



Agencja Restrukturyzacji i Modernizacji Rolnictwa

Zastępca Prezesa
Marek Deutsch

Wykonawcy

Wasze pismo z dnia:

Znak: Nasz znak:

Data:

ZP.258 .DPiZP.2610.18.2018.HŻ

02 .10.2018 r.

Dotyczy: postępowania o udzielenie zamówienia publicznego prowadzonego w trybie przetargu nieograniczonego na „Zakup usługi Utrzymania i rozwoju systemu informatycznego SI-Agencja*KI na okres 47 miesięcy”.

Działając na podstawie art. 38 ust. 4 ustawy z dnia 29 stycznia 2004 roku Prawo zamówień publicznych (Dz. U. z 2017 r., poz. 1579 ze zm.; dalej: „ustawa”) Agencja Restrukturyzacji i Modernizacji Rolnictwa z siedzibą w Warszawie przy Al. Jana Pawła II nr 70, zwana w dalszej treści pisma „Zamawiającym”, wprowadza następujące zmiany w treści specyfikacji istotnych warunków zamówienia (siwz):

Zmiana 1

Zamawiający zmienia termin składania i otwarcia ofert na dzień 16.10.2018 r.

W związku z tym rozdział IX.1. *Miejsce i termin składania i otwarcia ofert i JEDZ oraz miejsce i termin otwarcia ofert siwz*, pkt 1, 3 i 4 otrzymują poniższe brzmienie:

1. Ofertę należy umieścić w zamkniętej kopercie/paczce oznakowanej danymi Wykonawcy (nazwą i adresem) oraz zaadresowanej jak poniżej:

..... [nazwa i adres Wykonawcy]
Agencja Restrukturyzacji i Modernizacji Rolnictwa ul. Poleczki 33, 02-822 Warszawa „Oferta w postępowaniu na „Zakup usługi Utrzymania i rozwoju systemu informatycznego SI-Agencja*KI na okres 47 miesięcy”. Nr ref.: DPiZP.2610.18.2018 nie otwierać przed dniem 16.10.2018 r. godz. 12:00

3. Termin składania ofert i JEDZ w formie dokumentu elektronicznego, opatrzonego kwalifikowanym podpisem elektronicznym upływa w dniu **16.10.2018 r.** o godzinie **11:00**. Oferty otrzymane przez Zamawiającego po tym terminie zostaną zwrócone po upływie terminu przewidzianego na wniesienie odwołania, po uprzednim zawiadomieniu Wykonawcy o fakcie złożenia oferty po terminie.
4. Otwarcie ofert odbędzie się w dniu **16.10.2018 r.** o godzinie **12:00** w biurze Zamawiającego, pod adresem wskazanym w pkt 2.

Zmiana 2

Zamawiający dokonuje zmiany Załącznika 4C1 (Podręcznik IFPUG) do wzoru Umowy stanowiącej Załącznik nr 9 do siwz z wersji v2.6. na wersję v2.7. w brzmieniu stanowiącym załącznik do niniejszego pisma. Jednocześnie we wzorze Umowy, w tym w Załącznikach do wzoru Umowy, odwołania do wersji Podręcznika IFPUG v2.6. zastępuje się wersją Podręcznika IFPUG v2.7.

ZASTĘPCA PREZESA

Marek Deutsch

Umowa nr/DI/...../.....
Załącznik nr 4C1 Podręcznik IFPUG

Podręcznik stosowania metody
Punktów Funkcyjnych IFPUG
w Agencji Restrukturyzacji i Modernizacji
Rolnictwa

v2.7

1 października 2018

Spis treści

Wstęp	3
Definicje.....	4
Zakres stosowania metody PF IFPUG	5
Zasady stosowania metody PF IFPUG	8
Wprowadzenie do punktów funkcyjnych.....	8
Wstęp do metody PF IFPUG	8
Metoda PF IFPUG w ARiMR.....	10
Metoda PF IFPUG – zasady.....	12
Identyfikacja Plików danych IFPUG	14
Identyfikacja modelu funkcjonalnego IFPUG.....	14
Klasyfikacja Elementarnych Procesów.....	18
Wyliczenie rozmiaru funkcjonalnego elementów modelu IFPUG.....	21
Wyliczanie rozmiaru funkcjonalnego zmian.....	21
Wyliczanie zmian Plików Danych.....	22
Wyliczanie zmian Elementarnych procesów.....	23
Proces szacowania rozmiaru oprogramowania / zmiany oprogramowania.....	24
Zasady tworzenia i utrzymywania modelu artefaktów PF IFPUG	24
Zasady prezentacji wyników szacowania PF IFPUG w dokumentacji ofertowej.....	28
Zasady interpretacji metody PF IFPUG dla specyficznych typów funkcjonalności i zmian.	29
Stosowanie PF IFPUG	29
Stosowanie PF IFPUG dla złożonych funkcjonalności	34
Stosowanie PF IFPUG dla funkcjonalności sterowanych procesami.....	35
Stosowanie PF IFPUG dla złożonych raportów	36
Stosowanie PF IFPUG dla procesów wsadowych.....	36

WSTĘP

Niniejszy dokument (Podręcznik) ma na celu opisanie zakresu oraz zasad stosowania metody Punktów Funkcyjnych IFPUG w Agencji Restrukturyzacji i Modernizacji Rolnictwa. Niniejszy Podręcznik definiuje również zasady dokumentowania procesu wymiarowania oraz prezentacji wyników wymiarowania.

Metoda Punktów Funkcyjnych IFPUG (PF IFPUG) jest metodą pomiaru rozmiaru funkcjonalnego oprogramowania z perspektywy wymagań funkcjonalnych użytkownika. Metoda ta jest utrzymywana i rozwijana przez organizację *International Function Point Users Group* (www.ifpug.org).

Metodę PF IFPUG stosuje się w ARiMR zgodnie z zasadami opisanymi w dokumencie „Function Point Counting Practices Manual” w wersji 4.3.1 publikowanym przez organizację IFPUG między innymi w języku angielskim, a także publikacji organizacji NESMA (*Netherlands Software Metrics Association*) rozszerzającej zasady opisane w w/w dokumencie : „Function Point Analysis for Software Enhancement” (www.nesma.org, dokument dostępny w języku angielskim). Zasady opisane w publikacji organizacji NESMA mają zastosowanie dla szacowania zmian oprogramowania. Dokumenty te będą nazywane dokumentami bazowymi.

W Podręczniku zawarto tłumaczenia wybranych definicji oraz reguł pochodzących z wyżej wymienionych dokumentów. Celem tego zabiegu jest jedynie ogólne wprowadzenie czytelnika w zagadnienie metody punktów funkcyjnych oraz wprowadzenie kontekstu dla zasad opisanych w Podręczniku. Podręcznik dostarcza wskazówek w jaki sposób stosować zapisy dokumentów bazowych, może także je uściślać, natomiast z zasady nie zmienia ich znaczenia. W przypadku stwierdzenia rozbieżności pomiędzy interpretacją definicji (reguł) zawartych w dokumentach bazowych, a interpretacją ich tłumaczenia zawartą w niniejszym podręczniku, należy przyjąć interpretację wynikającą z treści dokumentów bazowych. Domyślnie wszystkie elementy opisane w dokumentach bazowych są stosowane w ARiMR, chyba że w podręczniku jawnie stwierdzono inaczej (np. że pewnych elementów metody się nie stosuje). Podręcznik może wprowadzać także elementy, które rozszerzają metodę opisaną w dokumentach bazowych.

Stosowanie metody punktów funkcyjnych zgodnie z Podręcznikiem wymaga znajomości dokumentów bazowych.

Załączniki do niniejszego dokumentu:

- Załącznik nr 1A ARiMR_model_IFPUG_przyklad.eap
- Załącznik nr 1B ARiMR_model_IFPUG_zmiana_przyklad.eap
- Załącznik nr 2 ARiMR_raport_z_wymiarowania_IFPUG.xlsx

Źródła wykorzystane w opracowaniu:

- www.ifpug.org - Podręcznik Metody w wersji 4.3.1 i inne materiały dostępne dla członków organizacji,
- www.nesma.org – materiały publiczne,
- M. A. Parthasarathy, *Practical Software Estimation*, Addison-Wesley & Infosys 2007, ISBN 0-321-43910-4

- M. Munds Schuh, C. Dekkers, The IT Measurement Compendium, Springer 2008, e-ISBN 978-3-540-68188-5
- Capers Jones, Estimating Software Costs, Bringing Realism To Estimating, Second Edition, The McGraw Hill Companies 2007, ISBN 13:978-0-07-148300-1, ISBN 10: 0-07-148300-4
- S. McConnell, Software Estimation: Demystifying the Black Art, Microsoftpress 2006, ISBN-13 978-0735605350

DEFINICJE

Funkcja transakcyjna – funkcjonalność oprogramowania, zapewniająca Użytkownikowi funkcjonalnemu, że jego wymagania w zakresie przetwarzania danych są spełnione. Funkcjami transakcyjnymi są w szczególności: Funkcja wejścia, Funkcja zapytania i Funkcja wyjścia.

Funkcja danych – cecha oprogramowania zapewniająca Użytkownikowi funkcjonalnemu, że jego wymagania w zakresie przechowywania danych wewnątrz jak i na zewnątrz oprogramowania są spełnione. Funkcje danych zarządzają Wewnętrznymi plikami danych (ILF) i Zewnętrznymi plikami danych (EIF).

DET (ang. Data Element Type) – unikalny, rozpoznawalny z punktu widzenia Użytkownika funkcjonalnego, niepowielony atrybut danych.

RET (ang. Record Element Type) – rozpoznawalna, z punktu widzenia Użytkownika funkcjonalnego, podgrupa (rekord) atrybutów (DETów) w ramach Funkcji danych.

Użytkownik funkcjonalny – osoba, system lub urządzenie, która/y w dowolnym momencie komunikuje się lub prowadzi interakcję z oprogramowaniem.

Wymaganie funkcjonalne użytkownika – wymaganie określające, co z punktu widzenia Użytkownika funkcjonalnego oprogramowanie ma robić, tzn. jaki cel Użytkownika funkcjonalnego ma oprogramowanie umożliwić.

Wymiarowanie – proces obejmujący wszystkie niezbędne czynności, do poprawnego określenia rozmiaru funkcjonalnego na podstawie Wymagań funkcjonalnych użytkownika oraz udokumentowania uzyskanych wyników wymiarowania zgodnie z niniejszym podręcznikiem

Strona przeprowadzająca wymiarowanie – zespół ekspertów, reprezentujący Podmiot, który na podstawie dokumentacji zarządczej projektu np. umowy, odpowiada za wykonanie Wymiarowania w całości.

Stan stabilny – punkt, w procesie przetwarzania danych przez oprogramowanie, w którym Wymaganie funkcjonalne użytkownika zostało w całości wykonane, jego cel został w pełni osiągnięty, nie pozostaje już nic do zrobienia, żeby go osiągnąć.

Przykład: Jeżeli Wymaganiem funkcjonalnym użytkownika jest wprowadzenie wniosku, to **stan stabilny** biznesu aplikacji zostanie osiągnięty, jeżeli wniosek zostanie w całości wprowadzony. Jeżeli system dostarcza użytkownikowi udogodnień, które pozwalają mu na przerwanie wprowadzania wniosku z zachowaniem wprowadzonych danych, a następnie wznowienie wprowadzania, to z punktu widzenia wymagania użytkownika, w momencie przerywania wprowadzania wniosku biznesu aplikacji nie będzie pozostawiony w stanie stabilnym, ponieważ cel wymagania nie został osiągnięty. Podobnie wprowadzenie części wniosku, np. pojedynczej sekcji, nie jest wystarczające do realizacji celu

użytkownika, po wprowadzeniu części wniosku, biznes aplikacji nie pozostaje w stanie stabilnym. Potrzebne są dodatkowe działania, które pozwolą na osiągnięcie celu użytkownika.

ZAKRES STOSOWANIA METODY PF IFPUG

Rozmiar funkcjonalny oprogramowania jest wyznaczany na podstawie określonego zbioru Wymagań funkcjonalnych użytkownika, które mają być realizowane przez dane oprogramowanie

Oprogramowanie mogą definiować także wymagania nie będące Wymaganiami funkcjonalnymi użytkownika (wymagania pozafunkcjonalne), wymagania te nie wpływają na rozmiar funkcjonalny oprogramowania.

Z punktu widzenia realizacji oprogramowania, wymagania pozafunkcjonalne definiują określony standard, który musi spełniać oprogramowanie realizujące Wymagania funkcjonalne użytkownika. Oznacza to, że oprogramowanie, oprócz dostarczania określonej funkcjonalności, musi spełniać określone standardy, zdefiniowane przez wymagania pozafunkcjonalne. O ile standard ten, jak to zostało wyżej powiedziane, nie wpływa na rozmiar funkcjonalny, to musi być wzięty pod uwagę, przy kalkulacji ceny punktu funkcyjnego.

Wymagania pozafunkcjonalne definiują standard oprogramowania w zakresie:

- dotrzymania SLA, związanego z wydajnością np. czasem przetwarzania, czy odpowiedzi oprogramowania,
- ergonomii użytkownika oprogramowania, zarówno w zakresie zapewnienia czytelnego i przejrzystego interfejsu użytkownika, jak i dostarczenia rozwiązań wspomagających zadania użytkownika, realizowane przy pomocy funkcjonalności oprogramowania, w tym między innymi:
 - wyświetlania komunikatów, informujących użytkownika o stanie przetwarzania danych,
 - wyświetlania informacji niezbędnej do podjęcia przez użytkownika właściwej decyzji, w tym odświeżanie danych,
 - potwierdzania krytycznych decyzji użytkownika,
 - możliwości anulowania realizowanych zadań,
 - możliwości przerwania, a następnie wznowienia pracy,
- bezpieczeństwa i audytowalności oprogramowania, w tym między innymi:
 - autentykacji użytkowników,
 - ograniczenia dostępu do funkcji i danych, do których użytkownik posiada uprawnienia,
 - rejestracji operacji wykonywanych przez użytkowników i zdarzeń systemowych,
- wykorzystania określonych technologii (np. w celu optymalnego wykorzystania będących w posiadaniu ARiMR zasobów),
- integracji z innymi systemami, w tym między innymi:
 - konfiguracji i formatu komunikatów udostępnianych przez system usług,
 - sposobu wymiany danych.

Wszelkie prace konieczne do realizacji Wymagań funkcjonalnych użytkownika lub zmian Wymagań funkcjonalnych użytkownika, zgodnie z opisanym wyżej standardem, traktuje się jako prace wliczone w cenę punktu funkcyjnego, w tym, między innymi:

- migrację danych niezbędną dla uruchomienia nowej / zmodyfikowanej funkcjonalności,
- utworzenie lub aktualizację danych w bazie danych na potrzeby działania nowej / zmodyfikowanej funkcjonalności,
- utworzenie lub aktualizację parametrów definiujących działanie aplikacji (w kontekście technologicznym jak i biznesowym) niezbędnych dla działania nowej / zmodyfikowanej funkcjonalności,
- utworzenie lub aktualizację słowników wspierających działanie aplikacji (w kontekście technologicznym jak i biznesowym) niezbędnych dla działania nowej / zmodyfikowanej funkcjonalności,
- optymalizację oprogramowania / struktur bazy danych, które pozwolą nowej/zmodyfikowanej funkcjonalności spełnić obowiązujące SLA,
- prace konfiguracyjne dotyczące elementów architektury, infrastruktury i oprogramowania standardowego, niezbędne dla uruchomienia nowej / zmodyfikowanej funkcjonalności,
- zaprojektowanie i zbudowanie interfejsu użytkownika, spełniającego przyjęte określone w projekcie normy, dla nowych / zmodyfikowanych funkcjonalności,
- zdefiniowanie i skonfigurowanie uprawnień do nowych / zmodyfikowanych funkcjonalności,
- przeprowadzenie analizy systemowej, w tym zaprojektowanie funkcjonalności, w taki sposób, aby spełniały przyjęte w projekcie rozwiązania, dla nowych / zmodyfikowanych funkcjonalności,
- opracowanie dokumentacji spełniającej określone w projekcie normy dla nowych / zmodyfikowanych funkcjonalności,
- przeprowadzenie testów zgodnie z przyjętymi w projekcie procedurami dla nowych / zmodyfikowanych funkcjonalności,

Przykład: W ramach modyfikacji przewidziano zmianę funkcjonalności rejestracji dokumentu. Aby zmiana była możliwa, konieczne jest dodanie nowych wartości do słownika typów dokumentu. Pracochłonność dodania tych wartości jest wliczona w cenę wynikającą z rozmiaru funkcjonalnego zmiany funkcjonalności rejestracji dokumentu.

Przykład: W ramach modyfikacji należy dodać kontrolę krzyżową wniosku. Wydajność kontroli musi spełniać przyjęte w projekcie wartości SLA. W tym celu konieczne jest dodanie nowych i optymalizacja istniejących struktur danych. Z optymalizacją wiąże się migracja danych wprowadzonych do tej pory wniosków, ze starych do nowych zoptymalizowanych struktur danych. Zarówno optymalizacja struktur jak i migracja danych jest wliczona w cenę wynikającą z rozmiaru funkcjonalnego, kontroli krzyżowej wniosku.

Przykład: W ramach modyfikacji należy zmienić funkcjonalność wprowadzania wniosku, poprzez rozszerzenie zakresu informacyjnego wniosku (dodanie nowych pól). Rozszerzenie zakresu informacyjnego powoduje, że zgodnie z przyjętymi standardami dotyczącymi interfejsu użytkownika, zakres wniosku trzeba rozmieścić na dwóch osobnych zakładkach (do tej pory cały zakres wniosku mieścił się na jednej). Dodanie nowej zakładki i rozmieszczenie pól na tych zakładkach jest wliczone w cenę wynikającą z rozmiaru funkcjonalnego, zmiany funkcjonalności wprowadzania wniosku.

