

Podręcznik wyceny oprogramowania  
w Agencji Restrukturyzacji  
i Modernizacji Rolnictwa

*v.4.1*

*26 marca 2024*

## Spis treści

1	Wstęp.....	4
2	Podstawowe pojęcia występujące w Podręczniku .....	4
3	Wycena Zmiany .....	7
3.1	Całkowity Rozmiar Zmiany .....	7
3.2	Wycena na podstawie Rozmiaru Funkcjonalnego.....	7
3.3	Rozmiar Funkcjonalny Zmiany .....	9
3.4	Pracochłonność Uzupełniająca Zmiany .....	9
3.4.1	Wymagania Wydajnościowe .....	10
4	Wprowadzenie do metody IFPUG.....	10
4.1	Metoda punktów funkcyjnych .....	10
4.2	Specyfikacja Wymagań.....	11
4.3	Użytkownik, granica oprogramowania, Artefakty IFPUG i Elementy Wyceny .....	11
4.3.1	Identyfikacja Plików Danych .....	16
4.3.2	Identyfikacja Elementarnych Procesów.....	20
4.3.3	Wyliczanie Rozmiaru Funkcjonalnego konwersji danych.....	28
4.3.4	Rozmiar Funkcjonalny Elementów Wyceny.....	29
4.4	Wyznaczanie rozmiaru funkcjonalnego zmian oprogramowania.....	29
5	Zasady wyznaczania rozmiaru Funkcjonalnego w ARiMR.....	31
5.1	Integracja z systemem hurtowni danych.....	44
5.2	Integracja z systemem BPMS .....	45
6	Wymagania Technologiczne .....	47
7	Dokumentowanie wyceny .....	48
7.1	Definicje .....	49
7.2	Ogólne zasady.....	49
7.3	Utworzenie i przygotowanie Pomiaru dla Zmiany .....	50
7.4	Przekazanie Pomiaru do Akceptacji .....	52
7.5	Opiniowanie i uzgadnianie wyceny .....	55

7.6	Utworzenie nowej wersji Pomiaru dla Zmiany .....	56
7.7	Utworzenie Pomiaru Różnicowego .....	57
8	Odstępstwa od zasad dokumentowania wyceny.....	63

## 1 WSTĘP

Niniejszy dokument (**Podręcznik**) ma na celu opisanie metody wyznaczania Rozmiaru Funkcjonalnego oprogramowania i zmian oprogramowania w Agencji Restrukturyzacji i Modernizacji Rolnictwa.

Metoda ta opiera się na:

- metodzie IFPUG (nazwa pochodzi od organizacji zajmującej się utrzymywaniem i rozwojem metody *International Function Point Users Group*, [www.ifpug.org](http://www.ifpug.org)) opisaney w dokumencie pt.: *Function Point Counting Practices Manual* w wersji 4.3.1 oraz
- publikacji NESMA (*Netherlands Software Metrics Association*, [www.nesma.org](http://www.nesma.org)) pt.: *Function Point Analysis for Software Enhancement*, rozszerzającej zasady metody IFPUG w zakresie wymiarowania zmian oprogramowania.

Obydwa dokumenty nazywane dalej Dokumentami Bazowymi są dostępne na stronach swoich organizacji w języku angielskim.

Choć najważniejsze koncepcje zawarte w Dokumentach Bazowych zostały przywołane w niniejszym Podręczniku, to, jako kompletny opis metody punktów funkcyjnych stosowanych w ARiMR, należy traktować łącznie opis zawarty w Podręczniku oraz Dokumentach Bazowych.

Podręcznik jest w pewnych obszarach uproszczeniem, a w innych doprecyzowaniem lub rozszerzeniem zasad opisanych w Dokumentach Bazowych. W przypadku stwierdzenia rozbieżności pomiędzy zapisami Podręcznika, a zapisami Dokumentów Bazowych, jako wiążące należy przyjąć zapisy Podręcznika.

W rozdziale Dokumentowanie wyceny opisano zasady dokumentowania procesu wyznaczania Rozmiaru Funkcjonalnego przeprowadzanego zgodnie z Podręcznikiem.

Źródła wykorzystane w opracowaniu:

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

## 2 PODSTAWOWE POJĘCIA WYSTĘPUJĄCE W PODRĘCZNIKU

**Aplikacja** – moduł oprogramowania zdefiniowany w Umowie, który posiada własne niezależne od reszty oprogramowania Granice Oprogramowania

**Artefakt IFPUG** – podstawowy element opisu oprogramowania, identyfikowany zgodnie z niniejszym Podręcznikiem na potrzeby wyznaczania Rozmiaru Funkcjonalnego. Szczegółowy opis Artefaktów IFPUG będzie przedstawiony w dalszej części Podręcznika.

**Całkowity Rozmiar Zmiany** – rozmiar Zmiany wyrażony w Punktach Funkcyjnych wyznaczony zgodnie z rozdziałem 3.1.

**Elementy Wyceny** – najmniejsze elementy Wymagań zidentyfikowane zgodnie z niniejszym Podręcznikiem, którym można przypisać określony rozmiar wyrażony w Punktach Funkcyjnych.

**Granica Oprogramowania** – granica, względem której dokonuje się pomiaru oprogramowania, zdefiniowana w rozdziale 4.3.

**Pracochłonność Uzupełniająca Zmiany** – szacunek pracochłonności dla Zmiany określony metodą ekspercką w Roboczościach zgodnie z zasadami opisanymi w rozdziale 3.4

**Proces Wsadowy** – proces realizowany przez oprogramowanie, którego celem jest przetworzenie danych, przy czym przetwarzane dane nie przekraczają Granic Oprogramowania, a są odczytywane z pamięci trwałej oraz wynik przetwarzania nie przekracza Granic Oprogramowania, a zapisywany jest w pamięci trwałej; wykonanie procesu nie wymaga interakcji z użytkownikiem.

**Rozmiar Funkcjonalny** – rozmiar funkcjonalny oprogramowania lub zmiany oprogramowania wyznaczony zgodnie z niniejszym Podręcznikiem i wyrażony w Punktach Funkcyjnych.

**Rozmiar Funkcjonalny Zmiany** – rozmiar Zmiany zgodny z definicją zawartą w rozdziale 3.3.

**Specyfikacja (Specyfikacja Wymagań)** – opis Wymagań, na podstawie którego wyznaczany jest Rozmiar Funkcjonalny.

**Umowa** – umowa na wytworzenie lub zmianę oprogramowania, w której ARiMR jest stroną zamawiającą oprogramowanie, w której wynagrodzenie za wytworzone lub zmienione oprogramowanie określa się na podstawie Rozmiaru Funkcjonalnego.

**Warstwa Technologiczna** – kategoria zgodna z definicją zawartą w rozdziale 6

**Wykonawca** – podmiot realizujący Zmiany, druga obok ARiMR strona Umowy.

**Wymaganie** – wymaganie Zamawiającego w odniesieniu do Aplikacji, w zakresie realizacji jego celów biznesowych, które stały się zobowiązaniami Wykonawcy związanymi z realizacją lub zmianą Aplikacji.

**Wymaganie Technologiczne** – wymaganie zgodne z definicją zawartą w rozdziale 6.

**Wymiarowanie** – proces obejmujący wszystkie niezbędne czynności do wyznaczenia Rozmiaru Funkcjonalnego na podstawie Specyfikacji.

**Zmiana** – wytworzenie lub zmiana oprogramowania wykonana na podstawie Wymagań, które są przedmiotem pojedynczego zamówienia ARiMR, zrealizowana na zasadach opisanych w Umowie.

## 3 WYCENA ZMIANY

### 3.1 CAŁKOWITY ROZMIAR ZMIANY

Na potrzeby wyceny Zmiany wyznacza się Całkowity Rozmiar Zmiany, na podstawie Rozmiaru Funkcjonalnego oraz Pracochłonności Uzupełniającej wymagań wchodzących w zakres Zmiany oraz ceny za Punkt Funkcyjny oraz ceny za Roboczodzień określonych w Umowie.

$$CRZ = RFZ + PUZ \cdot \frac{CRG}{CPF}$$

gdzie:

**CRZ** – Całkowity Rozmiar Zmiany pracochłonności wyrażony w Punktach Funkcyjnych,

**RFZ** – Rozmiar Funkcjonalny Zmiany wyrażony w Punktach Funkcyjnych,

**PUZ** – Pracochłonność Uzupełniająca Zmiany wymagań wchodzących w zakres Zmiany określona zgodnie z Podręcznikiem w Roboczodniach,

**CRG** – cena Roboczodnia określona w Umowie,

**CPF** – cena Punktu Funkcyjnego określona w Umowie.

### 3.2 WYCENA NA PODSTAWIE ROZMIARU FUNKCJONALNEGO

Koszt wytworzenia lub zmiany oprogramowania określony na podstawie Rozmiaru Funkcjonalnego wymagań obejmuje realizację tych wymagań zgodnie z normami obowiązującymi dla danego oprogramowania.

Normy te muszą być co najmniej zgodne, w danym obszarze, ze standardami i dobrymi praktykami obowiązującymi na rynku IT.

Normy te obejmują obszary definiowane przez wymagania pozafunkcjonalne i dotyczą m.in.:

- dotrzymania SLA związanego z wydajnością np. czasem przetwarzania czy odpowiedzi oprogramowania,
- ergonomii użytkownika oprogramowania oraz rozwiązań w zakresie interfejsu użytkownika, w tym m.in.:
  - sposobu wprowadzania i wyświetlania danych,
  - wyświetlania komunikatów informujących Użytkownika o stanie przetwarzania danych,
  - wyświetlania informacji niezbędnej do podjęcia przez Użytkownika właściwej decyzji, w tym odświeżania danych,
  - potwierdzania krytycznych decyzji użytkownika,

- możliwości anulowania realizowanych zadań,
  - możliwości przerwania, a następnie wznowienia pracy itp.
- bezpieczeństwa i audytowalności oprogramowania, w tym m.in.:
  - autentykacji użytkowników,
  - ograniczenia dostępu tylko do funkcji i danych,
  - rejestracji operacji wykonywanych i zdarzeń systemowych,
- monitorowania poprawności działania, dostępności i wydajności oprogramowania,
- zakresu i sposobu udokumentowania oprogramowania,
- stosowania określonych technologii,
- stosowania określonej architektury aplikacji, w tym m.in. podziału na moduły, warstwy, usługi itp.,
- sposobu integracji z innymi systemami, w tym m.in.:
  - stosowania lub niestosowania komponentów pośredniczących w komunikacji,
  - sposobu wymiany danych (np. poprzez bazę danych, pliki wymiany, usługi itp.),
  - konfiguracji i formatu komunikatów wymienianych przez usługi.

W/w normy wynikają z dokumentacji, kodu źródłowego oraz sposobu działania oprogramowania, którego rozwój jest przedmiotem Umowy.

W obszarach, w których, zgodnie z powyższym akapitem w/w normy pozostają nieokreślone, obowiązującymi są normy wynikające z Wymagań określonych w Umowie oraz ze standardów i dobrych praktyk obowiązujących na rynku IT.

Realizację Wymagań zgodnie z w/w normami uznaje się za wliczoną przez Wykonawcę w cenę Punktu Funkcyjnego.

W cenę Punktu Funkcyjnego uznaje się za wliczone wszystkie prace niezbędne do realizacji Wymagań zgodnie z w/w normami, w tym, m.in.:

- przeprowadzenie analizy systemowej i prac projektowych,
- zaprojektowanie i zbudowanie interfejsu użytkownika,
- wytworzenie kodu źródłowego,
- przeprowadzenie testów,
- opracowanie dokumentacji,
- zdefiniowanie i skonfigurowanie uprawnień do nowych / zmodyfikowanych funkcjonalności,
- migracja danych niezbędna do poprawnego działania nowej / zmodyfikowanej funkcjonalności,
- utworzenie lub aktualizacja danych niezbędne do poprawnego działania nowej / zmodyfikowanej funkcjonalności,



- utworzenie lub aktualizacja parametrów definiujących działanie aplikacji (w kontekście technologicznym jak i biznesowym) niezbędne do poprawnego działania nowej / zmodyfikowanej funkcjonalności,
- utworzenie lub aktualizacja słowników wspierających działanie aplikacji (w kontekście technologicznym jak i biznesowym) niezbędne do poprawnego działania nowej / zmodyfikowanej funkcjonalności,
- optymalizacja oprogramowania / struktur bazy danych, które pozwolą nowej/zmodyfikowanej funkcjonalności spełnić obowiązujące SLA,
- prace konfiguracyjne dotyczące elementów architektury, infrastruktury i oprogramowania standardowego, niezbędne do poprawnego działania nowej / zmodyfikowanej funkcjonalności,
- prace organizacyjne, koordynacyjne, warsztaty, szkolenia oraz spotkania niezbędne do realizacji wymagań zgodnie z w/w normami.

Żadne prace, które są niezbędne do realizacji wymagań zgodnie z w/w normami, wycenianych na podstawie ich Rozmiaru Funkcjonalnego, nie mogą być wyceniane odrębnie.

Wymagania, które potwierdzają zastosowanie w/w norm przy realizacji innych wymagań (w tym także związanych z wytworzeniem funkcjonalności w postaci nowej Aplikacji), wycenianych na podstawie ich Rozmiaru Funkcjonalnego, nie mogą być wyceniane odrębnie.

Jedynie wymagania, które zmieniają lub rozszerzają w/w normy dla oprogramowania, które zostało już wdrożone i jest eksploatowane produkcyjnie, wyceniane są odrębnie.

### 3.3 ROZMIAR FUNKCJONALNY ZMIANY

Rozmiar Funkcjonalny Zmiany to Rozmiar Funkcjonalny wyznaczony na podstawie wymagań funkcjonalnych, wymagań przejściowych związanych z konwersją danych oraz Wymagań Technologicznych wchodzących w zakres Zmiany.

### 3.4 PRACOCHOŃNNOŚĆ UZUPEŁNIAJĄCA ZMIANY

Na podstawie wymagań wchodzących w zakres Zmiany, które zgodnie z Podręcznikiem, spełniają następujące kryteria:

- nie można wyznaczyć ich Rozmiaru Funkcjonalnego (nie należą do grupy wymagań wskazanych w rozdziale 3.3, dla których wyznaczany jest Rozmiar Funkcjonalny),
- nie określają prac, które są elementem realizacji innych wymagań,
- ich realizacja nie jest wliczona w cenę Punktu Funkcyjnego,
- w przypadku Wymagań Wydajnościowych, spełniają kryteria opisane w rozdziale 3.4.1,

wyznaczana jest Pracochłonność Uzupełniająca Zmiany na podstawie oceny eksperckiej i określana w Roboczościach.

W przypadku wymagań, dla których zostanie określona pracochłonność zgodnie z poprzednim akapitem, niezbędne jest uzasadnienie szacunku pracochłonności, które powinno zawierać spis czynności składowych niezbędnych do realizacji danego wymagania, wraz z określeniem

pracochłonności tych czynności. Wymienione w spisie czynności powinny być określone z dokładnością co najmniej do dwóch Roboczodni, tzn. czynności o większej pracochłonności niż dwa Roboczodni, powinny być rozpisane na mniejsze czynności składowe.

### 3.4.1 WYMAGANIA WYDAJNOŚCIOWE

Wymagania Wydajnościowe, to wymagania polegające na nałożeniu ograniczeń polegających na przetworzeniu określonych danych, przez określoną funkcjonalność, w określonym czasie.

Wymagania Wydajnościowe podlegają odrębnej wycenie, jeżeli spełniają następujące kryteria:

- ograniczenia te wynikają ze zgłoszonych wymagań biznesowych ARiMR,
- jeżeli Wykonawca wykaże, że funkcjonalność, której dotyczy wymaganie, zrealizowana zgodnie z normami, o których mowa w rozdziale 3.2, nie spełnia w/w ograniczeń,
- w wymaganiu zostaną określone adekwatne rozwiązania oraz prace mające na celu optymalizację funkcjonalności, które Wykonawca planuje wprowadzić lub wykonać w celu spełnienia w/w ograniczeń.

