumentacji przetargowej zarówno w zakresie merytorycznym (uprości się przygotowywanie specyfikacji wymagań), jak również organizacyjnym – możliwe będzie powielenie w dokumentacji przetargowej zapisów odnośnie sposobu zarządzania wymaganiami. Zastosowanie jednolitych narzędzi do zarządzania wymaganiami uprości również współpracę z wykonawcą i umożliwi lepszą rozliczalność wykonawców z realizowanych przez nich prac, także poprzez dokładne zdefiniowanie obowiązków obu stron w zakresie zarządzania wymaganiami.

Przedstawione w niniejszym dokumencie założenia odnośnie identyfikowania wymagań oraz powiązania ich z rzeczywistymi potrzebami pozwolą na dostarczanie w ramach projektów rozwiązań odpowiadającym na rzeczywiste potrzeby interesariuszy. Przechowywanie informacji o powiązaniach wymagań z potrzebami zmniejszy ryzyko niezgodności dostarczanych rozwiązań z oczekiwaniami odbiorców lub z wytycznymi wynikającymi z przepisów prawnych, które pojawiało się w projektach w wyniku modyfikowania treści wymagań. Rozwiązanie to uprości również komunikację pomiędzy interesariuszami projektów a analitykami specyfikującymi wymagania.

Dzięki zastosowaniu zgodnej z koncepcją przedstawioną w niniejszym dokumencie jednolitej bazy wymagań dla systemu Geoportal oraz powiązanych z nim systemów dziedzinowych możliwe będzie obrazowanie powiązań pomiędzy poszczególnymi systemami na różnych poziomach szczegółowości (wymaganie / komponent / system), a także śledzenie stanu realizacji komponentów i systemów powiązanych. Jednolita baza wymagań ułatwi również określanie i śledzenie wpływu zmian w wymaganiach na poszczególne systemy, a także współpracy pomiędzy poszczególnymi systemami.

Opisane w niniejszym dokumencie procedury, które wymuszają na systemach dziedzinowych aktualizację atrybutów wymagań zgodnie z ich implementacją, pozwalają na łatwiejsze utrzymywanie systemu i jego rozwój, gdyż dostarczają odpowiedzi na pytania o rzeczywisty zaimplementowany zakres systemu. Możliwość generowania raportów z systemu pozwala także na łatwiejsze dokumentowanie budowy systemu.

Prowadzenie jednolitej bazy wymagań gromadzącej wymagania dla wielu systemów pozwala również na łatwiejsze identyfikowanie niezgodności i niespójności pomiędzy projektami, a także poprzez wykorzystanie raportów eskalowanie niezgodności do kierownictwa i jednostek nadzorujących.

Największą korzyścią z wdrożenia niniejszej metodyki oraz zbudowania jednolitej bazy wymagań będzie możliwość ponownego używania wymagań. Umożliwi to osiągnięcie wymiernych korzyści w sferze ekonomicznej – oszczędność czasu i pieniędzy związana z brakiem powielania tych samych

rozwiązań oraz tych samych prac związanych z budową i dostarczaniem rozwiązań, a także z definiowaniem wymagań, które zostały już zdefiniowane w innych projektach. Ponowne wykorzystanie wymagań uprości również integrację z istniejącymi systemami, gdyż możliwe będzie skorzystanie z doświadczeń innych projektów i wykorzystanie sprawdzonych mechanizmów integracji.