Prace, których celem nie jest realizacja nowej funkcjonalności lub zmiana funkcjonalności, np. zmiana wyżej opisanego standardu globalnie lub dla wskazanych funkcjonalności powinny być wliczone w ryczałtową cenę opieki technicznej lub szacowane w oparciu o eksperckie szacunki pracochłonności w roboczościach. Prace te dotyczą w szczególności:

- wprowadzenia nowych lub zmiany elementów technologii, na której opiera się oprogramowanie, np. zmiana wersji oprogramowania standardowego.
- wprowadzenia nowych lub zmiany elementów architektury, np. zmiana sposobu pobierania danych z innego systemu,
- wprowadzenia nowych lub zmiany elementów infrastruktury, np. dodanie nowych macierzy dyskowych,
- Archiwizacja danych w celu optymalizacji wolumenu danych,
- optymalizacji oprogramowania / struktury bazy danych w wyniku podwyższenia SLA w zakresie wydajności,
- globalnej zmiany w zakresie interfejsu użytkownika (z zasady dotyczącej wszystkich lub wszystkich określonej kategorii funkcjonalności użytkownika),
- wygenerowanie zestawienia danych ad-hoc, bezpośrednio z bazy danych.

Sposób rozliczania prac, z uwzględnieniem ich rodzaju, jest określany w umowie pomiędzy ARiMR, a Wykonawcą.

Przykład: W ramach modyfikacji oprogramowania należy na interfejsie użytkownika danego raportu, prezentować dodatkowo liczbę odrzuconych decyzji. Zmiana ta dotyczy funkcjonalności raportu i można wyznaczyć jej rozmiar funkcjonalny.

Przykład: W ramach modyfikacji oprogramowania na interfejsie użytkownika danego raportu, liczba decyzji o przyznaniu płatności powinna być wyświetlana kolorem zielonym, a liczba decyzji o odrzuceniu wniosku, kolorem czerwonym. Zmiana ta dotyczy konkretnych atrybutów przekraczających granice oprogramowania, mieści się w ramach przyjętego standardu użytkownika, jest to zatem zmiana funkcjonalności raportu i można wyznaczyć jej rozmiar funkcjonalny.

Przykład: W ramach modyfikacji oprogramowania należy dostosować interfejs użytkownika do urządzeń mobilnych, np. wprowadzić interfejs responsywny. Zmiana ta zmienia „standard” oprogramowania, nie można wyznaczyć jej rozmiaru funkcjonalnego. Ewentualny koszt zmiany można wyznaczyć w oparciu o eksperckie szacowanie pracochłonności.

Przykład: W ramach modyfikacji oprogramowania należy dostosować interfejs użytkownika tylko dla funkcjonalności wyświetlania raportu. Zmiana ta nie dotyczy konkretnych atrybutów przekraczających granice oprogramowania, ponadto zmienia ona przyjęty standard interfejsu użytkownika, tyle że wyjątkowo dla jednej funkcjonalności. Nie można wyznaczyć rozmiaru funkcjonalnego tej zmiany. Ewentualny koszt zmiany można wyznaczyć w oparciu o eksperckie szacowanie pracochłonności.

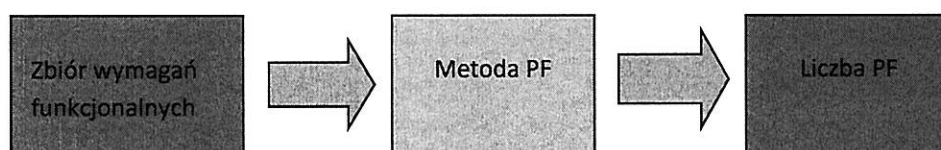
W ramach funkcjonalności, dla której ma zastosowanie metoda IFPUG zakłada się, iż koszt punktu funkcyjnego określony w umowie pomiędzy ARiMR i Wykonawcą zawiera w sobie koszt całego cyklu wytwórczego związanego z wytworzeniem oprogramowania o złożoności jednego punktu funkcyjnego.

ZASADY STOSOWANIA METODY PF IFPUG

WPROWADZENIE DO PUNKTÓW FUNKCYJNYCH

Metody punktów funkcyjnych to zbiory reguł, których zastosowanie do danego zbioru Wymagań funkcjonalnych użytkownika, pozwala wyznaczyć funkcjonalny rozmiar oprogramowania realizującego te wymagania. Oznacza to, iż na wejściu danej metody jest podawany zbiór wymagań a na wyjściu oczekiwana jest liczba punktów funkcyjnych wyznaczona dla tego zbioru wymagań.

Punkt funkcyjny jest abstrakcyjną miarą, definiowaną jako jednostka funkcjonalności biznesowej dostarczana przez mierzone oprogramowanie. Oznacza to, iż oprogramowanie o rozmiarze 100 punktów funkcyjnych dostarcza użytkownikowi 100 jednostkowych funkcjonalności biznesowych.

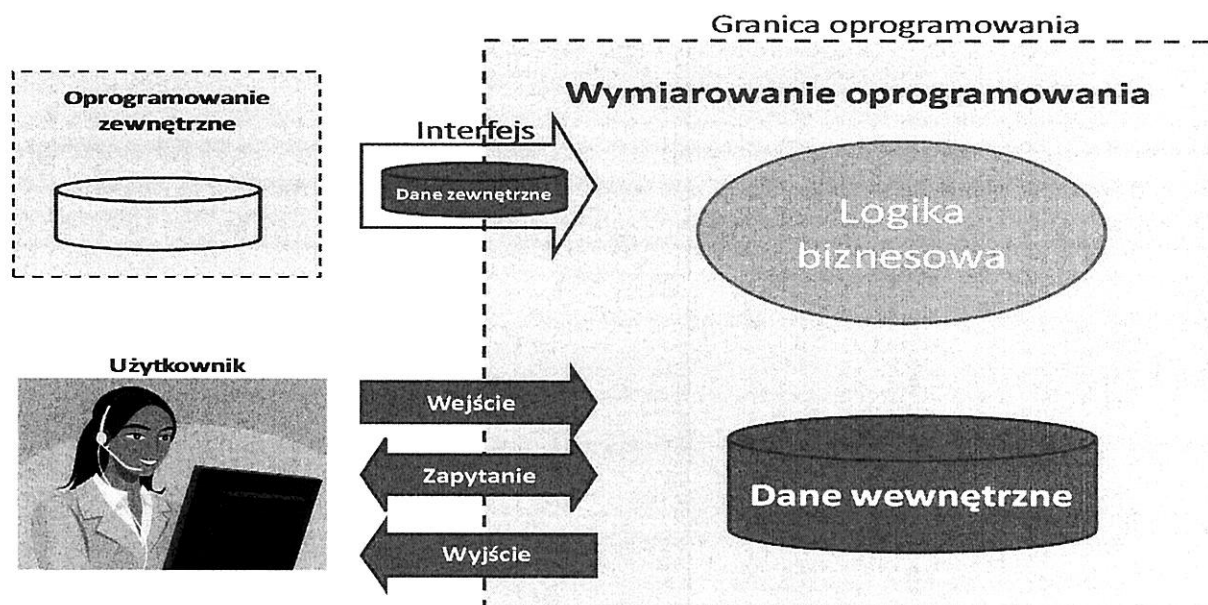


W przypadku zmian istniejącego oprogramowania, możemy mówić o rozmiarze samej zmiany, jak i rozmiarze oprogramowania po zmianie. W przypadku rozmiaru zmiany, liczba punktów funkcyjnych zmiany wynika wprost z wymagań definiujących daną zmianę. W przypadku rozmiaru oprogramowania po zmianie, liczba punktów funkcyjnych oprogramowania wynika z wymagań funkcjonalnych opisujących oprogramowanie po zmianie i może pozostać taka sama jak przed zmianą, może się zmniejszyć lub zwiększyć.

Przykład: W przypadku oprogramowania A o rozmiarze 100 PF zdefiniowano zmianę wprowadzającą nowe funkcje o rozmiarze 25 PF. Rozmiar zmiany to 25 PF a rozmiar oprogramowania A po zmianie to 125 PF. W przypadku oprogramowania B o rozmiarze 200 PF zdefiniowano zmianę modyfikującą istniejące funkcje - bez wprowadzania nowych. Rozmiar zmiany określono na 40 PF. Rozmiar oprogramowania B po zmianie to w dalszym ciągu 200 PF.

WSTĘP DO METODY PF IFPUG

Metoda Punktów Funkcyjnych IFPUG zakłada, iż rozmiar oprogramowania (wyrażony liczbą punktów funkcyjnych) może być wyznaczony w oparciu o analizę pięciu kluczowych elementów: funkcji wejścia, wyjścia, zapytań oraz wewnętrznego modelu danych i interfejsów do systemów zewnętrznych. Są to elementy z założenia wynikające z Wymagań funkcjonalnych użytkownika, a więc zrozumiałe z punktu widzenia Użytkownika funkcjonalnego. Elementy te **nie mogą** wynikać ze sposobu implementacji tych wymagań, czy użytej technologii. Jeżeli oprogramowanie ma realizować określony zbiór wymagań funkcjonalnych, to bez względu jak ono zostanie zrealizowane, będzie ono miało zawsze ten sam rozmiar funkcjonalny. Decyzje dotyczące użycia określonych bibliotek, oprogramowania systemowego, decyzje architektoniczne, liczby warstw itp. nie mogą wpływać na rozmiar funkcjonalny oprogramowania.



Powyższy rysunek przedstawia schemat wymiarowanego oprogramowania z punktu widzenia metody IFPUG. Wymiarowanie oprogramowania udostępnia użytkownikowi funkcje wyjścia / wejścia / zapytań, posiada swoją własną logikę biznesową definiującą sposób przetwarzania danych, przechowuje także określone dane wewnętrzne. Wymiarowane oprogramowanie korzysta z oprogramowania zewnętrznego przy pomocy określonego interfejsu, przy pomocy tego interfejsu ma dostęp do danych zewnętrznych. Zgodnie z metodą punktów funkcyjnych rozmiar oprogramowania wyraża złożoność logiki biznesowej aplikacji, którą później musi zrealizować odpowiedni kod źródłowy. Im bardziej złożona logika biznesowa tym większy rozmiar oprogramowania. Z punktu widzenia metody PF IFPUG złożoność Logiki biznesowej oprogramowania jest określaną przez dane wewnętrzne i dane zewnętrzne, które są przez tę logikę przetwarzane, a także funkcje wejścia, wyjścia oraz zapytania, które można przy pomocy tej logiki zrealizować.. Zbiór w/w elementów oprogramowania, czyli zbiór funkcji wejścia, wyjścia, zapytań, modelu danych wewnętrznych i zewnętrznych (na rysunku elementy te zostały oznaczone kolorem niebieskim) jest w dalszej części określany jako **model IFPUG**. Model IFPUG jest odzwierciedleniem wymagań funkcjonalnych użytkownika w postaci artefaktów definiowanych przez reguły metody PF IFPUG.

Metoda PF IFPUG wprowadza również pojęcie **granicy oprogramowania**. Granica oprogramowania wyznacza zakres szacowanego oprogramowania, jak i również służy do dookreślenia funkcji wejścia / wyjścia / zapytań jako funkcji przenoszących dane z zewnątrz do wewnątrz granicy lub z wewnątrz na zewnątrz granicy. Wewnętrzny model danych jest również traktowany jako mechanizm utrwalania danych wewnątrz granicy oprogramowania, podczas gdy interfejsy zewnętrzne są modelowane jako zbiory danych dostępne dla oprogramowania poza jego granicami.

Granica danego oprogramowania powinna być zgodnie z założeniami metody wyznaczona z punktu widzenia Użytkownika funkcjonalnego.. Wewnątrz granicy oprogramowania MUSI w całości znajdować się logika biznesowa oprogramowania, która w całości zrealizuje wymiarowany zbiór wymagań funkcjonalnych. Granica oprogramowania nie może być determinowana przez decyzje technologiczne, infrastrukturalne czy architektoniczne.

Przykład: Wymaganie polegające na wprowadzaniu danych z wniosku jest realizowane przy pomocy dwóch komponentów, jeden odpowiedzialny za dostarczenie interfejsu użytkownika, a drugi za

przeprowadzenie kontroli wprowadzonych danych, jednakże z punktu widzenia Użytkownika funkcjonalnego, architektura aplikacji nie jest istotna, i nie ma możliwości zdefiniowania wymagań oddzielnie dla każdego z tych komponentów. W takim wypadku komponenty te nie mogą być rozdzielone granicą oprogramowania, muszą leżeć w obrębie tej samej granicy oprogramowania, stanowią dwa, z potencjalnie wielu elementów wymiarowanego oprogramowania.

W przypadku szacowania zmian oprogramowania należy mieć na uwadze fakt, iż granica oprogramowania jest czymś niezależnym od zmiany – powinna być zdefiniowana dla danego zmienianego oprogramowania jako logicznej całości a nie, dla zmienianego fragmentu.

METODA PF IFPUG W ARIMR

Zgodnie z dokumentacją bazową procedura zastosowania metody PF IFPUG jest następująca:

- 1) Pozyskanie dostępnej dokumentacji opisującej funkcjonalność szacowanego oprogramowania lub zakres zmian funkcjonalności objęty szacowaną modyfikacją.
- 2) Określenie zakresu szacowania, granicy oprogramowania oraz identyfikacja wymagań funkcjonalnych .
- 3) Identyfikacja i pomiar funkcji danych
- 4) Identyfikacja i pomiar Elementarnych Procesów
- 5) Wyznaczenie rozmiaru funkcjonalnego
- 6) Udokumentowanie procesu wymiarowania

Z punktu widzenia szacowania projektów i modyfikacji w ARiMR kroki te są określane następująco:

Ad 1) Przez dostępną dokumentację rozumie się:

- a. w rozumieniu materiału będącego podstawą definiowania wymagań funkcjonalnych, które podlegają wymiarowaniu: specyfikację potrzeb funkcjonalnych użytkownika funkcjonalnego, opisującą zakres projektu lub modyfikacji
- b. w rozumieniu materiału pomocniczego, pozwalającego na właściwe zdefiniowanie wymagań funkcjonalnych, które będą podlegać wymiarowaniu, na podstawie potrzeb: dokumentację istniejącego oprogramowania ARiMR np. dokumentację analityczną, użytkownika itp..

Ad 2) Zakres wymiarowania wyznacza zakres projektu lub modyfikacji.

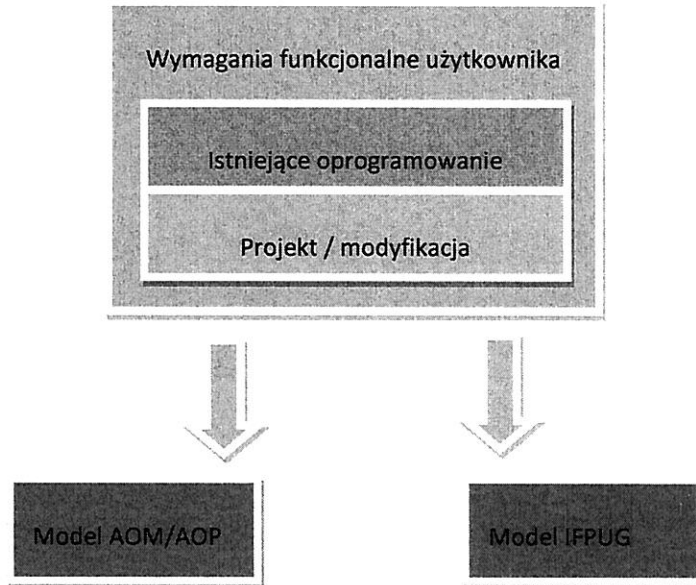
Granice szacowanego oprogramowania określone są przez ARiMR dla danego projektu lub umowy wykonawczej.

Wymagania funkcjonalne użytkownika, na podstawie których identyfikowane są elementy modelu IFPUG, są opracowywane przez Stronę przeprowadzającą wymiarowanie na podstawie dokumentacji, o której mowa w pkt. 1 oraz uzgodnień z ARiMR związanych ze sposobem realizacji potrzeb użytkownika. Wymagania muszą być unikatowe i jednoznacznie identyfikowane, a wszelkie uzgodnienia muszą zostać udokumentowane np. w postaci notatki ze spotkania. Przed przystąpieniem do kolejnego kroku wymiarowania Wymagania funkcjonalne użytkownika, muszą być zatwierdzone przez ARiMR.

Ad 3, 4, 5, 6) Zadania te w całości leżą po Stronie przeprowadzającej wymiarowanie.

Ad 5) Przy wyznaczaniu rozmiaru nie stosuje się współczynników VAF.

Zbiór Wymagań funkcjonalnych użytkownika dla danego oprogramowania jest nadrzędny zarówno dla modelu IFPUG jak i dla modelu AOM/AOP. Oznacza to również, że elementy modelu IFPUG, nie stają się automatycznie elementami modelu AOM/AOP i odwrotnie. Łącznikiem tych dwóch modeli są Wymagania funkcjonalne użytkownika, z którym muszą wynikać zarówno określone elementy modelu IFPUG jak i modelu AOM/AOP.



Powyższe oznacza, iż elementów modelu IFPUG nie należy wyprowadzać z istniejących elementów modelu AOM/AOP, ale wprost z ze zbioru wymaganiach funkcjonalnych użytkownika.