## 4 WPROWADZENIE DO METODY IFPUG

### 4.1 METODA PUNKTÓW FUNKCYJNYCH

Metoda IFPUG, a także inne metody punktów funkcyjnych (nazywane także metodami wymiarowania funkcjonalnego lub w skrócie metodami funkcjonalnymi), to zbiory reguł, które, na podstawie opisu funkcjonalności lub zmian funkcjonalności, określają jak wyznaczyć rozmiar funkcjonalny wyrażony w jednostkach, jakimi są punkty funkcyjne.

Opis funkcjonalności lub zmian funkcjonalności może mieć postać tekstową lub wizualną, sformułowaną w języku potocznym lub sformalizowaną przy pomocy określonej notacji np. UML. Założeniem metody jest analiza w/w opisu, przełożenie go na opis formalny funkcjonalności lub zmian funkcjonalności charakterystyczny dla danej metody (w przypadku metody IFPUG będzie to zbiór pozostających w odpowiedniej relacji Artefaktów IFPUG) i wyznaczenie na jego podstawie liczby punktów funkcyjnych.

Punkt funkcyjny jest abstrakcyjną miarą funkcjonalności dostarczanej przez oprogramowanie. Im oprogramowanie posiada więcej punktów funkcyjnych, tym więcej dostarcza funkcjonalności użytkownikowi.



Rysunek 1. Idea metody punktów funkcyjnych.

## 4.2 SPECYFIKACJA WYMAGAŃ

Wymagania zawarte w Specyfikacji muszą być jednoznacznie identyfikowane. Poziom granulacji Wymagań w Specyfikacji musi umożliwiać precyzyjne wskazanie opisu, na podstawie którego identyfikowane są Elementy Wyceny.

Każde Wymaganie musi dotyczyć konkretnej (w szczególnych przypadkach nowej) Aplikacji.

Specyfikacja powinna zawierać opis funkcjonalności z punktu widzenia Użytkownika, określać cele jakie osiągnie dzięki opisywanej funkcjonalności, korzystać z pojęć i przedstawiać zachowanie Aplikacji, które jest istotne z punktu widzenia celów Użytkownika.

Cała Specyfikacja powinna posługiwać się spójnie tymi samymi pojęciami. Te same typy obiektów i funkcji powinny być określane tą samą nazwą.

Opis Wymagań dotyczący zmian funkcjonalności lub utworzenia kolejnych wariantów funkcjonalności może odwoływać się do istniejącego i dostępnego dla obu Stron opisu analitycznego. W takich przypadkach należy wskazywać jednoznacznie artefakty analityczne (np. identyfikatory Przypadków Użycia), bądź ich odpowiednie fragmenty, w których ten opis się znajduje.

Jeżeli Specyfikacja posiada wiele wersji, dla których wyznaczany jest Rozmiar Funkcjonalny, to zmiany Specyfikacji w kolejnych wersjach, które wpływają na Rozmiar Funkcjonalny wyznaczony na podstawie Specyfikacji muszą być jednoznacznie wyróżnione.

## 4.3 UŻYTKOWNIK, GRANICA OPROGRAMOWANIA, ARTEFAKTY IFPUG I ELEMENTY WYCENY

Punktem wyjścia metody IFPUG jest analiza Specyfikacji z punktu widzenia **Użytkownika**. Użytkownik to abstrakcyjny uczestnik procesu biznesowego, który wchodzi w interakcję z oprogramowaniem, aby realizować swoje cele. Użytkownikiem może być osoba korzystająca z oprogramowania, ale także inne oprogramowanie czy urządzenie. Dla Użytkownika oprogramowanie jest „czarną skrzynką”. Oznacza to, że Użytkownik wie tylko, co oprogramowanie ma mu zapewnić, nie interesuje go to, jak to zapewni. Konsekwencją takiego podejścia jest abstrahowanie metody IFPUG od sposobu implementacji wymagań czy użytej technologii. Jeżeli

oprogramowanie ma realizować określoną funkcjonalność, to z punktu widzenia Użytkownika, bez względu jak ono zostanie zrealizowane, będzie miało zawsze ten sam rozmiar funkcjonalny. Decyzje dotyczące sposobu realizacji funkcjonalności nie wpływają na rozmiar funkcjonalny oprogramowania.

W związku z powyższym, przy identyfikacji Artefaktów IFPUG oraz Elementów Wyceny **nie** są brane pod uwagę takie elementy opisu Wymagania jak:

- opis czynności, które zostaną wykonane w celu realizacji Wymagania,
- opis artefaktów analitycznych, jakie zostaną wytworzone lub zmienione w wyniku realizacji Wymagania,
- wskazanie technologii, z użyciem których zostanie zrealizowane Wymaganie,
- opis procesów i struktur, które nie opisują w sposób bezpośredni celów Użytkownika.

Kolejnym ważnym pojęciem dla metody IFPUG jest **Granica Oprogramowania**.

Granica Oprogramowania jest abstrakcyjną granicą pomiędzy tym, co jest wymiarowanym oprogramowaniem, a Użytkownikiem. Wewnątrz granicy oprogramowania musi w całości znajdować się logika biznesowa oprogramowania, która w całości zrealizuje daną funkcjonalność, a także dane, którymi ta logika zarządza, tzn. zapisuje je i odczytuje w zależności od potrzeb.

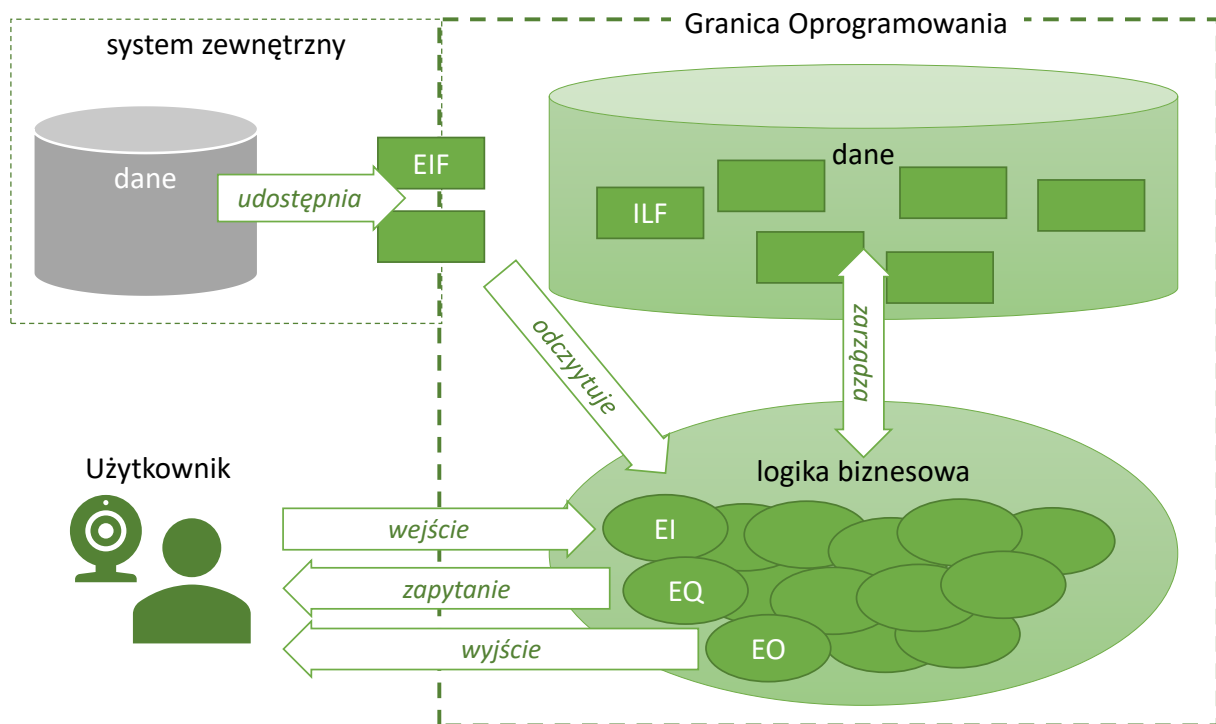
Poza Granicą Oprogramowania znajduje się oprogramowanie zewnętrzne, które zarządza danymi, do których logika biznesowa wymiarowanego oprogramowania ma dostęp.

Należy zwrócić uwagę, że oprogramowanie zewnętrzne (system zewnętrzny) z punktu widzenia wymiarowanego oprogramowania może pełnić dwie różne role: Użytkownika lub systemu udostępniającego dane. Ten sam system zewnętrzny może pełnić obydwie role dla wymiarowanego oprogramowania w zależności od sposobu interakcji. Jeżeli system zewnętrzny inicjuje interakcję, to pełni on rolę Użytkownika, jeżeli wymiarowane oprogramowanie inicjuje interakcję z systemem zewnętrznym, to pełni on rolę systemu udostępniającego dane.

Tym samym, jeżeli jeden system pełni dla drugiego systemu rolę Użytkownika, to drugi pełni dla pierwszego rolę systemu udostępniającego dane i odwrotnie.

Granica Oprogramowania nie jest determinowana przez decyzje technologiczne, infrastrukturalne czy architektoniczne, jest arbitralnym założeniem dla procesu Wymiarowania, na którym opiera się metoda IFPUG. Względem granicy oprogramowania identyfikuje się Artefakty IFPUG.

Granica Oprogramowania jest określana w Umowie. Granicę Oprogramowania definiuje się w postaci listy Aplikacji, czyli komponentów oprogramowania posiadających określoną funkcjonalność lub zakres odpowiedzialności, dla których dokonuje się niezależnego pomiaru oprogramowania. Konsekwencją niezależnego pomiaru oprogramowania Aplikacji jest sposób pomiaru, w którym inne Aplikacje traktowane są jak oprogramowanie zewnętrzne, mimo że rozwijane są w ramach tego samego projektu lub Umowy. Oprogramowanie pojedynczej Aplikacji najczęściej jest jednolite technologicznie i wspólnie administrowane (nie jest to jednak warunek konieczny).



**Rysunek 2. Oprogramowanie z punktu widzenia metody IFPUG.**

Metoda IFPUG definiuje następujące artefakty (dalej: **Artefakty IFPUG**):

- **EP** (ang. Elementary Process), **Elementarny Proces** – minimalna jednostka funkcjonalności realizująca jakiś cel Użytkownika, która spełnia jednocześnie następujące kryteria:
  - jest rozpoznawalna z punktu widzenia Użytkownika,
  - w pełni realizuje transakcję,
  - w całości zawiera się w sobie, tzn. nie występują poza nią żadne dodatkowe czynności niezbędne do jej zainicjowania czy wypełnienia celu Użytkownika,
  - 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.

**Przykład 4.2.A:** *Jeżeli wymaganiem użytkownika jest wprowadzenie wniosku, to stan stabilny biznesu aplikacji zostanie osiągnięty, jeżeli wniosek zostanie w całości wprowadzony i zapisany. Jeżeli użytkownik wprowadzi część danych wniosku i przerwie wprowadzanie, to w momencie przerwania wprowadzania wniosku biznesu aplikacji nie będzie pozostawiony w stanie stabilnym, ponieważ cel Użytkownika nie został osiągnięty, wymagane są dodatkowe czynności do jego osiągnięcia.*

- **EI** (ang. External Input) – funkcja wejścia, Elementarny Proces przetwarzający lub kontrolujący dane przychodzące z zewnątrz Granicy Oprogramowania, które są zapisywane w Granicach Oprogramowania lub modyfikują działanie oprogramowania.
- **EQ** (ang. External Inquiry) – funkcja zapytania, Elementarny Proces przetwarzający lub kontrolujący informacje Przesyłane na zewnątrz Granicy Oprogramowania.

- **EO** (ang. External Output) - funkcja wyjścia, Elementarny Proces przetwarzający lub kontrolujący informacje przesyłane na zewnątrz Granicy Oprogramowania, obejmujący logikę przetwarzania, wykraczającą poza logikę funkcji zapytania (EQ).
- **ILF** (ang. Internal Logical File) – wewnętrzny Plik Danych, rozpoznawana przez użytkownika grupa logicznie powiązanych danych, utrzymywana wewnątrz granicy wymiarowanego oprogramowania.
- **EIF** (ang. External Interface File) – zewnętrzny Plik Danych, rozpoznawana przez użytkownika grupa logicznie powiązanych danych, odczytywana przez wymiarowane oprogramowanie, utrzymywana poza granicami wymiarowanego oprogramowania, np. przez oprogramowanie zewnętrzne.
- **Plik Danych** (PD)– inaczej funkcja danych, ILF lub EIF.
- **FTR** (ang. File Type Referenced) – odniesienie do Pliku Danych, funkcja odczytu lub zapisu określonego Pliku danych wykonywana w ramach określonego Elementarnego procesu.
- **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 danych w obrębie Pliku Danych, w której atrybuty występują względem siebie w relacji 1 do 1, pełnią rolę atrybutów asocjacji Pliku Danych z innymi plikami danych lub jest specyficzna dla określonej specjalizacji Pliku Danych.

**Przykład 4.2.B:** *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.*

Identyfikacja wszystkich Artefaktów IFPUG musi wynikać jednoznacznie ze Specyfikacji Wymagań. Nie dopuszcza się identyfikacji Artefaktów IFPUG na podstawie domniemania, że opis zawarty w Specyfikacji Wymagań jest niepełny.

Elementarne Procesy oraz Pliki Danych, to Artefakty IFPUG, którym zgodnie z metodą opisaną w niniejszym Podręczniku (Rozdział 4.3.4) można przypisać konkretny Rozmiar Funkcjonalny, dlatego są one jednocześnie Elementami Wyceny.

Odrębność Wymagań nie oznacza odrębności zidentyfikowanych na ich podstawie Artefaktów IFPUG. Różne wymagania mogą opisywać różne aspekty tego samego Artefaktu IFPUG, z wielu różnych Wymagań może wynikać ten sam Artefakt IFPUG.

Rozpoznawalność, z punktu widzenia użytkownika, oznacza, że dany Artefakt IFPUG jest niezbędny do osiągnięcia celów biznesowych użytkownika. Cel biznesowy to cel, który może być opisany przy pomocy takich samych pojęć co proces biznesowy wspierany przez wymiarowane oprogramowanie, bez potrzeby odwoływania się do procesów i artefaktów technologii, w jakiej wykonane jest oprogramowanie.

**Przykład 4.2.C:** *Zapisanie danych w tabeli tymczasowej nie jest Artefaktem IFPUG, nie odnosi się do celów biznesowych Użytkownika, a jedynie określa czynności systemu.*

**Przykład 4.2.D:** *Wprowadzenie danych wniosku jest Artefaktem IFPUG, realizuje określony cel Użytkownika.*

**Przykład 4.2.E:** *Odświeżenie formularza nie jest Artefaktem IFPUG, nie odnosi się do celów biznesowych Użytkownika, a jedynie do czynności systemu.*

**Przykład 4.2.F:** *Modyfikacja danych wniosku jest Artefaktem IFPUG, realizuje określony cel Użytkownika.*

Choć metoda IFPUG mówi o wymaganiach funkcjonalnych użytkownika, to Użytkownika nie należy utożsamiać z osobą definiującą wymagania. Użytkownikiem może być inny system bądź urządzenie, które, co oczywiste, nie bierze udziału w definiowaniu wymagań. Osoby definiujące wymagania mogą posiadać świadomość, w jaki sposób działa oprogramowanie i rozumieć procesy i artefakty technologiczne, które stoją za działaniem oprogramowania, mogą zatem użyć tej wiedzy do definiowania wymagań. W takich przypadkach na podstawie tak sformułowanych wymagań należy określić cele biznesowe osiąmane przez Użytkownika, dla którego system będzie czarną skrzynką i dopiero na tej podstawie zidentyfikować Artefakty IFPUG.

**Przykład 4.2.G:** *Przedstawiciel biznesu sformułował następujące wymaganie „Powyżej przycisku „Zapisz” należy umieścić dodatkowy checkbox, a po zapisaniu danych ma się wyświetlić komunikat z nr wniosku”. W celu identyfikacji Artefaktów IFPUG stwierdzamy, że Użytkownik funkcjonalny potrzebuje zmiany Wprowadzania wniosku, dodatkowy checkbox oznacza także konieczność przechowywania w systemie dodatkowej informacji. Na tej podstawie identyfikujemy proces EI Wprowadzenie wniosku oraz ILF Wnioski, które zgodnie z tym wymaganiem będą wymagać zmiany.*

Występowanie artefaktu analitycznego, np.: przypadku użycia lub klasy opisującego dane oprogramowanie, nie jest wystarczającą podstawą do identyfikacji odpowiadającego mu Artefaktu IFPUG, chyba że opis zawarty w artefakcie analitycznym, spełnia definicję danego Artefaktu IFPUG.

**Przykład 4.2.H:** *Wymaganie opisuje zmianę funkcjonalności, która w analizie została opisana przez dwa przypadki użycia. Pierwszy „Wyznacz kwotę dopłaty”, a drugi „Sprawdź wielkość limitu”. Pierwszy przypadek w celu realizacji swojego celu musi wywołać drugi przypadek. Z punktu widzenia Użytkownika występuje tylko jeden Elementarny Proces „Wyznacz kwotę dopłaty”, funkcjonalność „Sprawdź wielkość limitu” jest elementem tego procesu odpowiedzialnym za element logiki biznesowej, który został wyodrębniony jako podfunkcja, w celu optymalizacji opisu analitycznego lub implementacji pod kątem ponownego użycia reguł biznesowych. Potrzeby Użytkownika nie uzasadniają występowania „Sprawdź wielkość limitu” jako Artefaktu IFPUG.*

W celu wyznaczenia Rozmiaru Funkcjonalnego oprogramowania konieczne jest zidentyfikowanie na podstawie Specyfikacji wszystkich Plików Danych oraz Elementarnych Procesów i wykazanie, że wynikają one ze Specyfikacji opisującej dane oprogramowanie. Część opisu Wymagania, która dotyczy pojedynczego Pliku Danych lub Elementarnego Procesu jest Elementem Wyceny.

W celu wyznaczenia Rozmiaru Funkcjonalnego zmian oprogramowania konieczne jest zidentyfikowanie i wykazanie, że wynikają ze Specyfikacji opisującej zmiany oprogramowania, wszystkich przypadków dodania, zmiany lub usunięcia Plików Danych oraz Elementarnych Procesów, które zostaną dodane, zmienione lub usunięte w wyniku zmiany oprogramowania.

#### 4.3.1 IDENTYFIKACJA PLIKÓW DANYCH

Identyfikacji Plików Danych dokonuje się wyłącznie na podstawie analizy zbioru danych, do których oprogramowanie musi się odwoływać lub którymi musi zarządzać, aby działało zgodnie ze Specyfikacją. Identyfikacja Plików Danych polega na wyodrębnieniu grup logicznie powiązanych danych z punktu widzenia funkcjonalności w taki sposób, że wszystkie powiązane wzajemnie ze sobą relacją zależności należy łączyć w pojedyncze Pliki Danych.

Relację zależności pomiędzy danymi definiujemy następująco:

- Dane zależne mają sens, tzn. można je zinterpretować tylko wtedy, jeżeli występują dane, od których zależą.
- Jeżeli usunięcie danych pociąga za sobą usunięcie innych powiązanych danych, to te inne powiązane dane są od nich zależne.

**Przykład 4.2.1.A:** *Wymaganie pozwala na rejestrację wniosków o płatność zawierających dane beneficjenta oraz dane działek. System posiada funkcjonalność wprowadzania i obsługi wniosków. System musi przechowywać dane wniosku, które zawierają w sobie dane działek. Dane działek we wniosku zależą od danych wniosku. Jeżeli zostanie usunięty wniosek, to także zostaną usunięte wszystkie dane działek zawarte we wniosku. Na podstawie wymagania zostanie zidentyfikowany jeden Plik Danych, który będzie obejmował wszystkie dane wniosku, także dane beneficjenta i działek.*

**Przykład 4.2.1.B:** *Inne wymaganie mówi, że na podstawie pozytywnie zweryfikowanych danych działek we wniosku tworzony jest rejestr działek, ponadto system musi zweryfikować dane beneficjenta z danymi z ewidencji podmiotów. Działki w rejestrze są już niezależne od wniosku. Usunięcie wniosku nie skutkuje usunięciem działki z rejestru, zatem identyfikuje się dodatkowy Plik Danych reprezentujący dane działki w rejestrze. Identyfikuje się także Plik Danych Podmiot, reprezentujący dane beneficjenta w ewidencji podmiotów, które są niezależne od danych beneficjenta we wniosku.*

Dane asocjacyjne, tzn. dane, które określają związek pomiędzy dwoma lub więcej Plikami Danych, należy traktować jako:

- odrębny Plik Danych, jeżeli są tworzone odrębnie niż Pliki Danych, które wiążą,
- część jednego z wiązanych Plików Danych, jeżeli są tworzone jednocześnie z danym Plikiem Danych.

**Przykład 4.2.1.C:** *Wymaganie z przykładów 4.2.1.A i 4.2.1.B rozszerzono w taki sposób, że system musi prezentować informację o dacie i o tym, na podstawie którego wniosku została utworzona dana działka w rejestrze. System musi przechowywać dane asocjacyjne pomiędzy działką a wnioskiem,*



które dodatkowo zawierają datę przeprowadzenia weryfikacji, w wyniku której działka została utworzona w rejestrze. Dane te stają się częścią Pliku Danych reprezentującego działkę przechowywaną w rejestrze, bowiem powstają w momencie umieszczenia działki w rejestrze.

**Przykład 4.2.1.D:** Wymagania z przykładów od 4.2.1.A do 4.2.1.C rozszerzono w taki sposób, że system musi umożliwiać przypisywanie do wniosku użytkownika, który będzie przeprowadzał jego obsługę. Obsługa może być realizowana w różnym zakresie (np. kontrola działek, kontrola beneficjenta itp.). Przypisując użytkownika do wniosku należy określić zakres, w jakim użytkownik ma obsługiwać wniosek. W systemie identyfikowany jest odrębny Plik Danych reprezentujący dane o powiązaniu wniosku z użytkownikiem, ponieważ powiązanie to jest tworzone niezależnie, gdy w systemie już istnieją dane wniosku i użytkownika.

Plik Danych musi być unikatowy, tzn. nie można identyfikować dwóch odrębnych Plików Danych, jeżeli zgodnie ze Specyfikacją grupują one te same dane, w ten sam sposób (mają analogiczną strukturę) oraz istnieją funkcjonalności, które traktują te dane w taki sam sposób, tzn. nie posiadają logiki, która by działała inaczej dla jednych i inaczej dla drugich danych.

**Przykład 4.2.1.E:** Oprogramowanie umożliwia wprowadzanie wniosków oraz korekt wniosków. Wnioski oraz ich korekty zawierają dokładnie takie same dane, a wprowadzanie wniosku i korekty jest realizowane dokładnie tak samo. System odróżnia wniosek od korekty po tym, jaki zostanie zaznaczony cel złożenia dokumentu. W systemie identyfikowany jest jeden Plik Danych reprezentujący zarówno dane wniosku, jak i korekty wniosku, ponieważ ich dane są takie same oraz występuje funkcjonalność wprowadzania, która traktuje wnioski i korekty tak samo.

**Przykład 4.2.1.F:** Oprogramowanie umożliwia wprowadzanie wniosków oraz korekt wniosków. W zakresie danych korekty znajduje się nr wniosku, do którego składana jest korekta, oraz uzasadnienie składania wniosku. Nie wszystkie dane wniosku można skorygować, zatem funkcja wprowadzania korekty umożliwia wprowadzanie tylko wybranych danych wniosku. Funkcjonalność wprowadzania nie jest tu wystarczającą przesłanką do wyodrębnienia jednego Pliku Danych jak w przykładzie 4.3.1.E.. Jeżeli system nie posiada innych funkcjonalności (Elementarnych Procesów), które w identyczny sposób będą traktowały wniosek i jego korektę, to powinny zostać zidentyfikowane dwa odrębne Pliki Danych: jeden reprezentujący dane wniosku, drugi - dane korekty wniosku.

**Uwaga:** Zasada unikatowości nie wyklucza występowania, poza funkcjonalnością traktującą dany Plik Danych w taki sam sposób, funkcjonalności przetwarzającej Plik Danych na kilka alternatywnych sposobów, a także funkcjonalności, które będą miały zastosowanie do podzbioru instancji danego Pliku Danych.

**Przykład 4.2.1.G:** Oprogramowanie z przykładu 4.2.1.E posiada funkcjonalność kontroli danych, która jest inna w przypadku wniosku i korekty. W systemie nadal identyfikowany jest jeden Plik Danych reprezentujący zarówno dane wniosku, jak i korekty, mimo występowania alternatywnych funkcji kontroli, gdzie jedna ma zastosowanie dla wniosków, a druga dla korekt, ponieważ warunki, o których mowa w przykładzie 4.2.1.E, nadal są spełnione.

Dla każdego Pliku Danych można określić jego DETy.

DETy Pliku Danych to jego atrybuty, które są rozpoznawalne dla użytkownika i odróżniane przez logikę biznesową Elementarnych Procesów.

DETy Pliku Danych muszą być unikalne w ramach Pliku Danych, tzn. DET nie może zawierać w sobie informacji, która jest zawarta w innym DET danego Pliku Danych.

DETy Pliku Danych muszą być niezależnymi atrybutami danych, tzn. muszą być niezależnie zapisywane lub odczytywane w ramach Elementarnych Procesów oprogramowania. Dane, które są zawsze odczytywane lub zapisywane razem, są nieodróżnialne dla Elementarnych Procesów i nie mogą być identyfikowane jako odrębne DETy danego Pliku Danych.

**Przykład 4.2.1.I:** *System umożliwia przechowywanie adresu beneficjentów. Zgodnie z wymaganiami system musi wyróżniać w danych adresowych kod pocztowy, miejscowość oraz tzw. drugą linię adresową, w której, może być zapisany nr ulicy, nr domu oraz nr mieszkania. Plik Danych reprezentujący adres będzie zawierał 3 DETy: kod pocztowy, miejscowości i drugą linię adresową. Nie identyfikuje się odrębnie DETów reprezentujących nazwę ulicy, nr domu oraz nr mieszkania, ponieważ funkcjonalność systemu nigdzie nie korzysta indywidualnie z tych danych, zawsze posługuje się danym scalonymi w postaci drugiej linii adresowej.*

**Przykład 4.2.1.J:** *Wymiarowane oprogramowanie przechowuje i przetwarza zawartość plików w całości, umożliwia przesłanie plików do repozytorium, pobranie, zmianę ich nazwy oraz lokalizacji. Pliki zawierają dane podmiotów opisujące m.in. ich nazwę, dane identyfikacyjne i adresowe w formacie XML, które nie są przetwarzane przez dane oprogramowanie. Po stronie wymiarowanego oprogramowania identyfikujemy 1 ILF reprezentujący przechowywany plik, a cała zawartość pliku stanowi 1 DET tego ILF. Przykładami innych DETów tego ILFa są: nazwa pliku, folder w którym występuje, data utworzenia, data ostatniej zmiany itp.*

Jeżeli wymagania Użytkownika wskazują, że Plik Danych powinien zawierać informacje o swoich związkach z innymi Plikami Danych (związki te go definiują), to atrybuty określające te związki z Plikami Danych są także DETami tego Pliku Danych. W przypadku związków jednokierunkowych DETy występują tylko po stronie Pliku Danych, w którym występuje informacja wskazująca na powiązany Plik Danych. W przypadku związków dwukierunkowych DETy występują w obydwu Plikach Danych.

**Przykład 4.2.1.K:** *Oprogramowanie umożliwia wskazanie jednostki organizacyjnej, w której obsługiwany będzie dany wniosek. W wyniku wykonania tej funkcji zostanie utworzony związek jednokierunkowy, w danych wniosku będzie występować wskazanie na jednostkę organizacyjną, w której będzie obsługiwany. Jednostka organizacyjna nie będzie zawierać informacji o wnioskach, które będą w niej obsługiwane. W Pliku Danych reprezentującym wniosek będzie występował DET reprezentujący wskazania na Plik Danych reprezentujący jednostkę organizacyjną. W Pliku Danych reprezentującym jednostkę organizacyjną nie będzie występował DET reprezentujący wskazanie na Plik Danych reprezentujący obsługiwane wnioski.*

**Przykład 4.2.1.L:** *W wyniku obsługi wniosku generowane jest wezwanie do złożenia korekty. We wniosku przechowywane są informacje o tym, jakie wezwania zostały wygenerowane na podstawie wniosku, a w wezwaniu przechowywane są informacje o tym, na podstawie jakiego wniosku zostało wygenerowane dane wezwanie. W Pliku Danych reprezentującym wniosek będzie występował DET reprezentujący wskazania na Plik Danych reprezentujący wezwanie, a w Pliku Danych reprezentującym wezwanie będzie występował DET reprezentujący wskazanie na Plik Danych reprezentujący wniosek, na podstawie którego wygenerowano dane wezwanie.*

Pojedyncze DETy mogą przyjmować wartości puste.

**Przykład 4.2.1.M:** *We wniosku występuje pole uwagi, które nie musi być wypełniane. Plik Danych reprezentujący wniosek zawiera DET reprezentujący pole uwagi. Wartość tego DETa będzie dla części wystąpień Pliku Danych przyjmować wartości puste.*

Pojedyncze DETy mogą przyjmować jednocześnie wiele wartości (wartością jest zbiór wartości).

Dotyczy to także DETów reprezentujących związki z innymi Plikami Danych o licznosci 1:n (jeden do wielu) lub n:n (wiele do wielu). Wartością tych DETów mogą być wskazania na wiele powiązanych Plików Danych.

**Przykład 4.2.1.N:** *We wniosku można zaznaczyć projekty, których wniosek dotyczy. Wniosek może dotyczyć wielu projektów. Plik Danych reprezentujący wniosek zawiera pojedynczy DET reprezentujący projekty, który będzie zawierał listę projektów, których dotyczy.*

DETy Pliku Danych grupowane są w RETy. RETy to podgrupy DETów danego Pliku Danych, które spełniają jeden z poniższych warunków:

- 1) jest ich więcej niż 1 i wszystkie występują w relacji 1:1 tylko względem siebie,
- 2) jest ich więcej niż 1 i stanowią dane asocjacyjne z innym Plikiem Danych lub Plikami Danych,
- 3) jest ich więcej niż 1 i stanowią dane specyficzne dla specjalizacji (podtypu) danego Pliku Danych.

Plik danych posiada zawsze co najmniej jeden RET.