Jeżeli dla danej modyfikacji zakres jest opisany wyłącznie poprzez model analityczny, w pierwszej kolejności na podstawie analizy odtwarzany jest pierwotny zbiór Wymagań funkcjonalnych użytkownika i dopiero na podstawie odtworzonego zbioru identyfikowane są elementy modelu IFPUG.

Z punktu widzenia szacowania, zbiór Wymagań funkcjonalnych użytkownika powinien zawierać wymagania na takim poziomie szczegółowości by było możliwe zidentyfikowanie elementów modelu, IFPUG czyli w szczególności, ale nie tylko:

- 1) Wymagania powinny opisywać konkretne cele, jakie użytkownik chce osiągnąć przy pomocy oprogramowania aczkolwiek nie jest niezbędne wskazywanie sekwencji kroków interakcji z oprogramowaniem, jakie muszą być przeprowadzone oraz reguł przetwarzania danych w tych krokach.
- 2) Wymagania powinny określać zakres danych, jakie ma przechowywać oprogramowanie aczkolwiek nie jest niezbędne wskazywanie typów, formatów, zasad walidacji i wyliczeń tych danych.
- 3) Wymagania powinny określać zakres danych wprowadzanych do systemu oraz wyprowadzanych do systemu przy realizacji określonych celów użytkownika, przy czym nie jest konieczne wskazywanie typów, formatów, zasad walidacji i wyliczeń tych danych.

Jeżeli zbiór Wymagań funkcjonalnych użytkownika podlegający wymiarowaniu nie spełnia powyższych kryteriów, to konieczne jest przeprowadzenie uzgodnień, które uszczegółowią go w taki sposób, aby spełniał w/w wymagania.

Powyższy minimalny zakres Wymagań funkcjonalnych użytkownika jest zakresem zdefiniowanym na potrzeby szacowania metodą IFPUG. Nie wyklucza to jednak oczywiście innych potrzeb, dla których te wymagania powinny być rozszerzone o dodatkowe elementy.

Jednym ze sposobów specyfikacji Wymagań funkcjonalnych użytkownika, o którym mowa powyżej, jest specyfikacja w postaci:

- zidentyfikowanych przypadków użycia celu użytkownika,
- zidentyfikowanych klas modelu dziedziny (logicznego modelu danych), które posiadają zidentyfikowane atrybuty,
- dziedziny opisu określającego, z których atrybutów klas logicznego modelu danych korzystają przypadki użycia

Pojęcie „zidentyfikowane” oznacza tu, iż te elementy są wyodrębnione i nazwane, natomiast nie jest wymagany opis sposobu ich realizacji, w tym: reguły, przebiegi, formaty itp. .

METODA PF IFPUG – ZASADY

Metoda IFPUG definiuje następujące artefakty jako elementy modelu IFPUG:

- **Elementarny proces (EP – Elementary Process)** – minimalna jednostka aktywności oprogramowania wynikająca z Wymagań funkcjonalnych użytkownika (Elementarny proces może wynikać z wielu Wymagań funkcjonalnych użytkownika, a także wiele Elementarnych Procesów może wynikać z jednego Wymagania funkcjonalnego użytkowników), która spełnia jednocześnie następujące kryteria:
 - jest zrozumiała z punktu widzenia Użytkownika funkcjonalnego,
 - w pełni realizuje transakcję,
 - w całości zawiera się w sobie, tzn. nie występują poza nią żadne dodatkowe czynności niezbędne do jej zainicjowania czy pełnego wypełnienia,
 - jej realizacja pozostawia biznes aplikacji w Stanie stabilnym, tzn. osiąga punkt, w którym Wymaganie funkcjonalne użytkownika zostało w całości wykonane, jego cel został w pełni osiągnięty, nie pozostaje już nic do zrobienia, żeby go osiągnąć.
- **Funkcja wejścia (EI – External Input)** – Elementarny proces przetwarzający lub kontrolujący informacje przychodzące z zewnątrz granicy oprogramowania.
- **Funkcja zapytania (EQ – External Inquiry)** – Elementarny proces przetwarzający lub kontrolujący informacje przesyłane na zewnątrz granicy oprogramowania.
- **Funkcja wyjścia (EO – External Output)** – Elementarny proces przetwarzający lub kontrolujący informacje przesyłane na zewnątrz granicy oprogramowania, obejmujący logikę przetwarzania, wykraczającą poza logikę Funkcji zapytania (EQ).
- **Plik danych (FTR – File Type Referenced)** – funkcja danych odczytywana lub zarządzana przez funkcję transakcyjną.
- **Wewnętrzny plik danych (ILF – Internal Logical File)** – rozpoznawana przez użytkownika grupa logicznie powiązanych danych, zarządzana wewnątrz granicy wymiarowanego oprogramowania
- **Zewnętrzny plik danych (EIF – External Interface File)** – rozpoznawana przez użytkownika grupa logicznie powiązanych danych, do którego odwołuje się wymiarowane oprogramowanie, lecz które są zarządzane poza granicami wymiarowanego oprogramowania, przez oprogramowanie zewnętrzne.

Przykład: W przypadku, gdy oprogramowanie umożliwia rejestrowanie dokumentów kancelaryjnych to funkcja rejestracji nowego dokumentu będzie elementarnym procesem typu EI, funkcja prezentująca raport z zarejestrowanych dokumentów będzie elementarnym procesem typu EQ, funkcja prezentująca statystykę tempa rejestracji dokumentów będzie elementarnym procesem typu EO gdyż wylicza dodatkowe dane, nieprzechowywane w aplikacji; zbiór danych przechowujący dane zarejestrowanych dokumentów będzie plikiem danych typu ILF a zbiór danych przechowujący dane klientów organizacji (zarządzanym w ramach innego oprogramowania) będzie plikiem danych typu EIF wykorzystywanym celem identyfikacji nadawcy dokumentu.

W celu wyznaczenia rozmiaru funkcjonalnego oprogramowania konieczne jest określenie dla definiującego to oprogramowanie zbioru Wymagań funkcjonalnych użytkownika wszystkich wyżej zdefiniowanych elementów modelu IFPUG tj. Elementarnych procesów EI, EO i EQ oraz Plików danych ILF i EIF, czyli przygotowanie modelu IFPUG danego oprogramowania. Przygotowanie modelu IFPUG oprogramowania wchodzi w zakres Wymiarowania.

W celu wyznaczenia rozmiaru funkcjonalnego zmiany oprogramowania konieczne jest zidentyfikowanie elementów w istniejącym modelu IFPUG oprogramowania, które ulegają zmianie lub które są usuwane, oraz zidentyfikowanie nowych elementów modelu IFPUG, których utworzenie wynika z Wymagań funkcjonalnych użytkownika, określających zakres modyfikacji, czyli przygotowanie modelu IFPUG modyfikacji. Przygotowanie modelu IFPUG modyfikacji wchodzi w zakres Wymiarowania.

W przypadku Wymiarowania zmiany, kiedy nie istnieje model IFPUG, dla oprogramowania, które jest modyfikowane, konieczne jest w pierwszym kroku przygotowanie modelu IFPUG dla oprogramowania przed zmianą. Przygotowanie w/w modelu IFPUG dla oprogramowania przed zmianą, wchodzi w zakres Wymiarowania danej zmiany.

IDENTYFIKACJA PLIKÓW DANYCH IFPUG

Celem określenia Plików danych, należy zidentyfikować grupy logicznie powiązanych danych z punktu widzenia Wymagań funkcjonalnych użytkownika. Dodatkowo, dane od siebie zależne należy łączyć w logiczne Pliki danych. Dane zależne to dane powiązane ze sobą relacją kompozycji, czyli:

- dane zależne istnieją wyłącznie, jeśli istnieją dane nadrzędne,
- usunięcie danych nadrzędnych oznacza usunięcie danych zależnych.

Przykład: Aplikacja do rejestracji dokumentów wpływających do organizacji opcjonalnie korzysta z systemu CRM celem identyfikacji nadawców. Dane podmiotów – nadawców z systemu CRM, z punktu widzenia szacowania aplikacji do rejestracji dokumentów, są modelowane jako plik danych EIF.

Dla każdego pliku danych należy określić jego unikalne atrybuty (DET – Data Element Type).

Atrybut pliku danych to każdy rozpoznawalny przez użytkownika, unikalny atrybut pliku danych, który jest zarządzany, zapisywany, aktualizowany lub odczytywany w ramach elementarnych procesów szacowanego oprogramowania. Atrybutami są również atrybuty związków danego Pliku danych z innymi Plikami danych, przy czym w zależności od nawigowalności tego związku, atrybut związku jest określany w jednym lub w obu powiązanych plikach danych.

Mając określone atrybuty poszczególnych plików danych należy zidentyfikować w ramach plików danych rekordy (RET – Record Element Type). Rekord grupuje w ramach pliku danych atrybuty które występują w liczności innej niż 1:1 do pliku danych (inaczej mówiąc są one w relacji kompozycji). Domyślnie każdy plik danych ma jeden rekord.

Przykład: Zbiór danych faktura to dane faktury plus lista pozycji faktury. Pozycja faktury jest elementem pliku danych Faktura i jest modelowana jako rekord w tym pliku danych. Pozycja faktury nie istnieje bez faktury, usunięcie faktury usuwa wszystkie powiązanie z nią pozycje faktury.

Przykład: Zbiór danych wniosek, posiada strukturę drzewiastą. Składa się z sekcji A, B, C i D, każda zawiera unikatowy zestaw atrybutów, sekcja A zawiera listę LA, która zawiera listę elementów posiadających unikatową strukturę, sekcja C również zawiera listę LB, przy czym elementy listy LB także zawierają listę elementów LLB. Cała opisana wyżej struktura to jeden Plik danych (ILF), który zawiera 4 rekordy (RET), jeden odpowiada wnioskowi i jego sekcjom (sekcje, które występują w relacji 1:1 do wniosku, należy traktować tylko jako element porządkujący atrybuty we wniosku, dlatego sekcji tych nie należy traktować jako odrębne RETy), kolejne elementom listy LA i LB, i ostatni najbardziej zagnieżdżonemu elementowi listy LLB.

Poziom zagnieżdżenia relacji wzajemnie ze sobą powiązanych danych, czy też cykliczność tych relacji nie ma wpływu na liczbę identyfikowanych Plików danych (ILF) czy też rekordów (RET).

Przykład: Zbiór danych składa się z danych osób, połączonych ze sobą relacją oznaczającą, że jedna osoba może reprezentować drugą, przy czym relacja ta nie posiada żadnych atrybutów tzn. jeżeli występuje między osobą A, a osobą B, to znaczy, że osoba A może reprezentować osobę B, a jeżeli nie występuje, to nie może. W rezultacie zbiór może tworzyć sieć powiązanych ze sobą osób. W takim wypadku zbiór taki identyfikujemy jako jeden Plik danych (ILF), posiadający jeden rekord (RET).

IDENTYFIKACJA MODELU FUNKCJONALNEGO IFPUG

Celem określenia Elementarnych procesów należy zidentyfikować funkcje minimalne, niezależne i odrębne od innych funkcji zidentyfikowanych jako Elementarne procesy, i które w pełni realizują cel użytkownika.

W ramach jednego Elementarnego procesu łączy się funkcje posiadające własną logikę biznesową, które są wariantami lub alternatywnymi przebiegami realizacji określonego celu użytkownika.

Różne warianty logiki biznesowej mogą wynikać m.in. z:

- różnych decyzji użytkownika,
- różnych wprowadzonych danych (np. wprowadzona data jest wcześniejsza lub późniejsza od bieżącej, użytkownik podał dane kompletne lub niekompletne),
- różnych warunków uruchomienia Elementarnego procesu (np. różne uprawnienia użytkowników),
- różnych stanów systemu (np. występowania lub niewystępowania określonych danych utrwalonych w granicach oprogramowania),
- różnych sposobów porządkowania i prezentowania danych np. ukrywania niewymaganych pól, prezentacji w sekcjach.

Różne warianty logiki biznesowej mogą korzystać z różnych Plików danych lub różnych atrybutów.

Tabela 1. Różne formy logiki biznesowej – składniki Elementarnego procesu, które nie powinny być identyfikowane jako odrębne Elementarne procesy.

Lp.	Forma logiki biznesowej
1	Walidacja danych Przykład: Przy wprowadzaniu nowego wniosku sprawdzane jest czy wszystkie pola obowiązkowe są wypełnione.
2	Wykonywanie obliczeń matematycznych Przykład: Przy generowaniu raportu obliczana jest średnia wielkość deklarowanej powierzchni, na podstawie wszystkich zatwierdzonych wniosków.
3	Przekształcanie wartości Przykład: Na podstawie deklarowanej powierzchni, wyznaczana jest grupa, do której należy wniosek, gdzie przynależność do grupy jest definiowana przez przedział deklarowanej powierzchni.
4	Filtrowanie i selekcjonowanie danych wg określonych kryteriów Przykład: Na podstawie podanego przedziału dat wyznaczany jest ze wszystkich wniosków, zbiór wniosków, których data złożenia mieści się w przedziale.
5	Sprawdzenie czy spełnione są warunki Przykład: Oprogramowanie inaczej przetwarza wnioski, które zostały złożone w przyjętym okresie składania wniosków, a inaczej wniosku złożone po tym okresie. Oprogramowanie sprawdza, czy spełnione zostały warunki, określonego wariantu przetwarzania wniosku i uruchamia go. Przykład: Oprogramowanie inaczej prezentuje dane użytkownikom, którzy pierwszy raz wykonują daną operację, niż użytkownikom, którzy wykonują ją po raz kolejny. Oprogramowanie sprawdza, czy użytkownik po raz pierwszy wykonuje daną operację, jeżeli tak to uruchamia odpowiedni wariant prezentacji danych.

6	Aktualizacja jednego lub więcej Plików wewnętrznych (ILF) Przykład: Oprogramowanie zapisuje wprowadzone dane wniosku.
7	Odwołanie się do jednego lub więcej Plików wewnętrznych (ILF) lub zewnętrznych (EIF) Przykład: Oprogramowanie wywołuje usługę, w celu sprawdzenia, czy w systemie zewnętrznym występuje podmiot posiadający określony nr NIP (EIF).
8	Pobranie danych lub informacji kontrolnych Przykład: W celu prezentacji listy wniosków, dane wniosków są odczytywane.
9	Utworzenie danych poprzez transformacje istniejących danych Przykład: Znak sprawy jest tworzony przez system na podstawie typu dokumentu inicjującego sprawę i daty utworzenia sprawy
10	Zmiana zachowania aplikacji Przykład: Kontrola wniosku może być przeprowadzona albo w trybie zwykłym albo uproszczonym, tryb kontroli cechuje się specyficznym zestawem czynności kontrolnych. Proces kontroli wniosku zostanie zmieniony, jeżeli zmienimy typ kontroli wniosku, ze zwykłego na uproszczony.
11	Przygotowanie danych w celu ich prezentacji poza granicą oprogramowania Przykład: Lista prezentowanych wniosków jest odpowiednio formatowana i wyświetlana użytkownikowi
12	Umożliwienie przyjęcia danych do spoza granicy oprogramowania Przykład: System wyświetla formatkę, która umożliwia użytkownikowi wprowadzanie i zapisanie danych wniosku
13	Sortowanie i porządkowanie danych Przykład: System sortuje wnioski wg daty złożenia wniosku Przykład: System grupuje wnioski wg właściwości miejscowej

Przykład: System umożliwia rejestrację danych podmiotów, przy czym rejestrowane są zarówno podmioty jako osoby fizyczne, jak i prawne. Elementarny proces EI Rejestracja podmiotu obsługuje oba te przypadki a zbiór atrybutów tego procesu jest łącznym zbiorem wszystkich atrybutów osób prawnych i fizycznych.

Przykład: System umożliwia weryfikację wniosku, poprzez wprowadzanie wyników weryfikacji, przy czym raz jako użytkownik, który ma węższe, a raz jako użytkownik, który ma szersze uprawnienia, co powoduje, że dla użytkownika, który posiada węższe uprawnienia nie są dostępne pewne opcje. Jeden Elementarny proces EI obejmuje zarówno weryfikację wykonywaną przez użytkownika o węższych uprawnieniach, jak i użytkownika o szerszych uprawnieniach.

Przykład: System umożliwia weryfikację wniosku, poprzez wprowadzanie wyników weryfikacji, przy czym jeżeli weryfikacja miała już wcześniej miejsce, to system wyświetla dodatkowo informacje dotyczące wcześniejszych weryfikacji. Zarówno weryfikacja wykonywana po raz pierwszy, jak i weryfikacja wykonywana za każdym następnym razem, to ten sam Elementarny proces EI.

Zidentyfikowane elementarne procesy należy sklasyfikować przypisując im jeden z trzech określonych przez metodę typów tj. EI, EO lub EQ.

Elementarny proces jest procesem typu EI, jeśli:

- 1) Zawiera logikę przetwarzania walidującą dane wprowadzane i utrwalane w granicach oprogramowania w ramach tego procesu, lub
- 2) Jego podstawowym zadaniem jest zarządzanie (dodawanie, usuwanie, modyfikacja) jednym lub więcej plikami danych w ramach procesu, lub
- 3) Jego podstawowym zadaniem jest modyfikacja zachowania oprogramowania po zakończeniu realizacji procesu (poprzez modyfikację określonych plików danych).

Przykład: Funkcja rejestracji dokumentu kancelaryjnego jest procesem EI gdyż umożliwia wprowadzenie do oprogramowania danych rejestrowanego dokumentu.