**Przykład 4.2.1.O:** *Dane faktury obejmują dane sprzedawcy, kupującego, datę wystawienia i sprzedaży oraz dane pozycji faktury. Plik Danych reprezentujący fakturę będzie posiadał 2 RETy: jeden reprezentujący dane podstawowe faktury i drugi reprezentujący dane pozycji faktury. Wszystkie DETy reprezentujące dane podstawowe występują względem siebie w relacji 1:1, analogicznie wszystkie dane pozycji faktury również występują względem siebie w relacji 1:1. DETy reprezentujące dane podstawowe występują względem DETów danych pozycji faktury w relacji 1:n dlatego są grupowane w odrębnych RETach.*

**Przykład 4.2.1.P:** *Wniosek zawiera w sobie sekcję A, B, C i D. Wszystkie sekcje występują tylko raz we wniosku. Plik Danych reprezentujący wniosek będzie zawierał jeden RET reprezentujący wszystkie dane wniosku, bo wszystkie dane występują względem siebie w relacji 1:1. Wyodrębnienie sekcji nie jest podstawą do wyodrębnienia RETów.*

**Przykład 4.2.1.R:** *Plik Danych reprezentujący działkę z przykładu 4.2.1.C zawiera 2 RETy: jeden zawierający dane działki i drugi reprezentujący dane asocjacyjne z wnioskiem, który oprócz wskazania na wniosek (klucz obcy) zawiera także datę przeprowadzenia weryfikacji wniosku, na podstawie której działkę umieszczono w rejestrze.*

**Przykład 4.2.1.S:** *Podmiot obsługiwany w systemie jest powiązany z podmiotami, które może reprezentować. Relacja reprezentacji nie posiada innych atrybutów poza wskazaniem na inny podmiot. Plik Danych reprezentujący podmiot posiada jeden RET, który będzie obejmował atrybut reprezentujący powiązanie z innym podmiotem. Dla danych asocjacyjnych, nie zawierających innych danych niż wskazanie na Plik Danych, nie identyfikuje się odrębnego RETa.*

**Przykład 4.2.1.T:** *Podmiot obsługiwany w systemie posiada dwie specjalizacje (podtypy): osoba oraz firma. Danymi specyficznymi dla osoby są imię i nazwisko oraz PESEL, danymi specyficznymi dla firmy są nazwa oraz REGON. Plik Danych reprezentujący podmiot posiada co najmniej 3 RETy: jeden obejmuje dane wspólne dla osoby i firmy np. NIP, drugi obejmuje dane specyficzne dla osoby, a trzeci dane specyficzne dla firmy.*

---

#### 4.3.2 IDENTYFIKACJA ELEMENTARNYCH PROCESÓW

Identyfikacji Elementarnych Procesów dokonuje się na podstawie analizy logiki biznesowej i zachowania systemu niezbędnych do jego działania zgodnie ze Specyfikacją. Identyfikacja Elementarnych Procesów polega na wyodrębnieniu elementarnych, niezależnych i unikatowych aktywności systemu, które w pełni realizują cel Użytkownika wynikający ze specyfikacji.

Celem Użytkownika może być wywołanie zmiany w systemie związanej z aktualizacją Plików Danych lub uzyskanie z systemu informacji. Cel Użytkownika musi mieć znaczenie z punktu widzenia procesu biznesowego realizowanego przez Użytkownika.

Wywołanie zmiany w systemie w wyniku Elementarnego Procesu musi zapewniać prawidłowe działanie innych Elementarnych Procesów, nie może powodować nieprawidłowego działania innych Elementarnych Procesów.

Uzyskane przez Użytkownika w ramach pojedynczego Elementarnego Procesu informacje muszą być kompletne i spójne, i mieć praktyczne znaczenie dla Użytkownika.

Elementarny Proces to ciąg aktywności systemu obejmujący interakcję z Użytkownikiem, który zawsze inicjowany jest przez Użytkownika, a kończy się w momencie osiągnięcia celu przez Użytkownika lub jego przerwania.

Elementarny Proces ma charakter transakcji. Przerwanie Elementarnego Procesu oznacza, że cel Użytkownika nie został osiągnięty, a stan systemu jest taki jak przed rozpoczęciem Elementarnego Procesu. Logika biznesowa przywracająca stan sprzed rozpoczęcia Elementarnego Procesu jest częścią danego Elementarnego Procesu.

Elementarne Procesy muszą być niezależne, Elementarny Proces musi realizować cel Użytkownika, bez konieczności uruchamiania innego Elementarnego Procesu, czyli nie może zależeć od wykonania

innego Elementarnego Procesu. Elementarne Procesy nie mogą logicznie wywoływać innych Elementarnych Procesów, aby zrealizować własny cel, muszą go realizować samodzielnie w całości.

Uwaga: Należy odróżnić zależność Elementarnych Procesów polegającą na zależności od danych, które są tworzone przez inne Elementarne Procesy, od zależności od bezpośredniego wykonania innego Elementarnego Procesu. Ta pierwsza zależność jest dopuszczalną i nieodzowną zależnością Elementarnych Procesów wspierających ten sam proces biznesowy. Ta druga jest wskazaniem, że zależne od siebie Elementarne Procesy stanowią elementy tego samego Elementarnego Procesu.

**Przykład 4.2.2.A:** *System umożliwia wprowadzanie wniosku oraz wygenerowanie decyzji na podstawie wniosku. Wygenerowanie decyzji korzysta z danych, które zostały utworzone przez wprowadzanie wniosku, lecz obydwie funkcjonalności wykonują się niezależnie. W systemie identyfikowane są dwa Elementarne Procesy: jeden reprezentuje wprowadzanie wniosku, a drugi - wygenerowanie decyzji.*

**Przykład 4.2.2.B:** *System umożliwia wprowadzanie wniosku oraz dokonuje walidacji wprowadzanych danych. Dopuszcza zapisanie danych, które spełniają określone kryteria. Celem wprowadzania wniosku jest utworzenie wniosku na podstawie danych wprowadzonych przez Użytkownika. Walidacja danych jest wykonywana po to, aby poprawnie zrealizować wprowadzanie wniosku, poza kontekstem wprowadzania i zapisu danych wniosku nie ma sensu. Wprowadzenie wniosku nie będzie kompletne, jeżeli nie będzie obejmować walidacji wniosku. Wprowadzanie, walidacja oraz zapis danych stanowią element jednego Elementarnego Procesu reprezentującego wprowadzanie wniosku.*

**Przykład 4.2.2.C:** *System umożliwia wprowadzanie wniosku, niezależnie umożliwia wykonanie sprawdzenia wniosku. Sprawdzenie może być uruchomione przez użytkownika po zapisaniu wniosku i skutkuje wyświetleniem ewentualnych nieprawidłowości wykrytych w danych wniosku. Zapisywane dane mogą zawierać nieprawidłowości. Celem wprowadzania wniosku jest utworzenie wniosku na podstawie danych wprowadzonych przez Użytkownika. Sprawdzenie danych nie jest konieczne, aby poprawnie zrealizować wprowadzanie wniosku. Sprawdzanie danych realizuje odrębny cel Użytkownika: dostarcza mu danych o nieprawidłowościach w danych zapisanego w systemie wniosku. Wprowadzanie wniosku i sprawdzenie danych wniosku to dwa niezależne Elementarne Procesy.*

**Przykład 4.2.2.D:** *System na życzenie użytkownika dokonuje weryfikacji wniosku, w wyniku weryfikacji ma określić, czy wniosek został złożony w terminie, czy z opóźnieniem, jeżeli z opóźnieniem, to ile ono wynosi, czy dane beneficjenta są zgodne z danymi w rejestrze podmiotów. Określenie czy wniosek został złożony w terminie, czy z opóźnieniem, a także określenie wielkości opóźnienia, stanowi ten sam Elementarny Proces. Informacje te wzajemnie od siebie zależą, stanowią różne wyniki tego samego sprawdzenia. Stwierdzenie, że wniosek został złożony w terminie, oznacza automatycznie, że nie został złożony z opóźnieniem i odwrotnie, stwierdzenie niezerowej wielkości opóźnienia oznacza, że został złożony z opóźnieniem, a nie został złożony w terminie. Określenie czy dane beneficjenta są zgodne z danymi w rejestrze podmiotów, jest odrębnym Elementarnym Procesem, wynik tego sprawdzenia jest niezależny od wyników badania terminowości.*

Użytkownik może jednocześnie zainicjować wiele Elementarnych Procesów, ale każdy z nich musi realizować swój odrębny cel. Procesy te mogą być uruchomione jednocześnie w momencie zainicjowania lub w wyniku wystąpienia określonych zdarzeń. Zdarzenia te mogą być wygenerowane przez inne Elementarne Procesy.

**Przykład 4.2.2.E:** *Codziennie w godzinach nocnych system hurtowni danych aktualizuje dane podmiotu na podstawie systemu źródłowego. Aktualizacja przebiega kaskadowo, najpierw aktualizowany jest obraz danych źródłowych podmiotu w warstwie ODS, na podstawie systemów źródłowych, następnie na ich podstawie dane podmiotu w warstwie HD, a na ich podstawie dane podmiotu w warstwie OLAP. W systemie identyfikuje się Elementarne Procesy, które odpowiadają aktualizacji danych podmiotu w warstwie ODS, aktualizacji danych podmiotu w warstwie HD oraz aktualizacji danych podmiotu w warstwie OLAP. Choć logicznie wszystkie są inicjowane przez użytkownika raz, to pierwszy uruchamiany codziennie o północy, drugi po zakończeniu pierwszego, a trzeci po zakończeniu drugiego.*

**Przykład 4.2.2.F:** *W przykładzie 4.2.2.C system po zapisaniu danych automatycznie uruchamia sprawdzenie danych wniosku. Zarówno wprowadzanie wniosku, jak i sprawdzenie są zainicjowane jednocześnie przez użytkownika. Wprowadzanie uruchamia się w momencie zainicjowania, a sprawdzenie po wystąpieniu zdarzenia polegającego na zakończeniu wprowadzania wniosku. Jednak sposób uruchamiania nie wpływa na identyfikację procesów, będą one takie same jak w opisano w przykładzie 4.2.2.C.*

**Przykład 4.2.2.G:** *W przykładzie 4.2.2.D wszystkie sprawdzenia mogą być uruchamiane jednocześnie przez użytkownika. Z punktu widzenia użytkownika system wykonuje je jednocześnie lecz nie wpływa to na liczbę zidentyfikowanych procesów, będzie taka sama jak w opisano to w przykładzie 4.2.2.D.*

Ten sam Elementarny Proces może być różnie wywoływany w wielu miejscach w systemie.

**Przykład 4.2.2.H:** *System umożliwia przeglądanie danych podmiotu. Funkcję przeglądu można uruchomić z poziomu listy wyszukanych podmiotów, a także z poziomu wniosku, po naciśnięciu przycisku umożliwiającego podgląd beneficjenta. W każdym przypadku funkcja przeglądu będzie działać tak samo. W systemie identyfikuje się pojedynczy Elementarny Proces, mimo że wywoływany jest w dwóch różnych miejscach, na dwa różne sposoby.*

Przebiegi oraz efekty działania Elementarnego Procesu mogą się od siebie różnić, Elementarny Proces może mieć różne warianty, które mogą zależeć m.in. od:

- wartości wprowadzanych danych (**Przykład 4.2.2.I:** *EP wprowadzania wniosku może mieć różny przebieg, jeżeli wprowadzona data jest wcześniejsza, a inny jeżeli jest późniejsza od bieżącej, jeżeli dane wniosku są kompletne, a inny jeżeli są niekompletne, mimo różnic identyfikowany jest pojedynczy EP wprowadzanie wniosku*),
- sposobu uruchomienia Elementarnego Procesu (**Przykład 4.2.2.J:** *wprowadzanie danych wniosku uruchamiane z poziomu listy typów wniosku, powoduje, że pole typ wniosku w formularzu wniosku jest domyślnie wypełnione wartością typu, wprowadzanie*



uruchamiane z menu głównego nie ma żadnych wartości domyślnych, mimo tych różnic w systemie identyfikowany jest pojedynczy EP wprowadzania wniosku),

- uprawnień Użytkowników inicjujących proces (**Przykład 4.2.2.K:** w przeglądzie wniosków, będzie dostępna opcja przeglądu danych beneficjenta tylko dla użytkowników posiadających uprawnienia do przeglądania podmiotów, mimo że funkcjonalność przeglądu będzie działała inaczej dla użytkowników o różnych uprawnieniach to identyfikowany jest pojedynczy EP przeglądu wniosku),
- danych zapisanych w systemie w momencie wykonania EP (**Przykład 4.2.2.L:** kontrola krzyżowa wniosku sprawdza, czy dany beneficjent złożył inny wniosek, przebieg kontroli będzie się różnił, w zależności od tego czy w systemie jest zapisany inny wniosek beneficjenta, niemniej jednak identyfikowany jest pojedynczy EP kontroli krzyżowej wniosku),
- sposobu prezentowania różnych wartości danych (**Przykład 4.2.2.M:** funkcjonalność przegląd wniosku ukrywa pola wniosku, które nie zostały wypełnione, przez to widoczny zakres informacyjny wniosku różni się w zależności od tego, które pola wniosku są wypełnione, choć widoczny efekt działania przeglądu ma różne warianty, to identyfikowany jest pojedynczy EP przeglądu wniosku).

Czynności wykonywane w ramach Elementarnego Procesu polegają na wykonaniu jednej z form logiki biznesowej wymienionych w Tabeli 1.

**Tabela 1. Formy logiki biznesowej – składniki Elementarnego Procesu, które nie powinny być identyfikowane jako odrębne Elementarne procesy.**

Lp.	Forma logiki biznesowej
1	<b>Walidacja danych</b> <b>Przykład:</b> Przy wprowadzaniu nowego wniosku sprawdzane jest, czy wszystkie pola obowiązkowe są wypełnione.
2	<b>Wykonywanie obliczeń matematycznych</b> <b>Przykład:</b> Przy generowaniu raportu obliczana jest średnia wielkość deklarowanej powierzchni na podstawie wszystkich zatwierdzonych wniosków.
3	<b>Przekształcanie wartości</b> <b>Przykład:</b> Na podstawie deklarowanej powierzchni wyznaczana jest grupa, do której należy wniosek, gdzie przynależność do grupy jest definiowana przez przedział deklarowanej powierzchni.
4	<b>Filtrowanie i selekcjonowanie danych wg określonych kryteriów</b> <b>Przykład:</b> Na podstawie podanego przedziału dat ze wszystkich wniosków wyznaczany jest zbiór wniosków, których data złożenia mieści się w przedziale.
5	<b>Sprawdzenie czy spełnione są warunki</b> <b>Przykład:</b> Oprogramowanie inaczej przetwarza wnioski, które zostały złożone w przyjętym okresie składania wniosków, a inaczej wnioski złożone po tym okresie. Oprogramowanie

	<p>sprawdza, czy spełnione zostały warunki określonego wariantu przetwarzania wniosku i uruchamia go.</p> <p><b>Przykład:</b> 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.</p>
6	<p><b>Aktualizacja jednego lub więcej ILF</b></p> <p><b>Przykład:</b> Oprogramowanie zapisuje wprowadzone dane wniosku.</p>
7	<p><b>Odczytywanie jednego lub więcej ILF lub EIF</b></p> <p><b>Przykład:</b> Oprogramowanie wywołuje usługę w celu sprawdzenia, czy w systemie zewnętrznym występuje podmiot posiadający określony nr NIP (EIF).</p>
8	<p><b>Utworzenie danych poprzez przekształcenie istniejących danych</b></p> <p><b>Przykład:</b> Znak sprawy jest tworzony przez system na podstawie typu dokumentu inicjującego sprawę i daty utworzenia sprawy</p>
9	<p><b>Przygotowanie danych w celu ich dostarczenia Użytkownikowi</b></p> <p><b>Przykład:</b> Lista prezentowanych wniosków jest odpowiednio formatowana i wyświetlana użytkownikowi</p>
10	<p><b>Przyjmowanie danych wchodzących w Granice Oprogramowania</b></p> <p><b>Przykład:</b> System wyświetla formatkę, która umożliwia użytkownikowi wprowadzanie i zapisanie danych wniosku</p>
11	<p><b>Sortowanie i porządkowanie danych</b></p> <p><b>Przykład:</b> System sortuje wnioski wg daty złożenia wniosku</p> <p><b>Przykład:</b> System grupuje wnioski wg właściwości miejscowej</p>

Zidentyfikowane Elementarne Procesy należy sklasyfikować przypisując im jeden z trzech określonych przez metodę typów tj. EI, EO lub EQ. Klasyfikacji dokonuje się przez analizę celów EP. Elementarny Proces jest procesem typu EI, jeśli jego głównym celem jest aktualizacja jednego lub więcej Plików wewnętrznych (ILF), czyli dodanie, modyfikacja lub usunięcie danych.

**Przykład 4.2.2.N:** *Rejestracja dokumentu kancelaryjnego jest procesem EI, gdyż jej głównym celem jest aktualizacja ILF reprezentującego dokument kancelaryjny.*

Elementarny Proces jest procesem typu EO, jeśli jego głównym celem jest dostarczenie danych użytkownikowi z wykorzystaniem przynajmniej jednej z następujących form logiki biznesowej:

- 1) Wykonywanie obliczeń matematycznych
- 2) Aktualizacja jednego lub więcej ILF
- 3) Utworzenie danych

**Uwaga:** Wyznaczania liczby oraz numerów porządkowych rekordów lub innych elementów danych dostarczanych Użytkownikowi w ramach danego EP nie traktuje się ani jako wykonywania obliczeń matematycznych, ani jako utworzenia danych. Tego typu operacje należy zaliczać do przygotowania



danych w celu dostarczenia ich Użytkownikowi. Występowanie tego typu czynności w EP nie jest wystarczającą podstawą do klasyfikacji EP jako EO.

**Przykład 4.2.2.O:** *Utworzenie raportu na podstawie zarejestrowanych dokumentów kancelaryjnych jest Procesem Elementarnym typu EO, gdyż zawiera dane o liczbie zarejestrowanych dokumentów z podziałem na stanowiska rejestracji, które trzeba utworzyć na podstawie danych zapisanych w systemie dokumentów kancelaryjnych.*

Elementarny Proces jest procesem typu EQ, jeśli jego głównym celem jest dostarczenie użytkownikowi danych i nie jest typu EO.

**Przykład 4.2.2.P:** *Prezentacja listy dokumentów kancelaryjnych jest procesem typu EQ, gdyż prezentuje wyłącznie dane zapisane w systemie i nie korzysta z form logiki charakterystycznych dla procesu EO, tzn. nie wykonuje obliczeń matematycznych, nie aktualizuje ILF, nie tworzy żadnych danych, nie zmienia zachowania oprogramowania.*

Klasyfikacja Elementarnych procesów zależy od celów, jakim służą, oraz logiki przetwarzania, jaką w sobie zawierają. Zależności te określają poniższe tabele.

**Tabela 2. Cele Elementarnych procesów w zależności od ich klasyfikacji.**

Cel	EI	EO	EQ
Aktualizacja jednego lub więcej ILF	Główny cel	Cel wspomagający	Niedozwolone
Dostarczenie danych użytkownikowi	Cel wspomagający	Główny cel	Główny cel

**Tabela 3. Formy logiki biznesowej występujące różnych typach Elementarnych Procesów**

Forma logiki biznesowej	EI	EO	EQ
1. Walidacja danych	T	T	T
2. Wykonywanie obliczeń matematycznych	T	O*	N
3. Przekształcanie wartości	T	T	T
4. Filtrowanie i selekcjonowanie danych wg określonych kryteriów	T	T	T
5. Sprawdzenie czy spełnione są warunki	T	T	T
6. Aktualizacja jednego lub więcej ILF	O	O*	N
7. Odczytywanie jednego lub więcej ILF lub EIF	T	T	O
8. Tworzenie danych poprzez przekształcanie istniejących danych	T	O*	N
9. Przygotowanie danych w celu ich dostarczenia Użytkownikowi	T	O	O

10. Przyjmowanie danych wchodzących w Granice Oprogramowania	O	T	T
11. Sortowanie lub porządkowanie danych	T	T	T

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

**Przykład 4.2.2.R:** Dla oprogramowania zdefiniowano elementarny proces EP1 Rejestracja przyjęcia dokumentu. Proces ten zakwalifikowano jako EI, gdyż jego głównym zadaniem jest aktualizacja ILF przechowującego dane dokumentów. Zgodnie z Tabelą 3 w procesie typu EI zawsze musi wystąpić forma logiki biznesowej nr 6 oraz 10. Warunek ten jest spełniony, co wynika z tabeli 4.

**Tabela 4. Weryfikacja występowania różnych form logiki biznesowej w EP1**

Forma logiki biznesowej	EI	Weryfikacja
1. Walidacja danych	T	Tak - walidacja czy wszystkie obowiązkowe dane dokumentu są wprowadzone
2. Wykonywanie obliczeń matematycznych	T	Brak
3. Przekształcanie wartości	T	Brak
4. Filtrowanie i selekcjonowanie danych wg określonych kryteriów	T	Brak
5. Sprawdzenie czy spełnione są warunki	T	Tak - sprawdzanie czy nadawca jest zarejestrowany
6. Aktualizacja jednego lub więcej ILF	O	Tak - aktualizowany jest ILF dokumentów
7. Odczytywanie jednego lub więcej ILF lub EIF	T	Tak - odczytywany jest opcjonalnie EIF z danymi nadawców
8. Tworzenie danych poprzez przekształcanie istniejących danych	T	Tak - tworzony jest nowy dokument
9. Przygotowanie danych w celu ich dostarczenia Użytkownikowi	T	Tak - przygotowanie komunikatów
10. Przyjmowanie danych wchodzących w Granice Oprogramowania	O	Tak - wprowadzanie danych przy pomocy formularza
11. Sortowanie lub porządkowanie danych	T	Brak

Dla każdego EP można określić:

- FTR, czyli wskazać odniesienia do Plików Danych,
- DET, czyli określić jego atrybuty.

Przez odniesienia do Plików Danych należy rozumieć przypadki występowania w danym EP form logiki biznesowej, które aktualizują lub odczytują określony ILF lub EIF.

Jeżeli w ramach danego EP wielokrotnie odwołujemy się do tego samego Pliku Danych, to zliczamy to jako pojedyncze odniesienie do tego Pliku Danych.

DETy Elementarnego Procesu to unikatowe i mające znaczenie dla Użytkownika atrybuty danych przekraczające Granicę Oprogramowania w trakcie wykonywania danego EP od Użytkownika do oprogramowania lub z oprogramowania do Użytkownika.

Elementarny Proces zawsze jest inicjowany przez Użytkownika. Użytkownik inicjuje Elementarny Proces przesyłając do oprogramowania dane niezbędne do uruchomienia Elementarnego Procesu.

Dane inicjujące składają się co najmniej z jednego DETa.

Liczba DETów reprezentujących dane inicjujące nie zależy od liczby sposobów i miejsc uruchamiania danego EP w oprogramowaniu.

**Przykład 4.2.2.S:** *W przypadku funkcjonalności wprowadzania nowego wniosku danymi inicjującymi są wprowadzone dane wniosku, informacja ta jest niezbędna do walidacji danych lub zapisania danych w ILF w ramach danego EP. W tym przypadku dane inicjujące są reprezentowane przez wszystkie wprowadzane DETy wniosku.*

*W przypadku modyfikacji danych wniosku lub usunięcia wniosku, danymi inicjującymi jest wskazanie na wniosek, który ma zostać zmodyfikowany lub usunięty, informacja ta jest niezbędna do pobrania wniosku w celu jego modyfikacji lub usunięcia i jest reprezentowana przez 1 DET. W przypadku funkcjonalności wyświetlenia listy wszystkich wniosków, danymi inicjującymi jest sygnał inicjujący odczyt danych wniosku, który będzie reprezentowany przez 1 DET. Listę wniosków można uruchomić z poziomu menu, a także z poziomu formularza do wprowadzania wniosku. Nie wpływa to na liczbę DETów reprezentujących sygnał inicjujących EP wyświetlania listy wniosków.*

Jeżeli w trakcie wykonania danego EP generowane są komunikaty dla Użytkownika, to są one reprezentowane przez pojedynczy DET danego EP.

Nie należy traktować jako DETów Elementarnych Procesów:

- opisów na formatkach, nagłówków 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 4.2.2.T:** *Raport z liczby zarejestrowanych dokumentów kancelaryjnych jest tworzony na zadany dzień i prezentuje liczbę dokumentów zarejestrowanych w poszczególnych stanowiskach rejestracyjnych. Na raporcie jest także prezentowana aktualna wersja aplikacji. Z punktu widzenia użytkownika wersja aplikacji jest nieistotna, to pomocniczy parametr techniczny umieszczony w raporcie w celu weryfikacji błędów oprogramowania. Tym samym, DETami EP są:*

- *zadany dzień, na jaki ma być utworzony raport (sygnał inicjujący),*
- *wyświetlana data określająca dzień, na jaki został utworzony raport,*
- *liczba dokumentów,*
- *nazwa stanowiska.*

#### 4.3.3 WYLICZANIE ROZMIARU FUNKCJONALNEGO KONWERSJI DANYCH

Konwersją danych nazywamy wymagania, mające na celu utworzenie, modyfikację lub usunięcie danych, z których korzysta oprogramowanie, i które są przedmiotem zidentyfikowanych w ramach wymiarowanego oprogramowania Plików Danych.

Konwersja danych może być wykonywana jednorazowo lub zgodnie z określonym harmonogramem.

Przykładami konwersji danych są:

- migracja danych – utworzenie danych na podstawie danych źródłowych,
- inicjowanie struktur danych np. słowników, uprawnień – utworzenie nowych danych zgodnie z określonymi kryteriami wynikającymi z wymagań,
- oznaczanie (oflagowanie) danych – modyfikacja danych, w celu ich specjalnego traktowania przez oprogramowanie, np. czasowe wyłączenie z użycia,
- korekta lub usunięcie wskazanych danych z systemu.

Rozmiar Funkcjonalny konwersji danych jest równy Rozmiarowi Funkcjonalnemu funkcjonalności, która realizowałaby tę konwersję danych.

**Przykład 3.5.3.A:** *Migracja danych obejmuje przeniesienie danych z Wniosków zapisanych w systemie zewnętrznym do nowej aplikacji. Migrację tę można opisać przez grupę funkcji, która przekształca dane zapisane w systemie zewnętrznym i zapisuje je w aplikacji. Funkcje te identyfikowane są jako Elementarne Procesy typu EI, które przekształcają dane wejściowe, spoza Granicy Oprogramowania i zapisują je w aplikacji. Rozmiar Funkcjonalny tych Elementarnych Procesów będzie Rozmiarem Funkcjonalnym wyżej opisanej migracji.*

**Przykład 3.5.3.B:** *Wymaganie obejmuje aktualizację słownika jednostek organizacyjnych. Aktualizacja może zostać przeprowadzona przez Elementarny Proces typu EI, którego danymi wejściowymi będą nowe wartości słownikowe i który nadpisze poprzednie wartości słownikowe nowymi. Rozmiar Funkcjonalny tego Elementarnego Procesu będzie Rozmiarem Funkcjonalnym wyżej opisanego wymagania.*

Należy zaznaczyć, że celem wymagań polegających na konwersji danych, nie jest utworzenie w/w funkcjonalności. Utworzenie takiej funkcjonalności (skryptu) może być jednym ze sposobów realizacji takich wymagań, w innych przypadkach będzie to jedynie hipotetyczna funkcjonalność, która nigdy nie powstanie i będzie służyć jedynie jako Element Wyceny.

Jeżeli w wyniku zmiany funkcjonalności, konieczna jest konwersja danych polegająca na nadaniu wartości inicjalnych (np. pustych, domyślnych) niezbędnych do prawidłowego działania zmienionej funkcjonalności, to konwersja ta uznawana jest za element prac związanych z tą zmianą funkcjonalności i nie jest wymiarowana odrębnie.

**Przykład 3.5.3.C:** Wymaganie polega na rozszerzeniu wniosku o datę akceptacji. Wszystkie wnioski o statusie zaakceptowany muszą mieć tę datę wypełnioną. Przed zmianą w systemie występowały już wnioski w stanie zaakceptowany. Aby po zmianie funkcjonalność działała poprawnie, pole data akceptacji tych wniosków musi być wypełnione pewną wartością. Przyjęto, że dla tych wniosków datą akceptacji jest data ostatniej aktualizacji, która jest jednym z atrybutów wniosku. Realizacja wymagania będzie zatem obejmować konwersję polegającą na zapisaniu dla istniejących wniosków w stanie zaakceptowany daty akceptacji równej dacie ich ostatniej aktualizacji. Konwersja ta jest elementem realizacji wymagania związanego z rozszerzeniem wniosku o datę akceptacji i nie wyznacza się dla niej odrębnie Rozmiaru Funkcjonalnego.

#### 4.3.4 ROZMIAR FUNKCJONALNY ELEMENTÓW WYCENY

Rozmiar Funkcjonalny Elementów Wyceny wyznacza się jedynie na podstawie ich klasyfikacji w oparciu o tablicę przedstawioną w Tabeli 5. Rozmiar funkcjonalny Elementów Wyceny.

**Tabela 5. Rozmiar funkcjonalny Elementów Wyceny**

Klasyfikacja Elementu Wyceny	Rozmiar Funkcjonalny Elementu Wyceny
ILF	8
EIF	5,5
EI	4
EQ	4
EO	5

Uwaga: Sposób wyznaczania Elementów Wyceny uproszczono względem oryginalnej metody IFPUG i nie zależy on już od liczby DETów, FTRów oraz RETów. Nie ma potrzeby ich wyznaczania na potrzeby wyliczenia rozmiaru funkcjonalnego, jednakże identyfikacja tych Artefaktów IFPUG może okazać się konieczna w celu wykazania unikatowości Plików Danych i Elementarnych Procesów lub rozstrzygnięcia na jakie występujące w oprogramowaniu Pliki Danych lub Elementarne Procesy wpływa dana Zmiana.

#### 4.4 WYZNACZANIE ROZMIARU FUNKCJONALNEGO ZMIAN OPROGRAMOWANIA

Rozmiar Funkcjonalny Elementów Wyceny opisany w rozdziale 4.3.4 dotyczy wytworzenia nowego oprogramowania. W przypadku zmian oprogramowania Elementy Wyceny dotyczą przypadków:

- dodania nowych Plików Danych lub Elementarnych Procesów,
- usunięcia istniejących w oprogramowaniu Plików Danych lub Elementarnych Procesów,
- modyfikacji istniejących w oprogramowaniu Plików Danych lub Elementarnych Procesów.

Plik Danych uważany jest za zmodyfikowany, jeżeli spełniony jest co najmniej jeden z poniższych warunków:

- dodawane są do niego DETy,

- usuwane są z niego DETy,
- zmienia się struktura jego RETów.

Elementarny Proces uważany jest za zmodyfikowany, jeżeli spełniony jest co najmniej jeden z poniższych warunków:

- dodawane są do niego DETy
- dodawane są do niego FTRy,
- usuwane są z niego DETy,
- usuwane są z niego FTRy,
- zmienia się jego logika biznesowa.

W celu określenia rozmiaru zmiany oprogramowania należy zidentyfikować w/w Elementy Wyceny, a następnie wyznaczyć ich Rozmiar Funkcjonalny zgodnie z następującym wzorem:

$$RF_{EW} = BR_{FEW} \cdot W_{TZ}$$

gdzie:

- $RF_{EW}$  – Rozmiar Funkcjonalny Elementu Wyceny,
- $BR_{FEW}$  – bazowy Rozmiar Funkcjonalny wyznaczony na podstawie typu Pliku Danych lub Elementarnego Procesu wyznaczony zgodnie z Tabela 5. Rozmiar funkcjonalny Elementów Wyceny,
- $W_{TZ}$  – Współczynnik wynikający z typu zmiany wyznaczony zgodnie z Tabela 6. Współczynnik wynikający z typu zmiany

**Tabela 6. Współczynnik wynikający z typu zmiany**

Typ zmiany	Rozmiar Funkcjonalny Elementu Wyceny
Dodanie PD lub EP	1
Modyfikacja PD lub EP	0,5
Usunięcie PD lub EP	0,4

Rozmiar Funkcjonalny zmiany oprogramowania jest sumą Rozmiarów Funkcjonalnych, wyznaczonych zgodnie ze wzorem powyżej, Elementów Wyceny zidentyfikowanych w ramach Specyfikacji opisującej tę zmianę.