Elementarny proces jest procesem typu EO, jeśli jego podstawowym zadaniem jest prezentacja danych użytkownikowi z wykorzystaniem przynajmniej jednego ze sposobów przetwarzania danych:

- 1) Obliczeń matematycznych.
- 2) Aktualizacji pliku lub plików ILF.
- 3) Utworzenia prezentowanych danych.
- 4) Modyfikacja zachowania oprogramowania po zakończeniu realizacji procesu (poprzez modyfikację określonych plików danych).

Przykład: Funkcja utworzenia raportu z zarejestrowanych dokumentów kancelaryjnych jest procesem typu EO, gdyż zawiera informacje o liczbie zarejestrowanych dokumentów z podziałem na stanowiska rejestracji.

Elementarny proces jest procesem typu EQ, jeśli jego zadaniem jest prezentacja informacji utrwalonych w granicach oprogramowania i nie zawiera przetwarzania danych wskazanego dla procesów typu EO.

Przykład: Funkcja wyszukiwania dokumentów kancelaryjnych jest procesem typu EQ, gdyż prezentuje wyłącznie dane dokumentów zapisane w bazie.

Dla każdego zidentyfikowanego i sklasyfikowanego elementarnego procesu należy następnie określić, z jakich plików danych ten proces korzysta (odczyt lub zapis) oraz jakie atrybuty są w ramach procesu wykorzystywane.

Jako atrybuty wykorzystywane przez elementarny proces należy traktować:

- unikalne i rozpoznawalne przez użytkownika atrybuty danych przekraczające granicę systemu podczas realizacji procesu,
- jeden atrybut, jeśli elementarny proces może prezentować użytkownikowi komunikaty (proste błędy / potwierdzenia) związane z realizacją procesu – niezależnie od liczby tych komunikatów,

Nie należy traktować jako atrybutów elementarnych procesów:

- opisów na formatach, nagłówek kolumn, nazw pól,
- znaczników wersji, znaczników typów, znaczników czasowych nie mających znaczenia biznesowego (tj. wpływającego na procesy biznesowe użytkownika),
- numerowania stron, wierszy, liczb porządkowych w tabelach,
- elementów nawigacyjnych interfejsu użytkownika, które nie są jednocześnie atrybutami przekraczającymi granicę oprogramowania

- elementów wspomagających wprowadzanie danych na interfejsie użytkownika,
- atrybutów generowanych w ramach realizacji procesu, ale nie przekraczających granicy oprogramowania,
- atrybutów odczytywanych z plików danych nieprzekraczających granicy oprogramowania.

Przykład: Raport z liczby zarejestrowanych dokumentów kancelaryjnych jest tworzony na zadany dzień i prezentuje liczbę dokumentów zarejestrowanych w poszczególnych stanowiskach rejestracyjnych. Na raporcie jest też prezentowana wersja formularza raportu oraz godzina wygenerowania. Z punktu widzenia użytkownika te dwa ostatnie parametry są nieistotne - są to parametry techniczne dostawcy oprogramowania kancelaryjnego. Tym samym, jako atrybuty procesu zalicza się atrybut dnia, na jaki jest tworzony raport, liczbę raportów i identyfikacje stanowiska. Dodatkowo mogą być również zliczone przycisk wykonania raportu na zadane parametry (jako realizacja akcji tego procesu) oraz ew. 1 DET dla komunikatu o błędnych parametrach.

KLASYFIKACJA ELEMENTARNYCH PROCESÓW

Główne cele Elementarnych procesów w zależności od ich klasyfikacji:

Główny cel	EI	EO	EQ
Modyfikacja zachowania oprogramowania	Główny cel	Cel wspomagający	Niedozwolone
Zarządzanie plikami danych ILF	Główny cel	Cel wspomagający	Niedozwolone
Prezentacja danych użytkownikowi	Cel wspomagający	Główny cel	Główny cel

Możliwe logiki przetwarzania danych w zależności od klasyfikacji Elementarnego procesu:

Logika przetwarzania	EI	EO	EQ
1. Przeprowadzane są walidacje atrybutów danych	T	T	T
2. Realizowane są obliczenia matematyczne na atrybutach danych	T	O*	N
3. Wartości są konwertowane pomiędzy różnymi systemami miar, jednostek lub z wykorzystaniem tablic konwersji	T	T	T
4. Dane są filtrowane i wybierane w oparciu o określone kryteria	T	T	T
5. Weryfikowane są reguły przetwarzania danych lub warunki biznesowe przeprowadzanie danej akcji	T	T	T
6. Jeden lub więcej plik ILF jest zaktualizowany	O*	O*	T
7. Jeden lub więcej plików ILF lub EIF jest odczytywany	T	T	O
8. Dane są odczytywane	T	T	O

9. Dane są tworzone	T	O*	N
10. Zachowanie oprogramowania jest zmienione po realizacji procesu	O*	O*	N
11. Dane są odczytywane i prezentowane poza granicą oprogramowania	T	O	O
12. Dane są wprowadzane i wprowadzane do granicy aplikacji (ale nie utrwalane)	O	T	T
13. Dane są sortowane lub przetwarzane pod kątem sposobu prezentacji	T	T	T

- T – elementarny proces może realizować daną logikę przetwarzania
- N – elementarny proces nie może realizować danej logiki przetwarzania
- O – elementarny proces musi realizować daną logikę przetwarzania
- O* – elementarny proces musi realizować przynajmniej jedną z logik oznaczoną dla niego jako O* (w ramach kolumny)

Przykład:

Dla oprogramowania zdefiniowano elementarny proces EP1 Rejestracja przyjęcia dokumentu. Proces ten zakwalifikowano jako EI, gdyż jego głównym zadaniem jest zarządzanie (poprzez wprowadzanie) dokumentami kancelaryjnymi. Lista kontrolna dla logik przetwarzania tego procesu jest poniżej. Z obowiązkowych logik, tj. 6, 10 i 12, proces EP1 realizuje dwie, przy czym logika 12 zawsze musi wystąpić dla procesu typu EI a z logik 6 i 10 przynajmniej jedna co jest spełnione.

Logika przetwarzania	EI	Weryfikacja
1. Przeprowadzane są walidacje atrybutów danych	T	Tak - EP1 przeprowadza walidacje
2. Realizowane są obliczenia matematyczne na atrybutach danych	T	Nie - EP1 nie przeprowadza wyliczeń
3. Wartości są konwertowane pomiędzy różnymi systemami miar, jednostek lub z wykorzystaniem tablic konwersji	T	Nie - EP1 nie przeprowadza konwersji
4. Dane są filtrowane i wybierane w oparciu o określone kryteria	T	Nie - EP1 nie posiada filtracji i wybierania
5. Weryfikowane są reguły przetwarzania danych lub warunki biznesowe przeprowadzanie danej akcji	T	Tak - EP1 przeprowadza weryfikacje reguł biznesowych
6. Jeden lub więcej plik ILF jest zaktualizowany	O*	Tak - aktualizowany jest zbiór ILF dokumentów kancelaryjnych
7. Jeden lub więcej plików ILF lub EIF jest odczytywany	T	Tak - odczytywany jest opcjonalnie EIF z danymi nadawców

8. Dane są odczytywane	T	Tak - dane są odczytywane
9. Dane są tworzone	T	Tak - tworzony jest nowy obiekt
10. Zachowanie oprogramowania jest zmienione po realizacji procesu	O*	Nie - zachowanie oprogramowania się nie zmienia
11. Dane są odczytywane i prezentowane poza granicą oprogramowania	T	Tak - część danych odczytanych z systemu jest prezentowana na formularz
12. Dane są wprowadzane i wprowadzane do granicy aplikacji (ale nie utrwalane)	O	Tak - użytkownik wprowadza dane przez formularz rejestracji
13. Dane są sortowane lub przetwarzane pod kątem sposobu prezentacji	T	Nie - proces EP1 nie sortuje danych.

WYLICZENIE ROZMIARU FUNKCJONALNEGO ELEMENTÓW MODELU IFPUG

Rozmiar funkcjonalny plików danych jest wypadkową liczby atrybutów i rekordów w danym pliku. Rozmiar funkcjonalny elementarnych procesów jest wypadkową liczby atrybutów procesu i liczby wykorzystywanych plików danych. Dla każdego z typów artefaktów modelu to jest zdefiniowana odpowiednia tabela wyliczeniowa.

ILF	DET 1 - 19	DET 20 - 50	DET > 50
RET 1	LOW (7)	LOW (7)	AVERAGE (10)
RET 2 - 5	LOW (7)	AVERAGE (10)	HIGH (15)
RET > 5	AVERAGE (10)	HIGH (15)	HIGH (15)
EIF	DET 1 - 19	DET 20 - 50	DET > 50
RET 1	LOW (5)	LOW (5)	AVERAGE (7)
RET 2 - 5	LOW (5)	AVERAGE (7)	HIGH (10)
RET > 5	AVERAGE (7)	HIGH (10)	HIGH (10)
EI	DET 1 - 4	DET 5 - 15	DET > 15
FTR < 2	LOW (3)	LOW (3)	AVERAGE (4)
FTR 2	LOW (3)	AVERAGE (4)	HIGH (6)
FTR > 2	AVERAGE (4)	HIGH (6)	HIGH (6)
EQ	DET 1 - 5	DET 6 - 19	DET > 19
FTR < 2	LOW (3)	LOW (3)	AVERAGE (4)
FTR 2	LOW (3)	AVERAGE (4)	HIGH (6)
FTR > 2	AVERAGE (4)	HIGH (6)	HIGH (6)
EO	DET 1 - 5	DET 6 - 19	DET > 19
FTR < 2	LOW (4)	LOW (4)	AVERAGE (5)
FTR 2 - 3	LOW (4)	AVERAGE (5)	HIGH (7)
FTR > 3	AVERAGE (5)	HIGH (7)	HIGH (7)

Suma rozmiarów funkcjonalnych wszystkich elementów modelu IFPUG jest rozmiarem oprogramowania w punktach funkcyjnych.

Przykład: Plik danych ILF o 25 atrybutach i dwóch rekordach ma złożoność 10 punktów funkcyjnych. Elementarny proces EQ o 3 atrybutach i jednym wykorzystywanym pliku FTR ma złożoność 3 punkty funkcyjne.

WYLICZANIE ROZMIARU FUNKCJONALNEGO ZMIAN

Do wyliczania rozmiaru funkcjonalnego zmian stosuje się metodę PF IFPUG z rozszerzeniem opracowanym przez organizację NESMA.

W przypadku zmiany oprogramowania wyróżnia się trzy różne rodzaje zmian:

- dodanie nowej funkcjonalności,
- usunięcie istniejącej funkcjonalności,
- modyfikacja istniejącej funkcjonalności.

W przypadku dodawania i usuwania funkcjonalności rozmiar funkcjonalny tych zmian wyznacza się jako rozmiar dodawanych lub usuwanych elementów modelu IFPUG, przy czym w przypadku usuwanych elementów stosuje współczynnik wpływu 0.40. Rozmiar funkcjonalny usuwanego elementu modelu jest iloczynem jego rozmiaru funkcjonalnego i podanego współczynnika wpływu. Powyższy współczynnik dla usuwanych elementów nie jest stosowany w przypadku wyłączenia całej aplikacji lub całego modułu z eksploatacji i utrzymania, dla takich przypadków stosuje się wycenę ekspercką.

W przypadku modyfikacji istniejącej funkcjonalności stosuje się dodatkowe reguły, większość z nich wynika z opracowania organizacji NESMA wspomnianego wcześniej w dokumencie.

WYLICZANIE ZMIAN PLIKÓW DANYCH

Plik danych jest uznawany za zmieniony, jeśli są do niego dodane atrybuty, odjęte atrybuty, zmieniony jest typ atrybutu lub zmieniona jest struktura rekordów. Inną możliwą zmianą pliku danych jest zmiana jego typu (EIF / ILF).

Celem wyliczenia rozmiaru zmiany pliku danych należy wyliczyć rozmiar funkcjonalny pliku danych po zmianie oraz wskazać liczbę zmienianych atrybutów. Mając te dane wylicza się procentowy wskaźnik zmiany. Ten wskaźnik służy do wyliczenia wskaźnika wpływu zmiany. Iloczyn wskaźnika wpływu zmiany oraz rozmiaru pliku po zmianie wyznacza rozmiar funkcjonalny zmiany pliku danych.

Procentowy wskaźnik zmiany: (liczba zmienionych atrybutów / liczba atrybutów przed zmianą) * 100%

Wskaźnik wpływu wyznacza się w oparciu o poniższą tabelę:

Procentowy wskaźnik zmiany	$p \leq 30$ %	$30\% < p \leq 60$ %	$60\% < p \leq 100\%$	$p > 100\%$
Wskaźnik wpływu	0.25	0.50	0.75	1.00

W przypadku, gdy zmianie ulegnie typ pliku danych lub struktura rekordów lub jedno i drugie to wskaźnik wpływu dla tej modyfikacji ma wartość 0.40. W przypadku, gdy jednocześnie uległy zmianie atrybuty, to jako wskaźnik wpływu, przyjmuje się wartość większą spośród wskaźnika wynikającego z modyfikacji typu lub struktury rekordów oraz wskaźnika wynikającego ze zmian atrybutów.

W przypadku, gdy dwa pliki danych są łączone w jeden to traktuje się to jako usunięcie dwóch plików i dodanie jednego nowego. W przypadku, gdy istniejący plik jest dzielony, to traktuje się to jako usunięcie jednego pliku i dodanie dwóch nowych.

Przykład: Plik danych ILF A posiada 15 atrybutów i jeden rekord. W ramach modyfikacji jest dodawanych kolejnych pięć a dodatkowo dla jednego istniejącego jest zmieniany typ danych. Złożoność pliku przed modyfikacją to 7 PF. Procentowy wskaźnik zmiany to $(6/15)*100\% = 40\%$.

Wskaźnik wpływu to 0.50. Złożoność pliku po modyfikacji to 10 PF. Złożoność zmiany to 10 PF * 50% = 5 PF.

WYLICZANIE ZMIAN ELEMENTARNYCH PROCESÓW

Elementarny proces jest uznawany za zmieniony, jeśli zachodzi jeden z poniższych warunków:

- jest dodany, ujęty lub zmodyfikowany atrybut Elementarnego procesu,
- jest zmodyfikowany Plik danych wykorzystywany przez Elementarny proces, Elementarny proces zaczął korzystać z nowego Pliku danych lub Elementarny proces przestał korzystać z Pliku danych
- zmieniona jest logika biznesowe Elementarnego procesu.

Celem wyliczenia rozmiaru zmiany Elementarnego procesu należy wyliczyć rozmiar funkcjonalny procesu po zmianie oraz wskazać liczbę zmienianych atrybutów i plików danych. Mając te dane wylicza się procentowy wskaźnik zmiany. Ten wskaźnik służy do wyliczenia wskaźnika wpływu zmiany. Iloczyn wskaźnika wpływu zmiany oraz rozmiaru procesu po zmianie wyznacza rozmiar funkcjonalny zmiany elementarnego procesu.

Procentowy wskaźnik zmiany atrybutów: (liczba zmienionych atrybutów / liczba atrybutów przed zmianą) * 100%

Procentowy wskaźnik zmiany plików: (liczba zmienionych plików / liczba plików przed zmianą) * 100%

Wskaźnik zmiany Elementarnego procesu wyznacza się w oparciu o poniższą tabelę:

Procentowy wskaźnik zmiany atrybutu	$pa \leq 60 \%$	$60 \% < pa \leq 100 \%$	$pa > 100\%$
Procentowy wskaźnik zmiany plików			
$pp \leq 60 \%$	0.25	0.50	0.75
$60\% < pp \leq 100 \%$	0.50	0.75	1.00
$pp > 100\%$	0.75	1.00	1.25

W przypadku, gdy zmiana dotyczy logiki przetwarzania to należy określić, które atrybuty i pliki danych są wykorzystywane w ramach zmiany logiki przetwarzania, przyjmując że atrybuty te/pliki uległy zmianie i uwzględnić je przy wyliczaniu wskaźnika zmiany zgodnie z zasadami określonymi powyżej.

Jeżeli wyliczona na podstawie tak wyznaczonych wskaźników zmiany wielkość zmiany EP, jest większa niż wielkość zmiany polegająca na usunięciu EP przed zmianą i dodaniu nowego EP po zmianie, to jako wielkość zmiany przyjmuje się mniejszą z tych wartości.

W przypadku podziałów / połączeń elementarnych procesów należy te zmiany traktować jako usuwanie starych i dodawanie nowych elementarnych procesów.

Przykład: Elementarny proces E1 korzysta z pliku danych ILF1 i wykorzystuje 10 atrybutów. Ma złożoność 3 PF. W wyniku zmiany, proces korzysta z dodatkowego pliku ILF2 oraz wykorzystuje kolejne dwa atrybuty. Procentowy wskaźnik zmiany atrybutów to $2/10 * 100\% = 20\%$. Procentowy wskaźnik zmiany plików to $1/1 * 100\% = 100\%$. Wskaźnik wpływu wynosi, zatem 0.75. Złożoność procesu po zmianie to 4 PF. Złożoność zmiany to $4 PF * 0.75 = 3PF$.

Jeżeli reguły przetwarzania są sterowane za pomocą parametrów biznesowych, które są zrozumiałe dla użytkownika, to zmiana reguł przetwarzania polegająca wyłącznie na zmianie wartości tych parametrów nie jest traktowana ani jako zmiana Elementarnych procesów, ani jako zmiana Plików danych.

Przykład: Jeżeli w trakcie wprowadzania wniosku, sprawdzane jest, czy wiek beneficjenta przekracza określony parametrem biznesowy próg, np. 60 lat, to zmiana wartości tego parametru np. na 65 lat, która skutkuje zmianą reguły walidacji, nie jest traktowana jako zmiana Elementarnego procesu związanego z wprowadzaniem wniosku.