**Przykład 4.4.A:** Zmiana oprogramowania polega na modyfikacji procesów EQ przeglądu wniosku, EI wprowadzenia danych wniosku, EI modyfikacji danych wniosku, usunięciu procesu EO kontroli kwoty wniosku na podstawie danych referencyjnych udostępnianych przez system zewnętrzny (EIF) (dane te nie są wykorzystywane przez żaden inny proces) oraz dodaniu procesu EO kontroli podmiotów. Rozmiar zmiany będzie obliczany następująco:

$$\text{Przegląd danych wniosku - modyfikacja EQ} = 4 \text{ PF} * 0,5 = 2 \text{ PF}$$

$$\text{Wprowadzenie danych wniosku - modyfikacja EI} = 4 \text{ PF} * 0,5 = 2 \text{ PF}$$

$$\text{Edycja danych wniosku - modyfikacja EI} = 4 \text{ PF} * 0,5 = 2 \text{ PF}$$

$$\text{Kontrola kwoty - usunięcie EO} = 5 \text{ PF} * 0,4 = 2 \text{ PF}$$

Dane referencyjne – usunięcie EIF =  $5,5 \text{ PF} * 0,4 = 2,2 \text{ PF}$

Kontrola podmiotów – dodanie EO =  $5 \text{ PF} * 1 = 5 \text{ PF}$

w sumie: **15,2 PF**

## 5 ZASADY WYZNACZANIA ROZMIARU FUNKCJONALNEGO W ARIMR

ID reguły	IFPUG001: Identyfikacja Plików danych - wyłączenia
Opis reguły	<p>Plików Danych (czy to w całości, czy w części) nie stanowią:</p> <ol style="list-style-type: none"><li>1) Dane, które nie wynikają z opisu danej funkcjonalności.</li><li>2) Zbiory tymczasowe (robocze, kontrolne, częściowe, pośrednie itp.) tworzone pomocniczo tylko na czas wykonania EP.</li><li>3) Dane redundantne powielane w celu optymalizacji czy zapewnienia wydajności danej funkcjonalności (np. przechowywanie tych samych danych w różnych formatach, dane buforowane).</li><li>4) Dane definiujące logikę biznesową zgodnie z zasadami opisanymi w IFPUG009.</li></ol>
Przykłady	<p>A. <i>Funkcjonalność naliczania w celu określenia kwoty płatności wyznacza pewne wartości pośrednie. Jeżeli wartości te nie są przechowywane w systemie w celu wykorzystania ich przez inne, aktualnie występujące funkcjonalności, to nie traktuje się ich jako Plików danych.</i></p> <p>B. <i>Funkcjonalność obejmuje prezentację danych, których przygotowanie jest złożone obliczeniowo. Przy jej realizacji podjęto decyzję, że w celu optymalizacji wydajności, przetworzone dane są buforowane, w celu ich prezentacji przy kolejnym uruchomieniu funkcjonalności. Operacja ta jest przezroczysta dla użytkownika funkcjonalności, z punktu widzenia jego wymagań nie ma znaczenia, czy dane są buforowane, czy nie. Dane te są redundantne, tzn. powstają na podstawie innych danych. Danych tych nie traktuje się jako odrębnych Plików danych.</i></p>

ID reguły	IFPUG002: Identyfikacja i wymiarowanie EIF
Opis reguły	<p>EIF reprezentują interfejsy udostępniane przez zewnętrzne oprogramowanie, z których korzysta wymiarowane oprogramowanie i identyfikowane są wyłącznie z punktu widzenia wymiarowanego oprogramowania.</p> <p>Sposób identyfikacji EIF nie zależy od tego w jaki sposób są przygotowywane udostępniane dane przez oprogramowanie zewnętrzne, zależy tylko od tego w jaki sposób korzysta z nich wymiarowane oprogramowanie.</p> <p>Sposób identyfikacji EIF nie zależy od tego, z ilu i jakich Plików Danych po stronie oprogramowania zewnętrznego pochodzą udostępniane dane.</p> <p>DETy EIFów wymiarowanego oprogramowania mogą być zarówno bezpośrednio udostępnianymi DETami Plików Danych występujących po stronie</p>

	<p>oprogramowania zewnętrznego, jak i atrybutami dynamicznie wygenerowanymi przez to oprogramowanie zewnętrzne.</p> <p>Dane udostępniane przez pojedynczą usługę oprogramowania zewnętrznego, z której korzysta wymiarowane oprogramowanie, zawsze identyfikowane są jako pojedynczy EIF.</p> <p>Liczba przypadków, w których wymiarowane oprogramowanie korzysta z danego interfejsu (usługi) nie wpływa na liczbę identyfikowanych EIFów. Jeżeli wymiarowane oprogramowanie korzysta w wielu miejscach z tego samego interfejsu, to identyfikowany jest dokładnie jeden EIF.</p>
Przykład	<p>A. <i>Aplikacja zewnętrzna udostępnia interfejs danych podmiotu, które pochodzą z dwóch odrębnych PD zidentyfikowanych w ramach tej aplikacji: rejestru podmiotów i zgłoszeń podmiotów. Wymiarowana Aplikacja korzysta z w/w interfejsu w co najmniej dwóch EP: rejestracji wniosku oraz kontroli danych wniosku. Po stronie wymiarowanej Aplikacji identyfikowany jest 1 i tylko 1 EIF reprezentujący w/w interfejs.</i></p>

ID reguły	IFPUG003: Pliki danych – niezależność od sposobu realizacji
Opis reguły	Liczba identyfikowanych Plików Danych nie zależy od postaci, technologii przechowywania oraz formatu zarządzanych przez oprogramowanie danych. Podzbiór danych wyodrębniony jedynie ze względów technologicznych nie jest podstawą od wyodrębnienia Pliku Danych.
Przykład	<p><i>Wymiarowane oprogramowanie przechowuje załączniki mapowe w postaci plików graficznych oraz odpowiadających im wpisów w bazie danych. Po stronie wymiarowanego oprogramowania identyfikowany jest 1 Plik Danych (ILF) reprezentujący załączniki mapowe jako całość, ponieważ fakt przechowywania załączników zarówno w postaci plików jak i wpisów w bazie danych nie powoduje wyodrębnienia Plików Danych zarówno dla części plikowej jak i bazodanowej, obydwie części stanowią 1 Plik Danych.</i></p> <p><i>W celu optymalizacji funkcjonalności, wymiarowane oprogramowanie, w zależności od metody wprowadzania, raz zapisuje dane podmiotu w pliku XML, a innym razem w relacyjnej bazie danych. Po stronie wymiarowanego oprogramowania identyfikowany jest tylko 1 Plik Danych reprezentujący dane Podmiotu, ponieważ różne sposoby zapisu danych podmiotu wynikające z optymalizacji działania funkcjonalności nie mają wpływu na sposób identyfikacji Plików Danych</i></p>

ID reguły	IFPUG006: Import z pliku i eksport do pliku
Opis reguły	Plików przechowywanych w środowisku lokalnym Użytkownika, nie traktuje się jako elementów wymiarowanego oprogramowania, ani jako danych, do których wymiarowane oprogramowanie ma dostęp, w związku z powyższym przyjmuje się poniższe zasady.



	<p>W przypadku funkcjonalności importu danych z pliku znajdującego się w środowisku lokalnym Użytkownika (plik importu), plik ten nie jest identyfikowany ani jako EIF, ani jako ILF.</p> <p>Proces Funkcjonalny, którego celem jest import danych z pliku (EP importu) jest typu EI.</p> <p>Plik importu nie jest FTRem EP importu.</p> <p>Atrybuty danych zapisane w pliku importu, które są przetwarzane przez EP importu (np. zapisywane w Granicach Oprogramowania) są DETami EP importu.</p> <p>Jeżeli zawartość pliku importu jest zapisywana przez EP importu jako całość to cała zawartość pliku importu jest pojedynczym DETem EP importu.</p> <p>Zmiana formatu pliku importu jest zmianą EP importu.</p> <p>W przypadku funkcjonalności eksportu danych do pliku znajdującego się w środowisku lokalnym Użytkownika (plik eksportu), plik ten nie jest identyfikowany ani jako EIF, ani jako ILF.</p> <p>Proces Funkcjonalny, którego celem jest eksport danych do pliku (EP eksportu) jest typu EO lub EQ.</p> <p>Plik eksportu nie jest FTRem EP eksportu.</p> <p>Atrybuty danych zapisywane w pliku eksportu przez EP eksportu są DETami EP eksportu.</p> <p>Jeżeli EP eksportu nie przetwarza poszczególnych pól eksportu, a jedynie umożliwia pobranie zawartości pliku jako całości, to cała zawartość pliku eksportu jest pojedynczym DETem EP eksportu.</p> <p>Zmiana formatu pliku eksportu jest zmianą EP eksportu.</p>
Przykład	<p>A. Funkcjonalność umożliwia import pliku xml, który zawiera kompletne dane podmiotu, z którego pobierane są nazwa, NIP i forma prawna Podmiotu. Identyfikowany jest EP reprezentujący import pliku, który zawiera następujące DETy: nazwę, NIP, formę prawną Podmiotu, pozostałe atrybuty podmiotu zapisane w pliku nie są DETami danego EP.</p> <p>B. Funkcjonalność umożliwia eksport danych kontaktowych do pliku csv. W pliku zapisywane są: imię, nazwisko, nr telefonu, adres email, zapisane w systemie. Identyfikowany jest EP reprezentujący eksport pliku, który zawiera następujące DETy: imię, nazwisko, nr telefonu, adres email.</p> <p>C. Funkcjonalność umożliwia pobranie całego pliku xml przechowywanego w repozytorium sytemu. Identyfikowany jest EP reprezentujący pobranie pliku, który zawiera DET reprezentujący całą zawartość xml pliku, atrybuty przechowywane w strukturze xml nie są brane pod uwagę.</p> <p>D. Wymaganie polega na zmianie formatu pliku, do którego eksportowane są dane podmiotu zapisane w systemie. Elementem Wyceny jest zmiana EQ eksportu danych podmiotu i jej rozmiar wynosi <math>0,5 * 4 PF = 2 PF</math>.</p>

ID reguły	IFPUG009: Dane definiujące logikę biznesową
-----------	---

Opis reguły	<p>Dane definiujące logikę biznesową funkcjonalności nie mogą stanowić Pliku Danych, chyba że system udostępnia funkcjonalność przeznaczoną do bezpośredniego zarządzania tego typu danymi.</p> <p>Do danych definiujących logikę biznesową funkcjonalności należą:</p> <ul style="list-style-type: none"> <li>a) domyślne wartości atrybutów,</li> <li>b) definicje dopuszczalnych zakresów wartości,</li> <li>c) kryteria poprawności atrybutów,</li> <li>d) definicje formatów danych,</li> <li>e) szablony lub style definiujące wygląd interfejsu użytkownika</li> <li>f) szablony generowanych dokumentów lub raportów</li> <li>g) treści błędów, opisów i komunikatów</li> <li>h) dane słownikowe,</li> <li>i) definicje symbolizacji warstw GIS,</li> <li>j) definicje reguł biznesowych,</li> <li>k) parametry biznesowe,</li> <li>l) logi systemowe,</li> <li>m) uprawnienia użytkownika,</li> <li>n) dane uwierzytelniające użytkownika,</li> <li>o) sesje użytkownika.</li> </ul> <p>Odczytów i zapisów do Plików Danych reprezentujących dane definiujące logikę biznesową, dokonywanych w trakcie wykonania EP, nie zlicza się jako FTR danego EP, chyba że te EP reprezentują funkcjonalność przeznaczoną do bezpośredniego zarządzania tego typu danymi.</p> <p>Prezentacja i modyfikacja danych definiujących logikę biznesową, które ma na celu wspieranie działania innych EP, nie może być traktowana jako funkcjonalność przeznaczona do bezpośredniego zarządzania tymi danymi.</p>
Przykład	<p>A. Funkcjonalność rejestracji dokumentu korzysta ze słownika typów dokumentów. System <u>nie posiada</u> funkcjonalności edycji ani przeglądu typów dokumentów. Nie jest identyfikowany ILF reprezentujący słownik typy dokumentów. Tym samym EP rejestracji dokumentu nie zawiera FTR reprezentującego odczyt słownika typów dokumentu. Wyświetlanie listy typów dokumentów w celu umożliwienia ich wyboru w ramach EP rejestracji dokumentu nie jest funkcjonalnością przeznaczoną do bezpośredniego zarządzania danymi słownikowymi.</p> <p>B. Funkcjonalność rejestracji dokumentu korzysta ze słownika typów dokumentów. System <u>posiada</u> funkcjonalność edycji i przeglądu typów dokumentów. Identyfikowany jest ILF reprezentujący słownik typy dokumentów, a także EP edycji i przeglądu słownika typów dokumentów zawiera FTR reprezentujący odwołanie do słownika typy dokumentów. Natomiast EP rejestracji dokumentu nie zawiera FTR reprezentującego odczyt słownika typów dokumentu.</p> <p>C. Funkcjonalność rejestracji wniosku korzysta z atrybutu data złożenia wniosku. Istnieje reguła, która określa, w jakim przedziale powinna się mieścić data złożenia wniosku, aby wniosek mógł zostać zaakceptowany.</p>

	<p><i>System nie umożliwi ani edycji dopuszczalnego zakresu dat, ani edycji definicji reguły. Nie jest identyfikowany ani ILF reprezentujący dopuszczalny zakres dat, ani ILF reprezentujący definicję reguły.</i></p> <p><i>D. Oprogramowanie korzysta z danych pozwalających na wyznaczenie domyślnych wartości, z jakimi inicjalizowana jest sprawa – domyślnie kolejny z sekwencji, data utworzenia – domyślnie bieżąca, typ sprawy – domyślny dla typu dokumentu inicjującego. Danych tych nie można zmienić przy pomocy funkcjonalności oprogramowania. Danych tych nie traktuje się jako Pliku danych, ani jako części Pliku danych.</i></p> <p><i>E. Wymaganie definiuje funkcjonalność generowania decyzji zgodnie z określonym szablonem. Istnieje funkcjonalność pozwalająca na edycję tego szablonu. Identyfikuje się ILF reprezentujący szablon, EP reprezentujący edycję szablonu zawiera FTR reprezentujący odwołanie do szablonu. EP reprezentujący generowanie decyzji nie zawiera FTR reprezentującego odwołanie od szablonu.</i></p> <p><i>F. Wymaganie definiuje funkcjonalność wprowadzania lub modyfikacji danych dokumentów. Funkcjonalność ta odnotowuje każdy przypadek modyfikacji danych dokumentu w standardowym logu systemowym dla celów audytowych. Istnieje funkcjonalność przeglądu logu systemowego. Identyfikuje się ILF reprezentujący log systemowy. EP reprezentujący przegląd logu systemowego zawiera FTR reprezentujący log systemowy. EP reprezentujący wprowadzanie lub modyfikację dokumentów nie zawiera FTR reprezentującego odwołanie do logu systemowego.</i></p> <p><i>G. Wymaganie opisuje funkcjonalność zarządzania słownikiem jednostek organizacyjnych oraz instrumentów finansowych. Słownik jednostek organizacyjnych definiuje zależności pomiędzy jednostkami organizacyjnymi. Słownik instrumentów opisany jest przy pomocy specyficznego zbioru atrybutów, które określają między innymi rodzaj i wielkość pomocy finansowej. W przypadku jednostek organizacyjnych funkcjonalność ma umożliwiać definiowanie, usuwanie i modyfikację jednostek oraz zależności między nimi. W przypadku instrumentów finansowych funkcjonalność ma umożliwiać tworzenie i wersjonowanie nowych instrumentów. Z punktu widzenia funkcjonalności bazowej zostaną zidentyfikowane dwa odrębne Pliki danych, jeden reprezentujący jednostki organizacyjne, drugi - instrumenty finansowe.</i></p>
--	--

ID reguły	IFPUG010: Wymiarowanie zmian danych definiujących logikę biznesową
Opis reguły	Jeżeli wymiarowane oprogramowanie nie posiada funkcjonalności służącej do bezpośredniego zarządzania danymi definiującymi logikę biznesową EP to rozmiar Plików Danych reprezentujących te dane wynosi 0 (uznaje się za wliczony w rozmiar EP, których logikę te dane definiują). Analogicznie ze zmianami i usunięciem tych Plików Danych, ich rozmiar także wynosi 0 (uznaje się za wliczony w rozmiar zmian EP, których logikę te zmiany danych definiują).