PROCES SZACOWANIA ROZMIARU OPROGRAMOWANIA / ZMIANY OPROGRAMOWANIA

ZASADY TWORZENIA I UTRZYMYWANIA MODELU ARTEFAKTÓW PF IFPUG

Każda aplikacja, zdefiniowana przez granice oprogramowania, posiada własny model IFPUG w ramach tego projektu. Pojedynczy model IFPUG powinien być utrzymywany jako jeden projekt EA. Struktura tego modelu powinna odzwierciedlać logiczny model funkcji i danych aplikacji.

Pliki danych są modelowane jako zbiór klas powiązanych ze sobą relacją kompozycji. Rekord nadrzędny Pliku danych jest modelowany przy pomocy klasy ze stereotypem «ILF» lub «EIF» (w zależności od typu Pliku danych), nazwa klasy powinna odzwierciedlać nazwę zidentyfikowanego Pliku danych. Pozostałe rekordy Plików danych są modelowane jako klasy w relacji kompozycji z klasą główną. Nazwy tych klas składają się z dwóch części: nazwy Pliku danych, nazwy rekordu oddzielonych znakiem „_”: <nazwa klas głównej>_<nazwa rekordu>. Relacje pomiędzy Plikami danych są modelowane, jako asocjacje klas ze wskazaniem nawigowalności. Atrybuty Pliku danych (DETy) są modelowane jako atrybuty klas, stanowiących danych Plik danych, bez określania ich typów oraz opisów chyba, iż opis jest niezbędny by prawidłowo zidentyfikować dany atrybut. W przypadku Plików danych zawierających duże ilości DETów, atrybuty klas mogą reprezentować grupy atrybutów, lecz w takim przypadku, określana jest liczba DETów reprezentowana przez dany atrybut klasy poprzez określenie licznosci atrybutu (pola „Lower bound” i „Upper bound” opisu atrybutu). Atrybuty związków pomiędzy plikami danych nie są modelowane dodatkowo – są odzwierciedlone poprzez asocjacje klas reprezentujących pliki danych i ich rekordy.

Elementarne procesy są modelowane jako przypadki użycia ze stereotypami «EI», «EQ» oraz «EO». Wykorzystanie Plików danych przez Elementarne procesy jest modelowane jako relacja «dependency» z przypadku użycia reprezentującego Elementarny proces do klasy nadrzędnej danego Pliku danych. Atrybuty wykorzystywane przez Elementarne procesy modelowane są jako atrybuty klasy ze stereotypem «EP», powiązanych z przypadkiem użycia modelującym Elementarne procesy. Związek pomiędzy przypadkiem użycia modelującym Elementarny proces a klasą modelującą atrybuty Elementarnego procesu modelowany jest przy pomocy relacji «dependency» z przypadku

użycia reprezentującego Elementarny proces do klasy modelującej atrybuty, nazwa przypadku użycia i w/w klasy musi być taka sama. Jeżeli atrybut wykorzystywany przez Elementarny proces jest także atrybutem Pliku danych, z którego korzysta Elementarny proces, to nazwa tego atrybutu musi być taka sama jak nazwa atrybutu Pliku danych oraz być poprzedzona nazwą klasy Pliku danych.

W przypadku wymiarowania zmian model IFPUG musi zawierać:

- elementy dodawane,
- elementy usuwane,
- elementy zmieniane, przy czym klasy modelujące Pliki danych i zbiory atrybutów Procesów elementarnych, powinny występować w modelu w wersji sprzed zmiany i po zmianie.

Model IFPUG dla zmian, musi posiadać określoną wersję i posiadać informację, na podstawie jakiej wersji modelu IFPUG, obrazującego oprogramowania przed zmianą, został wykonany.

Dodawane elementy modelu (klasy, przypadki użycia, atrybuty, relacje «dependency») oznaczane są tagiem (zakładka „tagged value”) „DODANE”

Usunięte elementy modelu oznaczane są tagiem „USUNIETE”.

Zmienione elementy modelu sprzed zmiany oznaczane są tagiem „PRZED ZMIANA”, a po zmianie tagiem „PO ZMIANIE”, przy czym przypadki użycia reprezentujące zmienione Elementarne procesy oznaczane są tagiem „ZMIENIONE”.

Relację pomiędzy klasą reprezentującą element zmieniany (przed zmianą), a zmieniony (po zmianie) oznaczamy relacją «trace».

Pliki danych (tylko klasa nadrzędna) oraz Elementarne procesy, elementy zmienione (w wersji po zmianie) oznaczana jest liczba dodawanych, zmienianych, usuwanych lub wykorzystywanych przez zmienioną logikę biznesową atrybutów tagiem „ZMIANA_ARYBUTY”.

Atrybuty Elementarnych procesów po zmianie, które wykorzystywane są przez zmienioną logikę biznesową oznaczane są tagiem „LOGIKA”.

Każdy Elementarny proces i Plik danych posiada swój unikalny niezmienny identyfikator postaci xy-zzzz gdzie:

- x - kod Systemu Informatycznego ARiMR,
- y - kod podsystemu Systemu Informatycznego ARiMR,
- zzzz - unikalny numer elementu modelu.

Identyfikator umieszczany jest w polu „Alias”.

Klasy i przypadki użycia reprezentujące elementy usuwane powinny być oznaczone kolorem czerwonym.

Klasy i przypadki użycia reprezentujące elementy zmieniane przed zmianą powinny być oznaczone kolorem szarym.

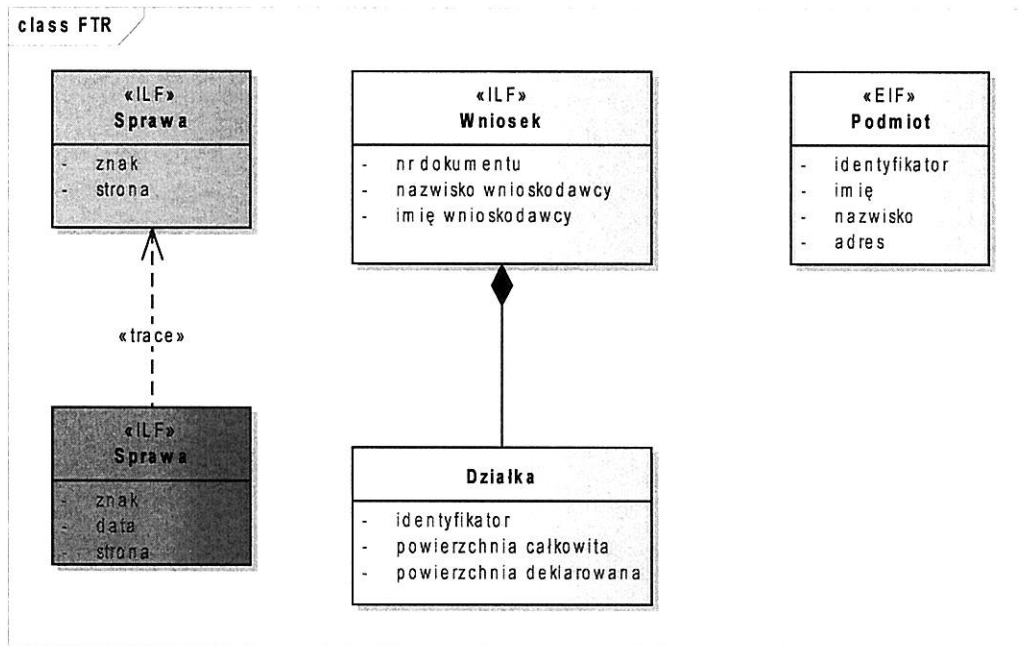
Klasy i przypadki użycia reprezentujące elementy nowe i zmienione po zmianie powinny być oznaczone kolorem zielonym.

Klasy i przypadki użycia reprezentujące elementy nie zmienione, powinny być oznaczone kolorem żółtym.

Przykładu modelu IFPUG załączono do Podręcznika:

- Załącznik nr 1A ARiMR_model_IFPUG_przyklad.eap
- Załącznik nr 1B ARiMR_model_IFPUG_zmiana_przyklad.eap

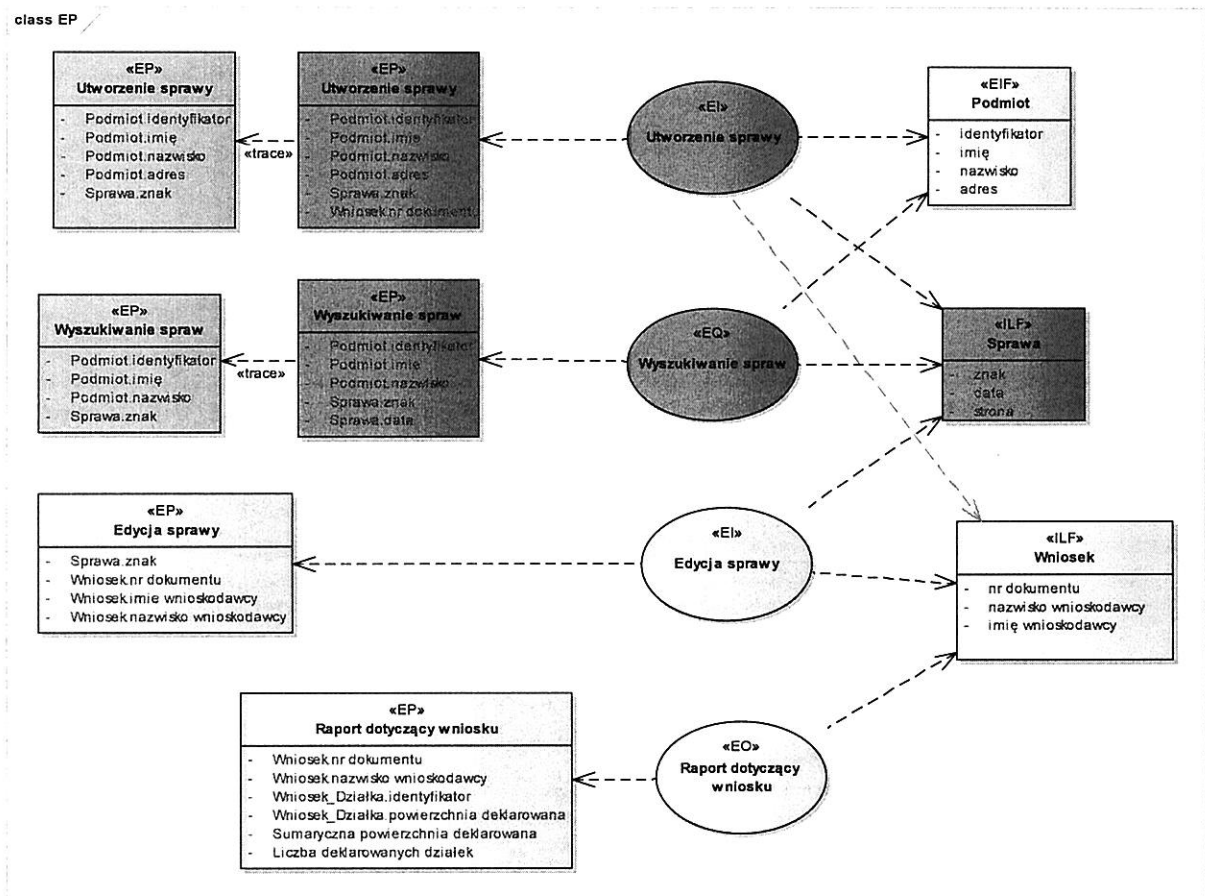
Przykład:



Rysunek 1 Modelowanie Plików danych

Rysunek 1 Modelowanie Plików danych przedstawia fragment modelu IFPUG pewnej zmiany, w którym (od lewej):

- jeden zidentyfikowany ILF „Sprawa”, który uległ zmianie, przed zmianą posiadał 2 DETy, po zmianie posiada 3 DETy, dodany został DET „data”,
- jeden zidentyfikowany ILF „Wniosek”, który nie uległ zmianie, który posiada 2 RETy (Wniosek, Działka) i 6 DETów (nr dokumentu, nazwisko wnioskodawcy, imię wnioskodawcy, identyfikator działki, powierzchnię całkowitą działki, powierzchnię deklarowaną działki),
- jeden zidentyfikowany EIF „Podmiot” posiadający jednego RETa, oraz 4 DETy (identyfikator, imię, nazwisko, adres).



Rysunek 2 Modelowanie Elementarnych procesów

Rysunek 2 Modelowanie Elementarnych procesów przedstawia fragment modelu IFPUG pewnej zmiany, w którym:

Funkcja wejściowa (EI) „Utworzenie sprawy” uległa zmianie. Funkcja ta korzysta z 3 Plików danych (FTR): „Podmiot”, „Sprawa”, „Wniosek”. W wyniku zmiany funkcja zaczęła korzystać z Pliku danych „Wniosek”. Korzysta też z Pliku danych „Sprawa”, która sama uległa zmianie. Po zmianie funkcja posiada 6 DETów, 4 z nich pochodzi z pliku Podmiot (pierwszy człon nazwy atrybutu), 1 z pliku „Sprawa” i 1 z pliku „Wniosek”. W wyniku zmiany Elementarnego procesu „Utworzenie sprawy” dodany został 1 DET „Wniosek.nr dokumentu”.

Funkcja zapytania (EQ) „Wyszukiwanie sprawy” uległa zmianie. Funkcja ta korzysta z 2 Plików danych: „Podmiot”, „Sprawa”. Po zmianie funkcja posiada 5 DETów: 3 DETy pochodzą z pliku „Podmiot”, a 2 z pliku „Sprawa”, w wyniku zmiany doszedł 1 DET o nazwie „Sprawa.data”.

Funkcja wejściowa (EI) „Edycja sprawy” została usunięta.

Funkcja wyjściowa (EO) „Raport dotyczący wniosku” nie uległa zmianie. Funkcja korzysta z jednego Pliku danych: „Wniosek”. Funkcja posiada 6 DETów, z czego 4 pochodzą z Wniosku, a 2 reprezentują dane, które są wyznaczane w trakcie Elementarnego procesu i prezentowane użytkownikowi

ZASADY PREZENTACJI WYNIKÓW SZACOWANIA PF IFPUG W DOKUMENTACJI OFERTOWEJ

W ramach przedstawienia wyników oszacowania w ofercie Wykonawca jest zobowiązany wskazać:

- dodawane elementy modelu IFPUG,
- usuwane elementy modelu IFPUG,
- modyfikowane elementy modelu IFPUG.

Dla każdego wskazywanego elementu modelu musi być podany jego numer, nazwa, wymaganie funkcjonalne, z którego wynika dany element, typ oraz ostateczny rozmiar funkcjonalny zmiany (w przypadku modyfikowanych elementów – rozmiar po uwzględnieniu współczynników wpływu).

Na żądanie ARiMR Wykonawca jest zobowiązany przekazać model IFPUG w postaci projektów EA przed zmianą i po zmianie tj. z modelem IFPUG reprezentującym stan oprogramowania przed wprowadzeniem Modyfikacji oraz po Modyfikacji.

Szablon raportu prezentującego wyniki wymiarowania IFPUG załączono do Podręcznika:

- Załącznik nr 2 ARiMR_raport_z_wymiarowania_IFPUG.xlsx

ZASADY INTERPRETACJI METODY PF IFPUG DLA SPECYFICZNYCH TYPÓW FUNKCJONALNOŚCI I ZMIAN

STOSOWANIE PF IFPUG

ID reguły	IFPUG001: Dane nie stanowiące plików ILF
Opis reguły	<p>Plików danych ILF (w całości lub w części) nie stanowią:</p> <ol style="list-style-type: none"> 1. Metadane mające znaczenie wyłącznie dla technicznego opisu danych biznesowych. 2. Pojedynczo występujące obiekty, posiadające małe ilości rzadko zmiennych atrybutów. 3. Dane statyczne, które nie są zmieniane w wyniku wykonania Elementarnych procesów. 4. Dane redundantne, powielane ze względów wydajnościowych lub technologicznych, w tym zbiory utworzone w celu optymalizacji wyszukiwań i przetwarzań (np. przechowywanie tych samych danych w różnych formatach). 5. Domyślne wartości dla obiektów tworzonych przez aplikację. 6. Enumeracje, zawierające definicje wartości dla atrybutów innych danych. 7. Definicje zakresów poprawności i formatów atrybutów dla innych danych. 8. Dane złączeniowe, czyli zbiory utworzone w celu technicznej realizacji powiązań, np. indeksów, widoków itp. 9. Zbiory tymczasowe (robocze, do sortowania, do przeszukiwania) tworzone podczas realizacji Elementarnych procesów, 10. Zbiory kontrolne monitorujące techniczne aspekty działania aplikacji.
Przykłady	<p>A. System korzysta ze struktur danych, które zawierają informacje o miejscu przechowywania słowników i szablonów dokumentów (np. adres i nazwa bazy danych, nazwa tabeli). Struktury te mają wyłącznie znaczenie</p>

	<p>techniczne, nie są traktowane jako Plik danych, ani jako część Pliku danych.</p> <p>B. Statyczne szablony generowanych przez system dokumentów, nie mogą być zmienione przez użytkownika, ponieważ system nie przewiduje takiej funkcjonalności. Szablony te nie są traktowane jako Pliki danych, ani jako część Plików danych.</p> <p>C. W systemie zdefiniowano enumerację określającą stany przetwarzania dokumentów składającą się z wartości: NOWY, ZATWIERDZONY, ANULOWANY. Enumeracja ta nie jest traktowana jako Plik danych, ani jako część Pliku danych.</p> <p>D. W systemie zdefiniowano zakres wartości, w jakim musi się mieścić data złożenia wniosku, aby wniosek mógł zostać zaakceptowany, zakres ten nie może być zmieniany przez użytkownika, ponieważ system nie przewiduje takiej funkcjonalności. Dane przechowujące akceptowalną datę złożenia wniosku nie są traktowane jako Plik danych, ani jako część Pliku danych.</p> <p>E. W systemie przechowuje się dane pozwalające na wyznaczenie domyślnych wartości, z jakimi tworzone jest nowa sprawa np. znak sprawy – domyślnie kolejny z sekwencji, data utworzenia – domyślnie bieżąca, typ sprawy – domyślny dla typu dokumentu inicjującego. Dane te nie są traktowane jako Plik danych, ani jako część Pliku danych.</p> <p>F. Istniejący na modelu dziedziny związek wiele do wielu pomiędzy klasą A i klasą B wymaga utworzenia w modelu relacyjnym tablicy złączeniowej T. Z punktu widzenia modelu danych IFPUG istnieją jednak dwa pliki danych: A oraz B. Tablica T jest daną techniczną i nie jest uwzględniana w szacowaniu aczkolwiek atrybuty wskazujące związek pomiędzy klasami A i B są elementami każdego z tych plików danych.</p> <p>G. Zidentyfikowane pliki danych należy zaklasyfikować jako pliki ILF, jeśli są to dane wewnątrz granic szacowanego oprogramowania lub jako pliki EIF, jeśli są to dane referencyjne poza granicami szacowanego oprogramowania.</p> <p>H. W celu śledzenia błędów aplikacja rejestruje wyjątki w specjalnym logu technicznym. Log ten jest wykorzystywany przez Wykonawcę do analizy błędów. Log nie jest plikiem danych IFPUG.</p>
--	--

ID reguły	IFPUG002: Zależność pomiędzy ILF, a EIF
Opis reguły	Jeśli dane wewnętrzne aplikacji A będące elementem pliku ILF, są udostępniane aplikacji B, jako EIF, to do oszacowania rozmiaru pliku EIF, brane są pod uwagę tylko te atrybuty, które są udostępniane aplikacji B przez aplikację A. W takim wypadku rozmiar pliku ILF aplikacji A, nie wpływa na rozmiar pliku EIF aplikacji B.