	<p>Jeżeli wymaganie jednoznacznie określa zmiany danych definiujących logikę biznesową EP (np. słowników, szablonów), które mogą wpływać na działanie EP, których logikę te dane definiują, to wymaganie to należy wymiarować tak, jak EP konwersji tych danych, a nie zmiany EP, których logika jest definiowana przez te dane.</p> <p>W/w proces konwersji danych nie jest uznawany za funkcjonalność służącą do bezpośredniego zarządzania danymi definiującymi logikę biznesową.</p> <p>Jeżeli wymagania opisują zmiany konkretnych EP, a zmiana ta pociąga za sobą zmianę danych definiujących logikę biznesową tych EP, to wymagania te należy wymiarować jako zmianę tych EP, bez wymiarowania odrębnie procesu konwersji danych definiujących logikę biznesową tego EP. Wymiarowanie zmian EP obejmuje zmianę danych definiujących logikę biznesową tych EP.</p>
Przykład	<p><b>Funkcjonalności rejestracji, wyszukiwania i przeglądu dokumentu korzystają ze słownika typów dokumentów. System nie posiada funkcjonalności edycji ani przeglądu typów dokumentów.</b></p> <p>A. Rozmiar ILF reprezentującego słownik: typy dokumentów wynosi 0.</p> <p>B. Wymaganie polega na dodaniu nowego typu dokumentu, zmiana ta będzie widoczna z poziomu wszystkich funkcjonalności korzystających ze słownika typy dokumentów. Poza rozszerzeniem słownika nie określono żadnych specyficznych zmian w tych funkcjonalnościach. Przy wymiarowaniu tego wymagania należy uwzględnić dodanie EP konwersji słownika typy dokumentów, nie należy uwzględniać zmian EP funkcjonalności, na których działanie wpływa dany słownik. Wymaganie nie obejmuje dodania słownika: typy dokumentów, ponieważ występował już wcześniej w zakresie wymiarowanego oprogramowania.</p> <p>C. Wymaganie polega na zmianie funkcjonalności rejestracji dokumentów, która obejmuje m.in. zmiany opisów typów rejestrowanych dokumentów, co wymaga zmiany słownika typy dokumentów. Ponadto Wymaganie opisuje, że w/w zmiana słownika powinna być odpowiednio uwzględniona przez inne funkcjonalności korzystające z tego słownika. Przy wymiarowaniu tego Wymagania należy uwzględnić zmianę EP rejestracji dokumentów, a także dodanie EP konwersji słownika typy dokumentów, bowiem zmiana słownika powinna wpływać na wszystkie funkcjonalności, które korzystają z danego słownika i nie ogranicza się tylko do EP rejestracji dokumentów.</p> <p><b>Funkcjonalność generowania decyzji korzysta z szablonu decyzji.</b></p> <p>D. System <u>nie posiada</u> funkcjonalności edycji szablonu decyzji. W systemie <u>nie zidentyfikowano</u> ILF reprezentującego szablon decyzji. Wymaganie polega na zmianie funkcjonalności generowania decyzji, która obejmuje przy okazji zmianę szablonu decyzji. Zmiana szablonu decyzji nie wpływa na inne funkcjonalności, zmiana szablonu ma wpływ tylko na EP generowanie decyzji. Przy wymiarowaniu Wymagania należy uwzględnić</p>

	<p>tylko zmianę EP reprezentującego generowanie decyzji. Zmiana szablonu zawiera się całkowicie w zmianie tego EP. Rozmiar PD reprezentującego szablon wynosi 0.</p> <p>E. System <u>posiada</u> funkcjonalność edycji szablonu decyzji. W systemie <u>zidentyfikowano</u> ILF reprezentujący szablon decyzji. Wymaganie polega na zmianie funkcjonalności generowania decyzji, która obejmuje zmianę szablonu decyzji. Zmiana szablonu decyzji nie wpływa na pozostałe funkcjonalności, zmiana ogranicza się do jednego EP. Przy wymiarowaniu Wymagania należy uwzględnić tylko zmianę EP reprezentującego generowanie decyzji. Zmiana szablonu zawiera się całkowicie w zmianie EP generowanie decyzji. Ponieważ zmianie ulegają dane szablonu, a nie jego DETy czy RETy, w zakres zmiany nie wchodzi zmiana ILF reprezentującego szablon.</p> <p>F. Wymaganie polega na zmianie szablonu decyzji, która będzie miała wpływ na funkcjonalność generowania decyzji. Wymaganie nie określa bezpośrednio żadnych zmian funkcjonalności generowania decyzji. Przy wymiarowaniu Wymagania należy uwzględnić dodanie EP konwersji danych szablonu decyzji. Zmiana EP generowania decyzji będzie ograniczona tylko do zmian wynikających ze zmiany szablonu.</p>
--	---

ID reguły	IFPUG011: Niezależność zmian Elementarnych Procesów od zmian Plików Danych
Opis reguły	<p>DETy Plików Danych odczytywanych lub zapisywanych w ramach EP, nie stają się automatycznie DETami danego EP. Przy identyfikacji DETów EP mają znaczenie tylko dane przekraczające Granicę Oprogramowania.</p> <p>Na liczbę DETów EP nie ma wpływu z ilu Plików Danych zostały odczytane lub do ilu Plików Danych zostaną zapisane dane przekraczające Granicę Oprogramowania.</p> <p>Zmiana PD nie oznacza automatycznie zmiany EP, który odczytuje z niego dane lub zapisuje do niego dane. Zmiana EP powinna być zidentyfikowana tylko wtedy, kiedy zmienia się logika biznesowa tego EP związana z odczytem z danego PD lub zapisem do danego PD.</p>
Przykład	<p>A. Funkcjonalność umożliwia wprowadzanie identyfikatora dokumentu. System wyodrębnia z identyfikatora sygnaturę typu dokumentu. Typ dokumentu oraz identyfikator są zapisywane w odrębnych DETach ILF dokumentu. EP reprezentujący wprowadzanie identyfikatora posiada tylko 1 DET reprezentujący identyfikator, w którym zaszyty jest typ dokumentu, bo z punktu widzenia Użytkownika jest on wprowadzany, a zatem przekracza granicę oprogramowania jako pojedynczy atrybut.</p> <p>B. Wyniki wyszukiwania dokumentów mają postać tabeli zawierającej 3 kolumny: numer dokumentu, typ dokumentu oraz datę utworzenia. W celu wyszukania dokumentów system odczytuje dane z wielu Plików Danych reprezentujących dokumenty różnych typów. Numer dokumentu, typ dokumentu, a także data utworzenia odczytywane są więc z kilku różnych Plików Danych. Elementarny proces odpowiedzialny za</p>

	<p>wyszukanie dokumentów będzie posiadał tylko 3 DETy reprezentujące kolumny prezentowane Użytkownikowi, bez względu na to z ilu Plików Danych zostaną odczytane prezentowane w tych kolumnach wartości.</p> <p>C. Funkcjonalność wyznacza i prezentuje użytkownikowi sumę kwot przyznanych płatności dla danego beneficjenta, wynikającą z różnych decyzji. System musi odczytać dane z 3 Plików danych, reprezentujących różne typy decyzji, ponadto kwoty przyznanych płatności na decyzjach rozbite są na składniki, zapisane w różnych atrybutach. Elementarny Proces odpowiedzialny za wyznaczenie i prezentację sumy kwot przyznanych płatności będzie posiadał jeden DET reprezentujący sumę kwot przyznanych płatności, bez względu na to z ilu Plików Danych oraz atrybutów będzie odczytywał dane potrzebne do wyznaczenia tej sumy.</p> <p>D. W systemie występuje funkcjonalność prezentująca listę dokumentów, na której widoczne są identyfikator, data złożenia oraz nazwa beneficjenta, które odczytywane są z danych dokumentu. Wymaganie polega na rozszerzeniu danych dokumentu o nowy atrybut: oczekiwana data rozpatrzenia, która nie jest prezentowana na w/w liście. Przy wymiarowaniu w/w wymagania należy uwzględnić zmianę PD reprezentującego dokument, bo dodawany jest do niego DET oczekiwana data rozpatrzenia. Natomiast przy wymiarowaniu tego wymagania nie powinna być brana pod uwagę zmiana EP reprezentującego prezentację listy dokumentów, bowiem nie zmienia się logika biznesowa EP związana z odczytem PD dokument, odczytywane są wciąż te same dane, na liście dokumentów nie jest wyświetlany nowy atrybut.</p> <p>E. W systemie występuje funkcjonalność wprowadzania danych dokumentu, która zapisuje wprowadzone dane. Wymaganie polega na dodaniu nowego atrybutu: oczekiwana data rozpatrzenia, która nie jest wprowadzana (nie jest DETem tego EP), lecz jest automatycznie generowana i zapisywana w danych dokumentu. Przy wymiarowaniu Wymagania należy uwzględnić zmianę PD reprezentującego dokument, bo dodawany jest do niego DET oczekiwana data rozpatrzenia oraz zmianę EP reprezentującego wprowadzanie danych dokumentu, bowiem zmienia się logika biznesowa EP związana z zapisem dokumentu: generowany i zapisywany jest dodatkowy atrybut.</p>
--	--

ID reguły	IFPUG012: DETy Elementarnego Procesu, a interfejs użytkownika
Opis reguły	<p>Akcje udostępniane Użytkownikowi w trakcie wykonania danego EP, które umożliwiają uruchomienie czynności zmierzających do realizacji celu Użytkownika lub innych EP, są traktowane jak DETy danego EP.</p> <p>Logika biznesowa (m.in. odczytanie Pliku Danych), niezbędna do udostępnienia akcji umożliwiającej uruchomienie innego EP, należy do EP, w ramach którego jest ta akcja udostępniana, natomiast nie jest częścią uruchamianego w wyniku tej akcji EP.</p>

	<p>Tym samym zmiana tej logiki biznesowej (np. skutkująca zmianą sposobu uruchamiania EP) oznacza zmianę EP uruchamiającego inny EP, a nie zmianą uruchamianego EP.</p> <p>Wszystkie zmiany interfejsu użytkownika w ramach danego EP uznaje się za zmiany danego EP.</p> <p>Menu aplikacji, strona główna, stałe elementy stron internetowych np. nagłówki lub stopki, które obok stałych treści obejmują jedynie listę dostępnych dla Użytkownika operacji są specyficznymi EP typu EO, których DETami są poszczególne pozycje menu.</p> <p>Zmiana struktury lub konstrukcji menu oznacza zmianę tych specyficznych EP, lecz nie oznacza zmiany EP uruchamianych z poziomu tego menu.</p>
Przykład	<p>A. <i>Z poziomu każdej pozycji wyników wyszukiwania dokumentów (EQ Wyszukanie dokumentów) możliwe jest uruchomienie przeglądu dokumentu poprzez dwukrotne kliknięcie na danej pozycji lub kliknięcie przycisku „Przełóżaj” dostępnej dla każdej pozycji. EP Wyszukanie dokumentów zawiera jeden DET reprezentujący możliwość uruchomienia EP przegląd dokumentu, mimo, że w ramach procesu wyszukiwania można na dwa sposoby uruchomić proces przeglądu.</i></p> <p>B. <i>W funkcjonalności opisanej powyżej zmieniono opis przycisku uruchamiającego przegląd z „Przełóżaj” na „Podgląd”. Powyższa zmiana jest zmianą EP Wyszukanie dokumentów, zmiana nie jest zmianą EP Przegląd dokumentu.</i></p> <p>C. <i>W funkcjonalności przeglądu podmiotu możliwe jest uruchomienie funkcjonalności wyświetlenia listy wniosków, w których dany podmiot jest beneficjentem. EP Przegląd podmiotu obejmuje odczyt PD podmiot (odwołanie do PD podmiot jest FTRem EP przegląd podmiotu) oraz wyświetla wywołanie EP przegląd listy wniosków podmiotu, które zawiera w sobie identyfikator danego podmiotu. W momencie uruchomienia EP przegląd listy wniosków identyfikator ten jest przekazywany do tego EP przegląd listy wniosków, gdzie na jego podstawie odczytywane są dane z PD wniosek. EP wyświetlenie listy wniosków dla danego podmiotu nie ma potrzeby odczytywania danych PD podmiotu, ponieważ identyfikator podmiotu został mu przekazany przy uruchomieniu, jest jego DETem.</i></p> <p>D. <i>W funkcjonalności wyszukiwania wprowadzono maskę wprowadzania dla kryterium: numer sprawy. Przy wymiarowaniu tego wymagania identyfikuje się zmianę EP wyszukiwania.</i></p> <p>E. <i>W funkcjonalności wprowadzania wniosku zmieniono sposób wyświetlania komunikatów w taki sposób, aby były bardziej widoczne dla użytkownika i zwiły się po upływie 10 sekund. Przy wymiarowaniu tego wymagania brana jest pod uwagę zmiana EP wprowadzania wniosku.</i></p>

ID reguły	IFPUG013: Wymiarowanie interfejsów i usług
Opis reguły	<p>Widoki bazodanowe oraz usługi udostępniane przez oprogramowanie na potrzeby systemów zewnętrznych identyfikowane są jako EP typu EQ lub EO niezależnie od ich implementacji i użytej technologii.</p> <p>W przypadku widoków zmaterializowanych <b>nie</b> jest identyfikowany dodatkowo ILF reprezentujący zmaterializowane dane widoku, materializację danych widoku traktuje się jako cechę użytej technologii, a nie element wymagań funkcjonalnych dotyczących udostępnianych danych.</p> <p>Dla każdej wywoływanej przez wymiarowane oprogramowanie i udostępnione przez oprogramowanie zewnętrzne usługi identyfikuje się 1 EIF.</p> <p>Dla każdego udostępnianego przez oprogramowanie zewnętrzne widoku bazodanowego, z którego korzysta wymiarowane oprogramowanie identyfikowany jest 1 EIF.</p> <p>Sposób udostępniania i wywoływania usług nie wpływa na identyfikowane Elementy Wyceny.</p> <p>W obrębie szyny danych lub innego medium, za którego pośrednictwem udostępniane są usługi nie identyfikuje się Elementów Wyceny. Elementy Wyceny identyfikuje się tylko:</p> <ul style="list-style-type: none"> <li>• po stronie oprogramowania, które dostarcza usług i są to Elementy Wyceny związane z EP,</li> <li>• po stronie oprogramowania, które korzysta z usługi (jest konsumentem usługi) i są to Elementy Wyceny związane z EIF.</li> </ul> <p>Szyna danych znajduje się zawsze poza Granicą Oprogramowania.</p> <p>Jeżeli funkcjonalność oprogramowania realizowana jest jako formularz (interfejs użytkownika), który wywołuje określoną usługę oraz zarówno formularz, jak i usługa znajdują się wewnątrz granicy oprogramowania, a do komunikacji między nimi nie służy żaden zewnętrzny komponent (znajdujący się poza granicą oprogramowania) np. zewnętrzna względem modułu szyna danych, to formularz i usługę traktuje się jako tą samą funkcjonalność, nie definiuje się dla nich odrębnych EP.</p>
Przykład	<p>1. <i>Aplikacja Web umożliwia wprowadzanie danych za pomocą formularza: F1. Dane z formularza są zapisywane w systemie za pomocą usługi 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 jest wynikiem decyzji dotyczącej technologii realizacji funkcjonalności wprowadzania danych.</i></p> <p><i>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.</i></p>