Przykład	<p>Aplikacja wprowadzania i kontroli wniosków przetwarza plik danych Wniosek składający się z 45 atrybutów. Aplikacja wprowadzania i kontroli udostępnia aplikacji naliczania 34 z 45 atrybutów wniosku.</p> <p>Dla aplikacji wprowadzania i kontroli zostanie zidentyfikowany plik ILF o liczbie DET: 45. Dla aplikacji naliczania zostanie zidentyfikowany plik EIF o liczbie DET: 34.</p>
----------	--

ID reguły	IFPUG003: Niezależność plików FTR od implementacji
Opis reguły	Z punktu widzenia wymiarowania, nie ma znaczenie w jaki sposób zostaną zaimplementowane zbiory ILF czy EIF, mogą to być tabele w relacyjnej bazie danych, zbiory w systemie plików, widoki bazodanowe udostępniane innej aplikacji, usługi udostępniające dane itd.
Przykład	Załączniki mapowe rejestrowanych dokumentów są przechowywane w sieciowym systemie plików. W oprogramowaniu są zatem trzy powiązane pliki danych: ILF Dokumenty oraz ILF Załączniki implementowane w bazie relacyjnej oraz ILF Załączniki Mapowane implementowany w systemie plików.

ID reguły	IFPUG005: Definiowanie zbioru wartości atrybutu w Elementarnych procesach
Opis reguły	<p>Jeżeli Elementarny proces umożliwia określenie zbioru wartości pojedynczego atrybutu, zliczany dla tego atrybutu 1 DET, bez względu na to, z ilu i jakich kontrolek oraz wartości korzysta się do określenia tego zbioru np. wielowyboru ze słownika, wyboru wartości z połączonych słowników, początku i końca przedziału wartości,</p> <p>Wybór, przy pomocy interfejsu użytkownika, jednego lub więcej rekordów z wyselekcjonowanego zbioru zaprezentowanego, np. w postaci listy, należy traktować jako akcję atrybut danego EP, czyli 1 DET.</p>
Przykład	<p>Jeżeli Elementarny proces umożliwia wyszukiwanie dokumentów poprzez podanie zakresu daty rejestracji: wartość wartości „od – do”, to jest to zliczane jako 1 DET, gdyż odnosi się on do pojedynczego atrybutu: daty rejestracji.</p> <p>Jeżeli oprogramowanie umożliwia wybór pojedynczego dokumentu z listy dokumentów, to należy taką możliwość wyboru zliczyć jako pojedynczy atrybut EP, w którym ta możliwość występuje (1 DET), gdyż dotyczy pojedynczego atrybutu wskazującego na konkretny dokument (identyfikator dokumentu)</p> <p>Jeżeli Elementarny proces umożliwia dla atrybutu „typ dokumentu” wybór z listy wartości będących połączeniem wartości słowników „typy dokumentów zwykłych” oraz „typy dokumentów specjalnych”, to zliczane jest to jako 1 DET, gdyż dotyczy pojedynczego atrybutu „typ dokumentu”.</p> <p>Jeżeli Elementarny proces umożliwia dla atrybutu „gmina” wybór gminy ze słownika hierarchicznego, którego wartości można zawęzić poprzez wybór</p>

	województwa oraz powiatu, a także poprzez ręczne wpisanie nazwy gminy i wyszukanie pasujących wartości w słowniku, to zliczany jest to jako 1 DET, gdyż dotyczy pojedynczego atrybutu „gmina”.
--	--

ID reguły	IFPUG006: Elementarny proces obejmujący aktualizację danych
Opis reguły	W ramach procesu aktualizacji nie szacuje się dodatkowo przekroczenia granicy systemu danych, jakie mają być zaprezentowane celem umożliwienia ich aktualizacji, przy czym jako atrybuty procesu zlicza się wszystkie atrybuty przekraczające granicę systemu (niezależnie od kierunku).
Przykład	Elementarny proces edycji danych podmiotu prezentuje dane istniejące w systemie i umożliwia ich aktualizację. System prezentuje 7 atrybutów podmiotu, z czego 5 można edytować. Liczba DET dla procesu to 8 – 7 unikalnych atrybutów przekracza granicę systemu oraz jedna akcja potwierdzenia zmiany danych podmiotu.

ID reguły	IFPUG007: Wymiarowanie procesu usuwania danych
Opis reguły	Jeżeli Elementarny proces obejmuje usuwanie danych, to zlicza się jedynie te atrybuty, które są w trakcie tego procesu usuwania prezentowane użytkownikowi a nie wszystkie atrybuty usuwanych danych.
Przykład	Oprogramowanie udostępnia funkcję usuwania dokumentów . Dokument zidentyfikowany jako plik ILF Dokument posiada 21 atrybutów. Jako potwierdzenie usunięcia dokumentu system prezentuje nazwę usuwanego dokumentu. Proces EI Usuwanie dokumenty będzie miał 4 atrybuty: wskazanie (id) usuwanego dokumentu, komunikat potwierdzenia, prezentowaną nazwę usuwanego dokumentu.

ID reguły	IFPUG008: Unikalność atrybutów Elementarnych procesów
Opis reguły	Unikalny atrybut Elementarnego procesu jest dla danego procesu zliczany raz - niezależnie od tego ile razy przekracza granicę oprogramowania w czasie realizacji tego procesu.
Przykład	W funkcji wyszukiwania dokumentów jednym z parametrów wyszukiwania jest data wpływu. Data ta jest prezentowana również na liście wyszukanych dokumentów. Atrybut ten jest zliczany jako 1 DET.

ID reguły	IFPUG009: Standardowe funkcjonalności
Opis reguły	W systemach ARiMR rejestracja operacji biznesowych, dostęp do parametrów biznesowych, słowników oraz uprawnień użytkowników jest standardowym elementem implementowanych funkcjonalności. W związku z tym, w

	<p>Elementarnych procesach nie zlicza się jako wykorzystywane pliki danych tych plików FTR, które dotyczą:</p> <ul style="list-style-type: none"> • rejestrowania operacji dla celów audytowych i archiwalnych, • dostępu do parametrów biznesowych, • dostępu do słowników, • dostępu do danych użytkowników w celu autentykacji oraz autoryzacji. <p>Powyższa zasada nie ma zastosowania dla Elementarnych procesów, które odpowiadają funkcjonalnością przeznaczonym do przetwarzania wyżej wymienionych danych, czyli:</p> <ul style="list-style-type: none"> • przeglądu logów operacji, • przeglądu i aktualizacji parametrów, • przeglądu i aktualizacji słowników, • przeglądu i aktualizacji danych operatorów.
Przykład	<p>Funkcja rejestracji dokumentów wykorzystuje słownik „Typy dokumentów”. Słownik ten jest słownikiem prostym przechowywanym w strukturze modelowanej jako plik danych ILF „Słownik prosty”. W ramach tej funkcji atrybut Elementarnego procesu jest atrybut „typ dokumentu”, ale przyjmuje się, że ten Elementarny proces nie korzysta z pliku ILF „Słownik prosty”.</p> <p>Natomiast Elementarny proces „Edycja słowników prostych” korzysta z pliku ILF „Słownik prosty”.</p>

ID reguły	IFPUG010: Słowniki – model danych
Opis reguły	<p>Słowniki są modelowane jako Pliki danych w znaczeniu struktur przechowujących dane słownikowe, a nie wydzielonych merytorycznie podzbiorów wartości. Jako odrębne Pliki danych FTR modelowane są różne struktury logiczne służące do przechowywania danych słownikowych.</p> <p>Modyfikacja zawartości merytorycznej słowników – zmiana, dodanie, usuwanie wartości ze słownika nie jest szacowana metodą PF IFPUG.</p> <p>Modyfikacja polegająca na wyodrębnieniu nowego słownika, w ramach istniejącej struktury słownikowej, nie podlega szacowaniu metodą PF IFPUG. Modyfikacje te uznaje się wliczone w cenę funkcjonalności korzystającej z wyodrębnionego słownika.</p>
Przykład	<ol style="list-style-type: none"> 1. Wszystkie słowniki proste, to znaczy takie, które nie mają żadnych zależności pomiędzy swoimi wartościami, modelowane są jako jeden zbiór danych „Słownik Prosty” o liczbie atrybutów największego ze słowników prostych. 2. Wszystkie słowniki hierarchiczne modelowane są jako zbiór danych „Słownik hierarchiczny” z określoną strukturą rekordów.

ID reguły	IFPUG012: Interfejs użytkownika - enumeracje
Opis reguły	Jeśli wartość atrybutu w procesie jest nadawana poprzez wybór wartości ze stałej listy (enumeratora), to taki atrybut, razem z jego wyborem, jest zliczany jako 1 DET.
Przykład	Atrybut wprowadzenia miesiąca ważności karty kredytowej jest modelowany jako 1 DET w elementarnym procesie zarówno, gdy edycja jest zrealizowana jako wprowadzenie numeru miesiąca, jak i wybór miesiąca z rozwijalnej listy.

ID reguły	IFPUG013: Interfejs użytkownika - akcje
Opis reguły	<p>Zgodnie z zasadami IFPUG, jako atrybuty elementarnych procesów zlicza się możliwe akcje uruchamiające inne elementarne procesy lub finalizujące dany elementarny proces. Zlicza się zawsze jeden DET dla danej akcji niezależnie od liczby sposobów jej wywołania. Jako DET nie zlicza się akcji związanych z nawigowaniem po interfejsie, krokach kreatora, dostępu do pomocy, akcji anulowania, o ile nie wiąże się z ona z elementarnym procesem realizującym to anulowanie a jest tylko wycofaniem transakcji / zamknięciem okna.</p> <p>W przypadku zmian wizualnych /układów kontrolki interfejsu użytkownika jako zmienione traktuje się wszystkie atrybuty wynikające z możliwości uruchamiania akcji w ramach danego procesu elementarnego.</p> <p>Metodą IFPUG nie szacuje się konstrukcji ani modyfikacji struktury menu aplikacji.</p>
Przykład	W ramach prezentacji listy zarejestrowanych w systemie dokumentów (EQ Wyszukanie dokumentów) możliwe jest uruchomienie EQ Przegląd dokumentu poprzez dwuklik na liście dokumentów lub wybranie opcji „Przeglądaj” dla wybranego dokumentu. Dla tej akcji liczony jest jeden atrybut elementarnego procesu.

STOSOWANIE PF IFPUG DLA ZŁOŻONYCH FUNKCJONALNOŚCI

ID reguły	IFPUG101: Dodatkowy poziom złożoności Elementarnych procesów
Opis reguły	Dla bardzo złożonych Elementarnych procesów definiuje się nowy poziom złożoności o nazwie „COMPLEX” zgodnie z Tabela 2. Rozszerzona tabela wyliczania złożoności Elementarnych procesów.
Przykład	Elementarny proces EI F1, w ramach którego granicę systemu przekracza 66 atrybutów i który korzysta z 3 plików danych ma złożoność 9 PF (jest typu Complex).

Tabela 2. Rozszerzona tabela wyliczania złożoności Elementarnych procesów

EI	DET 1 - 4	DET 5 - 15	DET 16 - 60	DET > 60
FTR < 2	LOW (3)	LOW (3)	AVERAGE (4)	HIGH (6)
FTR 2	LOW (3)	AVERAGE (4)	HIGH (6)	COMPLEX (9)
FTR > 2	AVERAGE (4)	HIGH (6)	HIGH (6)	COMPLEX (9)
EQ	DET 1 - 5	DET 6 - 19	DET 20 - 80	DET > 80
FTR < 2	LOW (3)	LOW (3)	AVERAGE (4)	HIGH (6)
FTR 2	LOW (3)	AVERAGE (4)	HIGH (6)	COMPLEX (9)
FTR > 2	AVERAGE (4)	HIGH (6)	HIGH (6)	COMPLEX (9)
EO	DET 1 - 5	DET 6 - 19	DET 20 - 80	DET > 80
FTR < 2	LOW (4)	LOW (4)	AVERAGE (5)	HIGH (7)
FTR 2 - 3	LOW (4)	AVERAGE (5)	HIGH (7)	COMPLEX (10)
FTR > 3	AVERAGE (5)	HIGH (7)	HIGH (7)	COMPLEX (10)

STOSOWANIE PF IFPUG DLA FUNKCJONALNOŚCI STEROWANYCH PROCESAMI

ID reguły	IFPUG201: Szacowanie formularzy web i usług SOA
Opis reguły	<p>W ramach warstwy formularzy Web należy zdefiniować odpowiednie elementarne procesy EI/EQ/EO realizujące określone cele bez uwzględnienia aspektów technologicznych, czyli zakładając, iż elementarne procesy odpowiadające tym formularzom korzystają bezpośrednio z modelu plików ILF/EIF. Zasady identyfikacji tych procesów powinny abstrahować od implementacji i bazować na zasadach IFPUG.</p> <p>W ramach warstwy usług SOA usługi udostępnione na szynie danych, podlegają szacowaniu metodą IFPUG. Zasady identyfikacji tych procesów powinny abstrahować od implementacji i bazować na zasadach IFPUG.</p> <p>Jeżeli Elementarny proces, wywołuje usługę udostępnioną na szynie danych (bez względu na to, czy sama aplikacja dostarcza tej usługi na szynę danych, czy nie), to przyjmuje się, że proces korzysta z EIF. Przy wymiarowaniu te EIF uwzględnia się atrybuty i rekordy zwracane przez usługę. Jeżeli usługa zwraca jedynie status wykonania, to liczony jest 1 DET.</p>
Przykład	<ol style="list-style-type: none"> 1. Aplikacja Web umożliwia wprowadzanie danych za pomocą formularza: F1. Dane z formularza są zapisywane w systemie za pomocą usługi SOA U1 zachowującej formularz elektroniczny. Formularz F1 i usługa U1 znajdują się wewnątrz tej samej granicy oprogramowania. Usługa U1 nie jest udostępniana innym systemom na szynie danych, jej implementacja

	<p>jest wynikiem decyzji dotyczącej technologii realizacji funkcjonalności wprowadzania danych.</p> <p>Dla oszacowania tej funkcjonalności należy zdefiniować jeden proces EI Wprowadzanie danych. Proces ten obejmuje wszystkie atrybuty EI oraz korzysta z wszystkich Plików danych, z których korzysta usługa U1. Nie jest definiowany odrębny Elementarny proces reprezentujący usługę U1.</p> <p>2. Aplikacja A udostępnia na szynie danych usługę udostępniania danych podmiotu. Z usługi tej korzysta aplikacja B do generowania raportu zawierającego dane podmiotu. Dla aplikacji A identyfikowany jest Elementarny proces (EQ) Udostępnianie danych podmiotu. Dla aplikacji B, Elementarny proces generowania raportu (EO) korzysta z EIF Dane podmiotu.</p>
--	--

STOSOWANIE PF IFPUG DLA ZŁOŻONYCH RAPORTÓW

ID reguły	IFPUG301: Szacowanie złożonych raportów
Opis reguły	<p>Jeżeli proces obliczania agregatu (lub grupy agregatów), których wartości są prezentowane w postaci pól raportu, i których proces obliczenia jest niezależny od procesu obliczenia pozostałych agregatów, korzysta z więcej niż 60 atrybutów, to może być traktowany jako odrębny Elementarny proces typu EO.</p> <p>Tak wyodrębniony proces jest wyłączany z Elementarnego procesu Generowania raportu.</p> <p>Nie można łączyć, agregatów w grupę, jeżeli procesy ich obliczenia są niezależne.</p>
Przykład	Przykładem raportu złożonego jest Tabela X - część kolumn tego raportu spełnia wymogi w/w reguły i są dla nich definiowane niezależne procesy elementarne.

STOSOWANIE PF IFPUG DLA PROCESÓW WSADOWYCH

ID reguły	IFPUG501: Szacowanie procesów wsadowych
Opis reguły	W przypadku szacowania procesów wsadowych, w ramach których wynik działania tego procesu (rozumiany jako cel biznesowy) nie przekracza granicy systemu przyjmuje się, iż wszystkie atrybuty aktualizowane jako element realizacji celu biznesowego, zlicza się do złożoności tego procesu.
Przykład	Proces wsadowy, P1 ma za zadanie ustawić jako „nowy” status wszystkich obiektów klasy A, które na koniec dnia roboczego nie mają statusu „zatwierdzone”. Jako efekt działania procesu ma być zaprezentowana liczba zmodyfikowanych obiektów. Dla każdego aktualizowanego obiektu proces ma

	<p>utworzyć operację klasy B zawierającą informacje o zmianie statusu zawierającą alfanumeryczną kopię danych obiektu A. Zmiana statusu danych obiektu A na „nowy” powoduje ustalenie niektórym jego atrybutom wartości domyślnych.</p>
--	---

Z punktu widzenia szacowania, w powyższym procesie EI jedynym atrybutem przekraczającym granicę systemu jest liczba zmodyfikowanych obiektów. Jednakże uwzględniając regułę IFPUG50 1 jako atrybuty procesu należy uznać również resetowane atrybuty obiektu A oraz atrybut jego statusu, jak i atrybuty obiektu tworzonej operacji.

Poniższy wzór został wygenerowany z formatu EAP na potrzeby wydruku materiałów
do przedmiotowego postępowania.

W trakcie realizacji umowy obowiązującym formatem będzie format EAP tożsamy zakresem
informacyjnym z niniejszym wydrukiem.

Model Report

Model IFPUG

Version • Proposed



Date/Time Generated:

Author:

Table of Contents

Model IFPUG	3
EP diagram	3
FTR diagram	4
Edycja sprawy	4
Raport dotyczący wniosku	4
Utworzenie sprawy.....	5
Wyszukiwanie spraw	5
Działa	6
Edycja sprawy	6
Podmiot	7
Raport dotyczący wniosku	7
Sprawa.....	8
Utworzenie sprawy.....	9
Wniosek powierzchniowy	9
Wyszukiwanie spraw	10

Model IFPUG

Package in package 'Model'

Model IFPUG
Version Phase 1.0 Proposed
created on 2017-05-08. Last modified 2017-05-08

EP diagram

Class diagram in package 'Model IFPUG'

EP
Version 1.0
SPI created on 2017-05-08. Last modified 2017-05-16

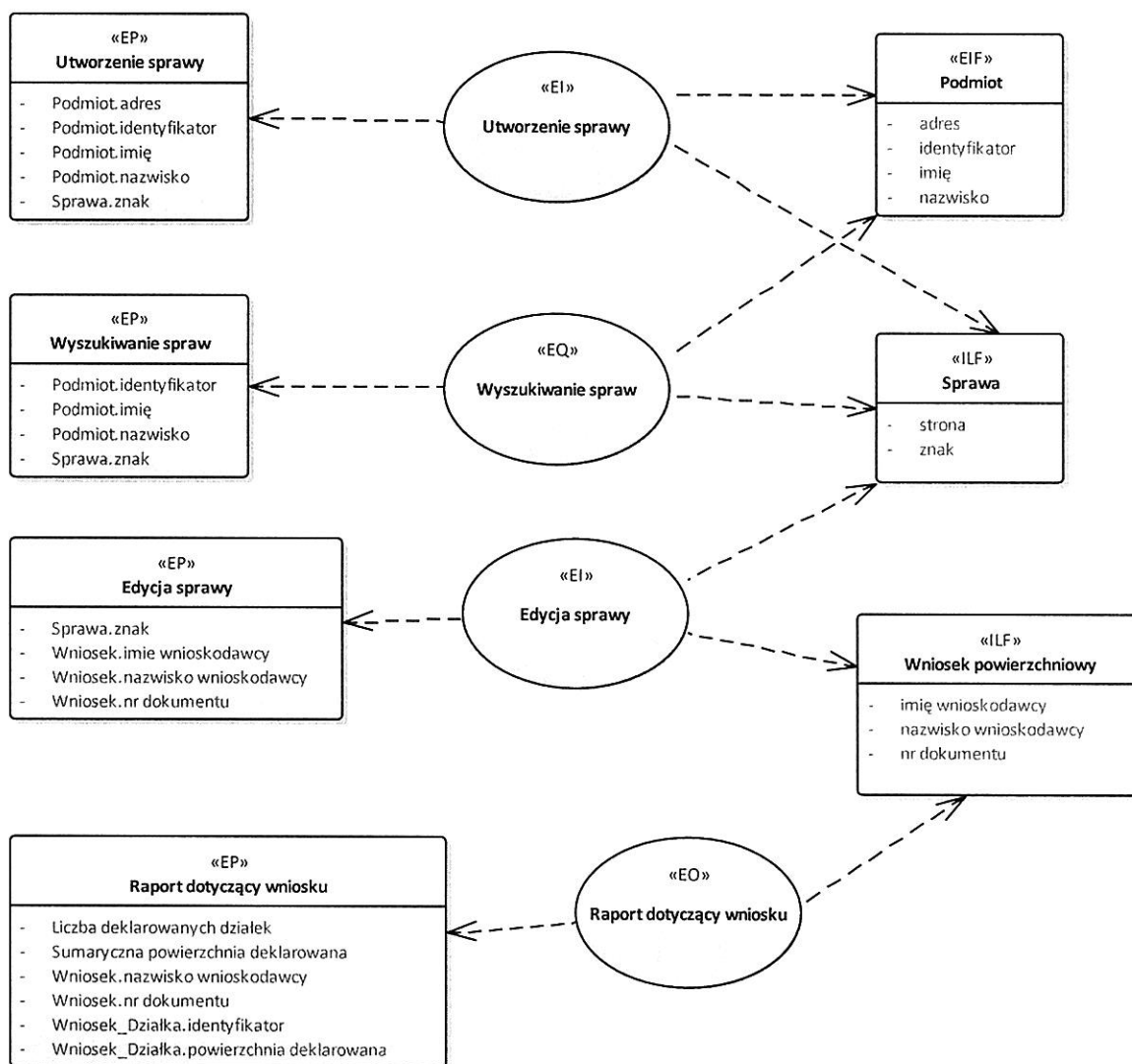


Figure 1: EP

FTR diagram

Class diagram in package 'Model IFPUG'

FTR
Version 1.0

SP1 created on 2017-05-08. Last modified 2017-05-08

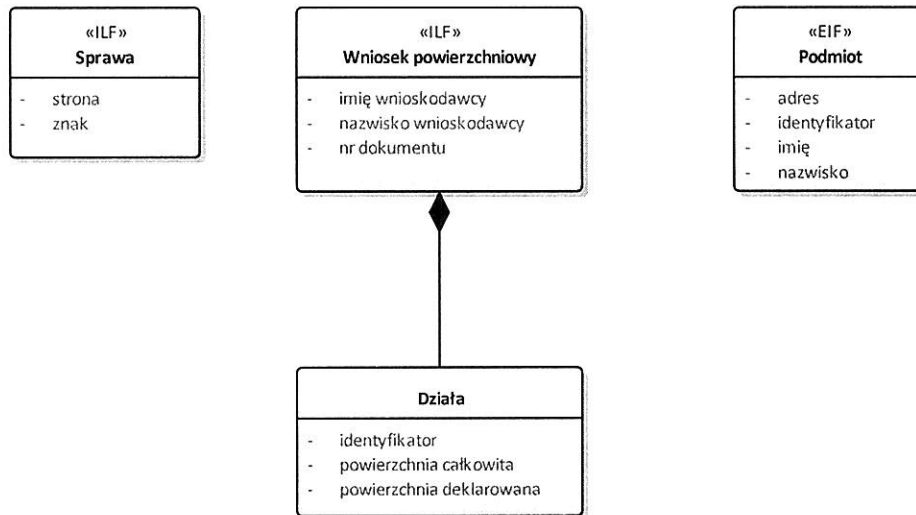


Figure 2: FTR

Edycja sprawy

UseCase «EI» in package 'Model IFPUG'

Edycja sprawy
Version 1.0 Phase 1.0 Proposed



SP1 created on 2017-05-16. Last modified 2017-05-16

CONNECTORS	
	Dependency Source -> Destination From: Edycja sprawy : UseCase, Public To: Edycja sprawy : Class, Public
	Dependency Source -> Destination From: Edycja sprawy : UseCase, Public To: Wniosek powierzchniowy : Class, Public
	Dependency Source -> Destination From: Edycja sprawy : UseCase, Public To: Sprawa : Class, Public

Raport dotyczący wniosku

UseCase «EO» in package 'Model IFPUG'




Raport dotyczący wniosku
Version 1.0 Phase 1.0 Proposed
SPI created on 2017-05-16. Last modified 2017-05-16

CONNECTORS	
 Dependency	Source -> Destination
From:	Raport dotyczący wniosku : UseCase, Public
To:	Raport dotyczący wniosku : Class, Public
 Dependency	Source -> Destination
From:	Raport dotyczący wniosku : UseCase, Public
To:	Wniosek powierzchniowy : Class, Public

Utworzenie sprawy

UseCase «EI» in package 'Model IFPUG'



Utworzenie sprawy
Version 1.0 Phase 1.0 Proposed
SPI created on 2017-05-16. Last modified 2017-05-16

CONNECTORS	
 Dependency	Source -> Destination
From:	Utworzenie sprawy : UseCase, Public
To:	Podmiot : Class, Public
 Dependency	Source -> Destination
From:	Utworzenie sprawy : UseCase, Public
To:	Sprawa : Class, Public
 Dependency	Source -> Destination
From:	Utworzenie sprawy : UseCase, Public
To:	Utworzenie sprawy : Class, Public


Wyszukiwanie spraw

UseCase «EQ» in package 'Model IFPUG'

Wyszukiwanie spraw
Version 1.0 Phase 1.0 Proposed
SPI created on 2017-05-16. Last modified 2017-05-16

CONNECTORS	
 Dependency	Source -> Destination
From:	Wyszukiwanie spraw : UseCase, Public
To:	Podmiot : Class, Public
 Dependency	Source -> Destination
From:	Wyszukiwanie spraw : UseCase, Public
To:	Sprawa : Class, Public

CONNECTORS


 **Dependency** Source -> Destination
 From: Wyszukiwanie spraw : UseCase, Public
 To: Wyszukiwanie spraw : Class, Public

Działa


Class in package 'Model IFPUG'


Działa
 Version 1.0 Phase 1.0 Proposed
 SPI created on 2017-05-08. Last modified 2017-05-17


OUTGOING STRUCTURAL RELATIONSHIPS

 Aggregation from Działa to «ILF» Wniosek powierzchniowy
| Direction is 'Source -> Destination'. |

ATTRIBUTES

 identyfikator : Private | Is static False. Containment is Not Specified. |

 powierzchnia calkowita : Private | Is static False. Containment is Not Specified. |


 powierzchnia deklarowana : Private | Is static False. Containment is Not Specified. |

Edycja sprawy

Class «EP» in package 'Model IFPUG'

Edycja sprawy
 Version 1.0 Phase 1.0 Proposed
 SPI created on 2017-05-08. Last modified 2017-05-16

CONNECTORS

 **Dependency** Source -> Destination
 From: Edycja sprawy : UseCase, Public
 To: Edycja sprawy : Class, Public

ATTRIBUTES

 Sprawa.znak : Private | Is static False. Containment is Not Specified. |

 Wniosek.imie wnioskodawcy : Private | Is static False. Containment is Not Specified. |

ATTRIBUTES	
☛ Wniosek.nazwisko wnioskodawcy : Private	[Is static False. Containment is Not Specified.]
☛ Wniosek.nr dokumentu : Private	[Is static False. Containment is Not Specified.]

Podmiot

Class «EIF» in package 'Model IFPUG'

Podmiot
Version 1.0 Phase 1.0 Proposed
SPI created on 2017-05-08. Last modified 2017-05-08

CONNECTORS	
☛ Dependency Source -> Destination From: Wyszukiwanie spraw : UseCase, Public To: Podmiot : Class, Public	
☛ Dependency Source -> Destination From: Utworzenie sprawy : UseCase, Public To: Podmiot : Class, Public	

ATTRIBUTES	
☛ adres : Private	[Is static False. Containment is Not Specified.]
☛ identyfikator : Private	[Is static False. Containment is Not Specified.]
☛ imię : Private	[Is static False. Containment is Not Specified.]
☛ nazwisko : Private	[Is static False. Containment is Not Specified.]

Raport dotyczący wniosku

Class «EP» in package 'Model IFPUG'





Raport dotyczący wniosku
Version 1.0 Phase 1.0 Proposed
SPI created on 2017-05-08. Last modified 2017-05-16

CONNECTORS	
 Dependency Source -> Destination From: Raport dotyczący wniosku : UseCase, Public To: Raport dotyczący wniosku : Class, Public	
ATTRIBUTES	
 Liczba deklarowanych działek : Private	[Is static False. Containment is Not Specified.]
 Sumaryczna powierzchnia deklarowana : Private	[Is static False. Containment is Not Specified.]
 Wniosek.nazwisko wnioskodawcy : Private	[Is static False. Containment is Not Specified.]
 Wniosek.nr dokumentu : Private	[Is static False. Containment is Not Specified.]
 Wniosek_Dzialka.identyfikator : Private	[Is static False. Containment is Not Specified.]
 Wniosek_Dzialka.powierzchnia deklarowana : Private	[Is static False. Containment is Not Specified.]

Sprawa

Class «ILF» in package 'Model IFPUG'

Sprawa
Version 1.0 Phase 1.0 Proposed
SPI created on 2017-05-08. Last modified 2017-05-08

CONNECTORS	
 Dependency Source -> Destination From: Wyszukiwanie spraw : UseCase, Public To: Sprawa : Class, Public	
 Dependency Source -> Destination From: Utworzenie sprawy : UseCase, Public To: Sprawa : Class, Public	
 Dependency Source -> Destination From: Edycja sprawy : UseCase, Public To: Sprawa : Class, Public	
ATTRIBUTES	
 strona : Private	[Is static False. Containment is Not Specified.]

ATTRIBUTES

znak : Private


[Is static False, Containment is Not Specified.]

Utworzenie sprawy

Class «EP» in package 'Model IFPUG'

Utworzenie sprawy
Version 1.0 Phase 1.0 Proposed
SPI created on 2017-05-08. Last modified 2017-05-16

CONNECTORS

 **Dependency** Source -> Destination
From: Utworzenie sprawy : UseCase, Public
To: Utworzenie sprawy : Class, Public

ATTRIBUTES

Podmiot.adres : Private

[Is static False, Containment is Not Specified.]

Podmiot.identyfikator : Private

[Is static False, Containment is Not Specified.]

Podmiot.imię : Private

[Is static False, Containment is Not Specified.]

Podmiot.nazwisko : Private

[Is static False, Containment is Not Specified.]

Sprawa.znak : Private

[Is static False, Containment is Not Specified.]

Wniosek powierzchniowy






Class «ILF» in package 'Model IFPUG'

Wniosek powierzchniowy
Version 1.0 Phase 1.0 Proposed
SPI created on 2017-05-08. Last modified 2017-05-08

INCOMING STRUCTURAL RELATIONSHIPS

⇒ Aggregation from Dział to «ILF» Wniosek powierzchniowy






[Direction is 'Source -> Destination'.]

CONNECTORS	
 Dependency Source -> Destination From: Edycja sprawy : UseCase, Public To: Wniosek powierzchniowy : Class, Public	
 Dependency Source -> Destination From: Raport dotyczący wniosku : UseCase, Public To: Wniosek powierzchniowy : Class, Public	
ATTRIBUTES	
 imię wnioskodawcy : Private	[Is static False. Containment is Not Specified.]
 nazwisko wnioskodawcy : Private	[Is static False. Containment is Not Specified.]
 nr dokumentu : Private	[Is static False. Containment is Not Specified.]

Wyszukiwanie spraw

Class «EP» in package 'Model IFPUG'

Wyszukiwanie spraw
 Version 1.0 Phase 1.0 Proposed
 SPI created on 2017-05-08. Last modified 2017-05-16

CONNECTORS	
 Dependency Source -> Destination From: Wyszukiwanie spraw : UseCase, Public To: Wyszukiwanie spraw : Class, Public	
ATTRIBUTES	
 Podmiot.identyfikator : Private	[Is static False. Containment is Not Specified.]
 Podmiot.imię : Private	[Is static False. Containment is Not Specified.]
 Podmiot.nazwisko : Private	[Is static False. Containment is Not Specified.]
 Sprawa.znak : Private	[Is static False. Containment is Not Specified.]

Poniższy wzór został wygenerowany z formatu EAP na potrzeby wydruku materiałów do przedmiotowego postępowania.
W trakcie realizacji umowy obowiązującym formatem będzie format EAP tożsamy zakresem informacyjnym z niniejszym wydrukiem.

Model Report

Model IFPUG

Version • Proposed



Date/Time Generated:

Author:

Table of Contents

Model IFPUG	3
EP diagram	3
FTR diagram	3
Edycja sprawy	4
Raport dotyczący wniosku	4
Utworzenie sprawy	5
Wyszukiwanie spraw	5
Działka	6
Edycja sprawy	6
Podmiot	7
Raport dotyczący wniosku	7
Sprawa	8
Sprawa	9
Utworzenie sprawy	9
Utworzenie sprawy	10
Wniosek	11
Wyszukiwanie spraw	11
Wyszukiwanie spraw	12

Model IFPUG

Package in package 'Model'

Model IFPUG
Version Phase 1.0 Proposed
created on 2017-05-08. Last modified 2017-05-08

EP diagram

Class diagram in package 'Model IFPUG'

EP
Version 1.0
SPI created on 2017-05-08. Last modified 2017-05-16

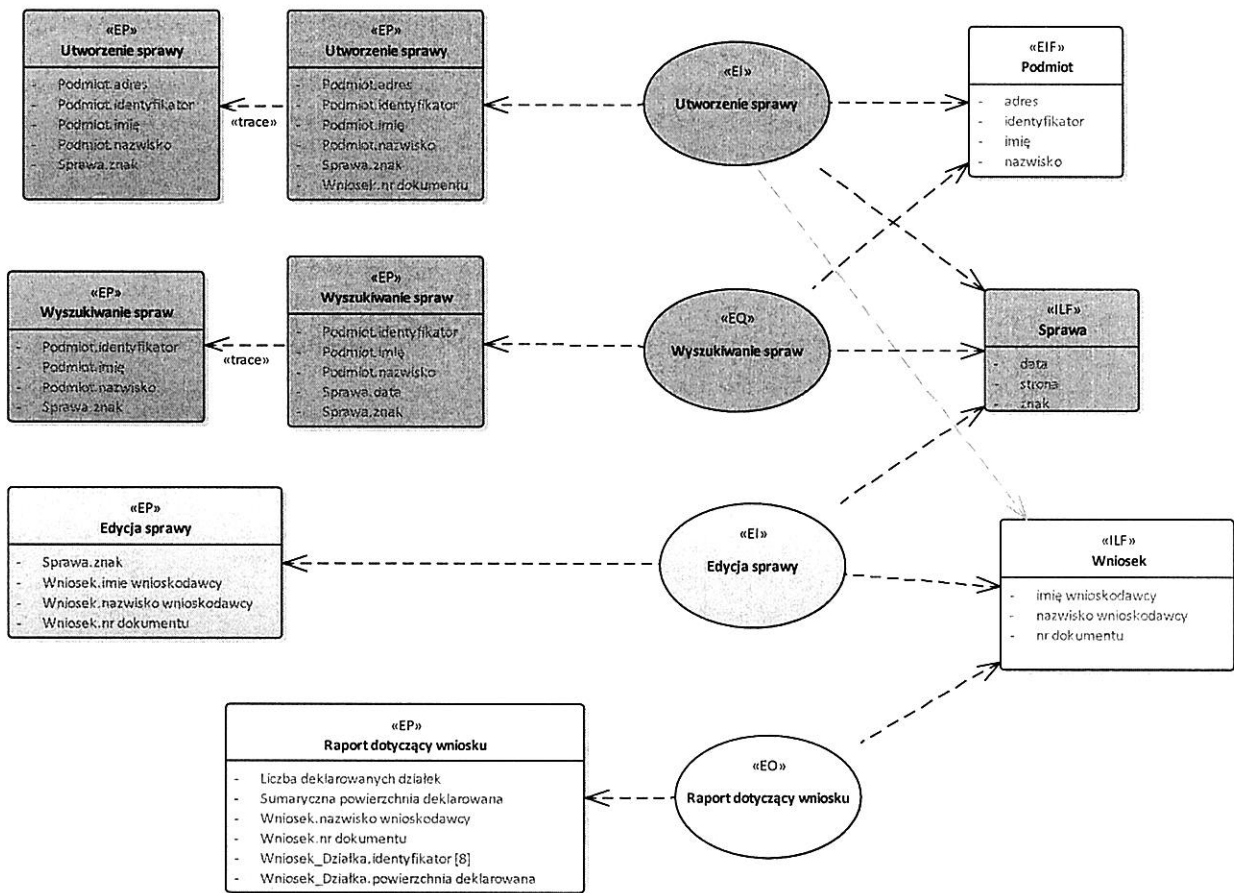


Figure 1: EP

FTR diagram

Class diagram in package 'Model IFPUG'

FTR
Version 1.0
SPI created on 2017-05-08. Last modified 2017-05-08

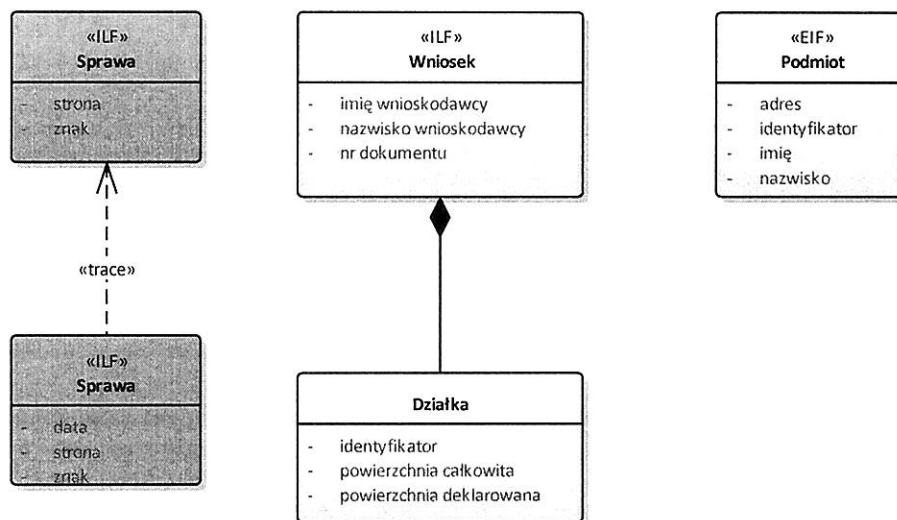


Figure 2: FTR

Edycja sprawy

UseCase «EI» in package 'Model IFPUG'

Edycja sprawy
 Version 1.0 Phase 1.0 Proposed
 SPI created on 2017-05-16. Last modified 2017-05-16


CONNECTORS	
	Dependency Source -> Destination From: Edycja sprawy : UseCase, Public To: Edycja sprawy : Class, Public
	Dependency Source -> Destination From: Edycja sprawy : UseCase, Public To: Wniosek : Class, Public
	Dependency Source -> Destination From: Edycja sprawy : UseCase, Public To: Sprawa : Class, Public


Raport dotyczący wniosku

UseCase «EO» in package 'Model IFPUG'

Raport dotyczący wniosku
 Version 1.0 Phase 1.0 Proposed
 SPI created on 2017-05-16. Last modified 2017-05-16

CONNECTORS

 **Dependency** Source -> Destination
 From: Raport dotyczący wniosku : UseCase, Public
 To: Wniosek : Class, Public


 **Dependency** Source -> Destination
 From: Raport dotyczący wniosku : UseCase, Public
 To: Raport dotyczący wniosku : Class, Public


Utworzenie sprawy


UseCase «EI» in package 'Model IFPUG'


Utworzenie sprawy
 Version 1.0 Phase 1.0 Proposed
 SPI created on 2017-05-16. Last modified 2017-05-16

CONNECTORS

 **Dependency** Source -> Destination
 From: Utworzenie sprawy : UseCase, Public
 To: Sprawa : Class, Public

 **Dependency** Source -> Destination
 From: Utworzenie sprawy : UseCase, Public
 To: Wniosek : Class, Public

 **Dependency** Source -> Destination
 From: Utworzenie sprawy : UseCase, Public
 To: Utworzenie sprawy : Class, Public


 **Dependency** Source -> Destination
 From: Utworzenie sprawy : UseCase, Public
 To: Podmiot : Class, Public


Wyszukiwanie spraw

UseCase «EQ» in package 'Model IFPUG'

Wyszukiwanie spraw
 Version 1.0 Phase 1.0 Proposed
 SPI created on 2017-05-16. Last modified 2017-05-16

CONNECTORS

 **Dependency** Source -> Destination
 From: Wyszukiwanie spraw : UseCase, Public
 To: Wyszukiwanie spraw : Class, Public

 **Dependency** Source -> Destination
 From: Wyszukiwanie spraw : UseCase, Public
 To: Sprawa : Class, Public




CONNECTORS	
 Dependency	Source -> Destination
From:	Wyszukiwanie spraw : UseCase, Public
To:	Podmiot : Class, Public

Działka

Class in package 'Model IFPUG'

Działka
Version 1.0 Phase 1.0 Implemented
SPI created on 2017-05-08. Last modified 2017-05-17

OUTGOING STRUCTURAL RELATIONSHIPS	
 Aggregation from Działka to «ILF» Wniosek	[Direction is 'Source -> Destination'.]

ATTRIBUTES	
 identyfikator : Private	[Is static False. Containment is Not Specified.]
 powierzchnia całkowita : Private	[Is static False. Containment is Not Specified.]
 powierzchnia deklarowana : Private	[Is static False. Containment is Not Specified.]

Edycja sprawy

Class «EP» in package 'Model IFPUG'

Edycja sprawy
Version 1.0 Phase 1.0 Validated
SPI created on 2017-05-08. Last modified 2017-05-16

CONNECTORS	
 Dependency	Source -> Destination
From:	Edycja sprawy : UseCase, Public
To:	Edycja sprawy : Class, Public

ATTRIBUTES	
 Sprawa.znak : Private	[Is static False. Containment is Not Specified.]
 Wniosek.imie wnioskodawcy : Private	[Is static False. Containment is Not Specified.]

ATTRIBUTES	
✎ Wniosek.nazwisko wnioskodawcy : Private	[Is static False, Containment is Not Specified.]
✎ Wniosek.nr dokumentu : Private	[Is static False, Containment is Not Specified.]

Podmiot

Class «EIF» in package 'Model IFPUG'

Podmiot
Version 1.0 Phase 1.0 Implemented
SPI created on 2017-05-08. Last modified 2017-05-08








CONNECTORS	
✎ Dependency Source -> Destination From: Wyszukiwanie spraw : UseCase, Public To: Podmiot : Class, Public	
✎ Dependency Source -> Destination From: Utworzenie sprawy : UseCase, Public To: Podmiot : Class, Public	

ATTRIBUTES	
✎ adres : Private	[Is static False, Containment is Not Specified.]
✎ identyfikator : Private	[Is static False, Containment is Not Specified.]
✎ imię : Private	[Is static False, Containment is Not Specified.]
✎ nazwisko : Private	[Is static False, Containment is Not Specified.]

Raport dotyczący wniosku

Class «EP» in package 'Model IFPUG'





Raport dotyczący wniosku
Version 1.0 Phase 1.0 Implemented
SPI created on 2017-05-08. Last modified 2017-05-16

CONNECTORS	
 Dependency Source -> Destination From: Raport dotyczący wniosku : UseCase, Public To: Raport dotyczący wniosku : Class, Public	
ATTRIBUTES	
 Liczba deklarowanych działek : Private	[Is static False. Containment is Not Specified.]
 Sumaryczna powierzchnia deklarowana : Private	[Is static False. Containment is Not Specified.]
 Wniosek.nazwisko wnioskodawcy : Private	[Is static False. Containment is Not Specified.]
 Wniosek.nr dokumentu : Private	[Is static False. Containment is Not Specified.]
 Wniosek_Dzialka.identyfikator : Private Multiplicity: ([8], Allow duplicates: 0, Is ordered: False)	[Is static False. Containment is Not Specified.]
 Wniosek_Dzialka.powierzchnia deklarowana : Private	[Is static False. Containment is Not Specified.]

Sprawa

Class «ILF» in package 'Model IFPUG'

Sprawa
 Version 1.0 Phase 1.0 Proposed
 SPI created on 2017-05-08. Last modified 2017-05-15

CONNECTORS	
 Trace «trace» Source -> Destination From: Sprawa : Class, Public To: Sprawa : Class, Public	
 Dependency Source -> Destination From: Wyszukiwanie spraw : UseCase, Public To: Sprawa : Class, Public	
 Dependency Source -> Destination From: Utworzenie sprawy : UseCase, Public To: Sprawa : Class, Public	
 Dependency Source -> Destination From: Edycja sprawy : UseCase, Public To: Sprawa : Class, Public	

ATTRIBUTES	
data : Private	[Is static False. Containment is Not Specified.]
strona : Private	[Is static False. Containment is Not Specified.]
znak : Private	[Is static False. Containment is Not Specified.]

Sprawa

Class «ILF» in package 'Model IFPUG'

Sprawa
Version 1.0 Phase 1.0 Approved
SPI created on 2017-05-08. Last modified 2017-05-15

CONNECTORS	
 Trace «trace» Source -> Destination From: Sprawa : Class, Public To: Sprawa : Class, Public	

ATTRIBUTES	
strona : Private	[Is static False. Containment is Not Specified.]
znak : Private	[Is static False. Containment is Not Specified.]

Utworzenie sprawy

Class «EP» in package 'Model IFPUG'

Utworzenie sprawy
Version 1.0 Phase 1.0 Approved
SPI created on 2017-05-08. Last modified 2017-05-16

CONNECTORS	
 Trace «trace» Source -> Destination From: Utworzenie sprawy : Class, Public To: Utworzenie sprawy : Class, Public	



ATTRIBUTES	
Podmiot.adres : Private	[Is static False. Containment is Not Specified.]

ATTRIBUTES	
Podmiot.identyfikator : Private	[Is static False. Containment is Not Specified.]
Podmiot.imię : Private	[Is static False. Containment is Not Specified.]
Podmiot.nazwisko : Private	[Is static False. Containment is Not Specified.]
Sprawa.znak : Private	[Is static False. Containment is Not Specified.]

Utworzenie sprawy

Class «EP» in package 'Model IFPUG'

Utworzenie sprawy
Version 1.0 Phase 1.0 Proposed
SPI created on 2017-05-08. Last modified 2017-05-16

CONNECTORS	
 Trace «trace» Source -> Destination From: Utworzenie sprawy : Class, Public To: Utworzenie sprawy : Class, Public	
 Dependency Source -> Destination From: Utworzenie sprawy : UseCase, Public To: Utworzenie sprawy : Class, Public	

ATTRIBUTES	
Podmiot.adres : Private	[Is static False. Containment is Not Specified.]
Podmiot.identyfikator : Private	[Is static False. Containment is Not Specified.]
Podmiot.imię : Private	[Is static False. Containment is Not Specified.]
Podmiot.nazwisko : Private	[Is static False. Containment is Not Specified.]
Sprawa.znak : Private	[Is static False. Containment is Not Specified.]
Wniosek.nr dokumentu : Private	[Is static False. Containment is Not Specified.]

ATTRIBUTES

Properties:

DODANY =

[Is static False. Containment is Not Specified.]

Wniosek

Class «ILF» in package 'Model IFPUG'

Wniosek

Version 1.0 Phase 1.0 Implemented


SPI created on 2017-05-08. Last modified 2017-05-15


INCOMING STRUCTURAL RELATIONSHIPS


⇒ Aggregation from Działka to «ILF» Wniosek

[Direction is 'Source ⇒ Destination'.]

CONNECTORS

 **Dependency** Source -> Destination
 From: Raport dotyczący wniosku : UseCase, Public
 To: Wniosek : Class, Public

 **Dependency** Source -> Destination
 From: Utworzenie sprawy : UseCase, Public
 To: Wniosek : Class, Public

 **Dependency** Source -> Destination
 From: Edycja sprawy : UseCase, Public
 To: Wniosek : Class, Public

ATTRIBUTES

❏ imię wnioskodawcy : Private

[Is static False. Containment is Not Specified.]

❏ nazwisko wnioskodawcy : Private

[Is static False. Containment is Not Specified.]

❏ nr dokumentu : Private

[Is static False. Containment is Not Specified.]






Wyszukiwanie spraw

Class «EP» in package 'Model IFPUG'

Wyszukiwanie spraw

Version 1.0 Phase 1.0 Approved








SPI created on 2017-05-08. Last modified 2017-05-16

CONNECTORS	
 Trace «trace» Source -> Destination From: Wyszukiwanie spraw : Class, Public To: Wyszukiwanie spraw : Class, Public	
ATTRIBUTES	
 Podmiot.identyfikator : Private	[Is static False. Containment is Not Specified.]
 Podmiot.imię : Private	[Is static False. Containment is Not Specified.]
 Podmiot.nazwisko : Private	[Is static False. Containment is Not Specified.]
 Sprawa.znak : Private	[Is static False. Containment is Not Specified.]

Wyszukiwanie spraw

Class «EP» in package 'Model IFPUG'

Wyszukiwanie spraw
 Version 1.0 Phase 1.0 Proposed
 SPI created on 2017-05-08. Last modified 2017-05-16

CONNECTORS	
 Trace «trace» Source -> Destination From: Wyszukiwanie spraw : Class, Public To: Wyszukiwanie spraw : Class, Public	
 Dependency Source -> Destination From: Wyszukiwanie spraw : UseCase, Public To: Wyszukiwanie spraw : Class, Public	
ATTRIBUTES	
 Podmiot.identyfikator : Private	[Is static False. Containment is Not Specified.]
 Podmiot.imię : Private	[Is static False. Containment is Not Specified.]
 Podmiot.nazwisko : Private	[Is static False. Containment is Not Specified.]
 Sprawa.data : Private Properties: DODANY =	[Is static False. Containment is Not Specified.]
 Sprawa.znak : Private	[Is static False. Containment is Not Specified.]