	<p>2. Aplikacja A udostępnia na szynie danych usługę, która udostępnia nazwę oraz nr NIP podmiotu. Podmiot zawiera dodatkowo dane dotyczące roku utworzenia. Z usługi tej korzysta aplikacja B do generowania raportu zawierającego dane podmiotu.</p> <p>Przy wymiarowaniu aplikacji A:</p> <ul style="list-style-type: none"> <li>• identyfikowany jest ILF Dane podmiotu,</li> <li>• identyfikowany jest EQ Udostępnianie danych podmiotu.</li> </ul> <p>Przy wymiarowaniu aplikacji B:</p> <ul style="list-style-type: none"> <li>• identyfikowany jest EIF Dane podmiotu,</li> <li>• identyfikowany jest EO generowanie raportu.</li> </ul> <p>Po stronie szyny danych nie są identyfikowane żadne Elementy Wyceny.</p> <p>3. Aplikacja A udostępnia aplikacji B dane z wniosku i decyzji w postaci pojedynczego widoku bazodanowego. Wniosek posiada następujące pola: nr wniosku, nazwa podmiotu, data złożenia wniosku. Decyzja posiada następujące pola: nr decyzji, nazwa podmiotu, data decyzji, status decyzji. Decyzja jest powiązana z wnioskiem, którego dotyczy. Udostępniany widok bazodanowy posiada następujące pola: nr wniosku, nazwa podmiotu, data złożenia wniosku, nr decyzji, data decyzji, status decyzji. Aplikacja B wyświetla udostępniane przez aplikację A dane.</p> <p>Przy wymiarowaniu aplikacji A:</p> <ul style="list-style-type: none"> <li>• identyfikowany jest 1 ILF Wniosek,</li> <li>• identyfikowany jest 1 ILF Decyzja,</li> <li>• identyfikowany jest 1 EQ Udostępnianie danych wniosku i decyzji.</li> </ul> <p>Przy wymiarowaniu aplikacji B:</p> <ul style="list-style-type: none"> <li>• identyfikowany jest 1 EIF Udostępnione dane wniosku i decyzji,</li> <li>• identyfikowany jest 1 EQ Wyświetlanie danych wniosku i decyzji.</li> </ul>
--	---

ID reguły	IFPUG015:Wymiarowanie funkcjonalności szablonowej
Opis reguły	<p>Jeżeli opis funkcjonalności w całości opiera się na odwołaniu do innej, występującej w oprogramowaniu, konkretnej funkcjonalności, to przy określaniu Rozmiaru Funkcjonalnego Elementów Wyceny związanych z tą funkcjonalnością jest stosowany współczynnik korygujący o wartości 0,5.</p> <p>Funkcjonalność, do której odwołuje się opis funkcjonalności nazywana jest szablonem.</p> <p>Funkcjonalność, która odwołuje się do szablonu, nazywana jest funkcjonalnością szablonową.</p> <p>Jeżeli opis dla kilku różnych funkcjonalności jest wspólny, to przy określaniu Rozmiaru Funkcjonalnego Elementów Wyceny wszystkich tych funkcjonalności, poza jedną, wybraną, jest stosowany współczynnik korygujący o wartości 0,5.</p>

	<p>Wybrana funkcjonalność, dla której nie stosuje się współczynnika korygującego, nazywana jest szablonem. Pozostałe funkcjonalności, dla których stosuje się współczynnik korygujący nazywane są funkcjonalnościami szablonowymi.</p> <p>Jeżeli opis zmian dla kilku różnych funkcjonalności jest wspólny i ma taki sam wpływ na te funkcjonalności, to przy określaniu Rozmiaru Funkcjonalnego Elementów Wyceny dla wszystkich zmian tych funkcjonalności poza jedną wybraną, stosowany jest współczynnik korygujący o wartości 0,5.</p> <p>Wybrana zmiana funkcjonalności, dla której nie stosuje się współczynnika korygującego, nazywana jest szablonem. Pozostałe zmiany funkcjonalności, dla których stosuje się współczynnik korygujący nazywane są zmianami szablonowymi.</p> <p>Szablon i funkcjonalności/zmiany szablonowe mogą dotyczyć różnych zakresów przetwarzanych danych, jednak muszą mieć taki sam opis przetwarzania danych (reguł, logiki biznesowej). Jeżeli pewne funkcjonalności/zmiany dotyczą różnych zakresów przetwarzanych danych, z których wynikają inne reguły przetwarzania danych, reguły te powinny być opisane w Wymaganiach, a to oznacza, że pomiędzy tymi funkcjonalnościami/zmianami nie zachodzi relacja: szablon – funkcjonalność szablonowa i stosowanie współczynnika korygującego jest nieuzasadnione.</p> <p>Współczynnika korygującego nie stosuje się w przypadku wyznaczania Rozmiaru Funkcjonalnego związanego z usuwaniem funkcjonalności.</p>
Przykład	<p>A. Wymaganie polega na realizacji funkcjonalności akceptacji wniosku B, która działa tak samo jak funkcjonalność akceptacji wniosku A. Przy wymiarowaniu tego wymagania identyfikowany jest proces EI reprezentujący akceptację wniosku B, którego rozmiar wynosi: 4 PF (bazowy RF EI) * 0,5 (współczynnik korygujący) = 2 PF. Rozmiar wymagania wynosi 2 PF.</p> <p>B. Wymaganie polega na realizacji funkcjonalności podglądu decyzji A, B, C. Podgląd wszystkich decyzji powinien działać tak samo i prezentować wszystkie dane decyzji, funkcjonalności będą się różnić jedynie zakresem prezentowanych danych specyficznym dla każdej z decyzji. Przy wymiarowaniu tego wymagania identyfikowane są 3 procesy EQ, podgląd decyzji A zostaje wybrany jako szablon i jego rozmiar wynosi 4 PF (bazowy RF EQ), kolejne 2 są funkcjonalnościami szablonowymi, RF podglądu decyzji B wynosi 4 PF (bazowy RF EQ) * 0,5 (współczynnik korygujący) = 2 PF tak samo RF podglądu decyzji C wynosi 4 PF (bazowy RF EQ) * 0,5 (współczynnik korygujący) = 2 PF, w sumie RF wymagania wynosi 8 PF.</p> <p>C. Wymaganie polega na realizacji funkcjonalności podglądu decyzji A, B, C. Podgląd wszystkich decyzji powinien działać tak samo i prezentować wszystkie dane decyzji, jednak w przypadku decyzji C powinny być wyświetlone dodatkowo pewne dane z powiązanego z nią wniosku.</p>

	<p>Przy wymiarowaniu tego wymagania identyfikowane są 3 procesy EQ. W przypadku podglądu A i B opis realizacji jest wspólny, dlatego podgląd decyzji A zostaje wybrany jako szablon i jego rozmiar wynosi 4 PF (bazowy RF EQ), podgląd decyzji B jest funkcjonalnością szablonową RF procesu, który ją reprezentuje i wynosi 4 PF (bazowy RF EQ) * 0,5 (współczynnik korygujący) = 2 PF. Opis podglądu decyzji C obejmuje specyficzne dla tej decyzji reguły i jego RF wynosi 4 PF (bazowy RF EQ), w sumie RF wymagania wynosi 10 PF.</p> <p>D. Wymaganie polega na zmianie podglądu decyzji A, B, C, polegającej na ukryciu pola uwagi oraz data adnotacji występujących we wszystkich decyzjach.</p> <p>Przy wymiarowaniu tego wymagania identyfikowane są 3 zmiany procesów EQ, zmiana podglądu decyzji A zostaje wybrana jako szablon i jej RF wynosi 4 PF (bazowy RF EQ) * 0,5 (współczynnik związany z typem zmiany) = 2 P, kolejne 2 są zmianami szablonowymi, RF zmiany podgląd decyzji B wynosi 4 PF (bazowy RF EQ) * 0,5 (współczynnik związany z typem zmiany) * 0,5 (współczynnik korygujący) = 1 PF tak samo RF zmiany podglądu decyzji C wynosi 4 PF (bazowy RF EQ) * 0,5 (współczynnik związany z typem zmiany) * 0,5 (współczynnik korygujący) = 1 PF, w sumie RF wymagania wynosi 4 PF.</p>
--	--

ID reguły	IFPUG401: Wymiarowanie Procesów Wsadowych i procesów ETL
Opis reguły	<p>Dla każdego Procesu Wsadowego, który może wykonać się niezależnie od innych Procesów Wsadowych identyfikowany jest pojedynczy Proces Elementarny typu EI.</p> <p>Dla Procesów Elementarnych reprezentujących Procesy Wsadowe identyfikuje się:</p> <ul style="list-style-type: none"> <li>• 1 DET reprezentujący sygnał inicjujący <b>lub</b> 1 DET na każdy parametr Procesu Wsadowego sterujący wykonaniem danego procesu <b>oraz</b></li> <li>• 1 DET reprezentujący sygnał/komunikat zwrotny EP.</li> </ul> <p>Jeżeli Proces Wsadowy przetwarza dane przyrostowo, to w w/w EP identyfikuje się dodatkowo 1 DET reprezentujący parametr określający, które dane były do tej pory przetworzone np.: znacznik czasowy.</p> <p>Atrybuty danych przetwarzanych przez Proces Wsadowy nie są identyfikowane jako DETy w/w EP.</p> <p>FTRami w/w EP są PD reprezentujące dane przetwarzane przez Proces Wsadowy lub dane będące wynikiem przetwarzania Procesu Wsadowego.</p> <p>Procesy ETL są wymiarowane tak samo jak Procesy Wsadowe.</p>
Przykład	<p>A. <i>Celem procesu wsadowego P1 jest zmiana statusu na „nowy”, wszystkich obiektów klasy A, których status jest inny niż „zatwierdzony”. Dla każdej</i></p>

	<p>aktualizacji wykonanej zgodnie z powyższymi warunkami proces rejestruje obiekt klasy B.</p> <p>Identyfikowany jest pojedynczy EP typu EI odpowiadający P1, który posiada 2 DETy:</p> <ul style="list-style-type: none"> <li>• sygnał inicjujący,</li> <li>• komunikat zwrotny potwierdzający wykonanie procesu,</li> </ul> <p>oraz 2 FTRy reprezentujące klasę A i B.</p> <p>B. Proces ETL ładuje przyrostowo dane do tabeli sankcje, z danych źródłowych przechowywanych w tabelach sankcje A, sankcje B, sankcje C.</p> <p>Identyfikowany jest pojedynczy EP typu EI odpowiadający procesowi ETL, który posiada 3 DETy:</p> <ul style="list-style-type: none"> <li>• sygnał inicjujący,</li> <li>• znacznik pobranych danych,</li> <li>• komunikat zwrotny,</li> </ul> <p>oraz 4 FTRy reprezentujące ładowaną tabelę sankcje i 3 tabele źródłowe sankcje A, sankcje B, sankcje C.</p>
--	--

## 5.1 INTEGRACJA Z SYSTEMEM HURTOWNI DANYCH

Jeżeli oprogramowanie obejmuje przygotowywanie „migawek” na potrzeby hurtowni danych zawierających przyrosty danych lub różnice danych, to część oprogramowania odpowiedzialną za ten proces traktujemy jako odrębną Aplikację (Aplikacja STAGE) i dla niej ma zastosowanie reguła IFPUG402.

ID reguły	IFPUG402: Aplikacja STAGE
Opis reguły	<p>Jeżeli zgodnie ze Specyfikacją w Aplikacji STAGE wykonywana jest kopia danych źródłowych lub „migawka” zawierająca dane przyrostowe lub różnice danych to w ramach Aplikacji STAGE dla każdej tabeli źródłowej, dla której wykonywana jest kopia lub migawka identyfikuje się:</p> <ul style="list-style-type: none"> <li>• 1 EIF – reprezentujący tabelę źródłową,</li> <li>• 1 ILF – reprezentujący strukturę w której przechowywana jest kopia lub migawka,</li> <li>• 1 EI – reprezentujący proces ETL odpowiedzialny za kopiowanie lub wykonywanie migawki.</li> </ul> <p>W zakres w/w EI wchodzi także usuwanie danych ze struktury, w której przechowywana jest kopia lub migawka i nie jest ono wykazywane jako odrębny EP.</p> <p>W ramach Aplikacji STAGE nie są identyfikowane żadne inne Elementy Wyceny.</p>

	Przy określaniu Rozmiaru Funkcjonalnego Elementów Wyceny identyfikowanych w Aplikacji STAGE stosuje się współczynnik korygujący <b>0,4</b> .
Przykład	<p>A. Aplikacja STAGE pobiera dane z tabeli źródłowej Wniosek. Przy wymiarowaniu identyfikowany jest 1 EIF, 1 ILF oraz 1 EI.</p> <p>Rozmiar takiej zmiany wynosi: <math>(5,5PF (EIF) + 8 PF (ILF) + 4 PF (EI)) * 0,4</math> (współczynnik korygujący) = <math>17,5 * 0,4 PF = 7 PF</math></p>

## 5.2 INTEGRACJA Z SYSTEMEM BPMS

Jeżeli oprogramowanie realizowane jest z użyciem technologii BPMS, oprogramowanie BPMS (silnik procesów biznesowych) traktujemy jako odrębną Aplikację (nazywaną Aplikacją BPMS), dostarczającą funkcjonalności do obsługi przepływu prac, z którą integrują się pozostałe Aplikacje, nazywane dalej Aplikacjami Merytorycznymi. W takim przypadku zastosowanie mają poniższe reguły.

ID reguły	IFPUG014: Aplikacja BPMS
Opis reguły	<p>Po stronie Aplikacji BPMS, na każdy proces biznesowy obsługiwany przez Aplikację BPMS identyfikowany jest:</p> <ul style="list-style-type: none"> <li>• 1 ILF reprezentujący instancję procesu biznesowego,</li> <li>• 1 ILF reprezentujący konfigurację procesu biznesowego,</li> <li>• 1 EI, który umożliwia utworzenie nowej instancji procesu biznesowego przez Aplikację Merytoryczną,</li> <li>• 1 EI, który umożliwia aktualizację instancji procesu biznesowego, przez Aplikację Merytoryczną,</li> <li>• 1 EI, który umożliwia anulowanie (usunięcie) instancji procesu biznesowego, przez Aplikację Merytoryczną,</li> <li>• 1 EQ, który udostępnia Aplikacji Merytorycznej dane o stanie instancji procesu biznesowego.</li> </ul> <p>Po stronie Aplikacji Merytorycznej, na każdy Elementarny Proces zdefiniowany po stronie Aplikacji BPMS, reprezentującej usługę, z której korzysta Aplikacja Merytoryczna, definiowany jest pojedynczy EIF zgodnie z regułą IFPUG013.</p>
Przykład	<p><i>Aplikacja kancelaryjna obsługuje rejestrację, podgląd i wycofanie zgłoszeń zwierzęcych. Aplikacja kontrolingowa obsługuje wprowadzanie, podgląd danych, skierowanie do poprawy, anulowanie zgłoszenia, przetwarzanie zgłoszenia, potwierdzenie zgłoszenia. Workflow obsługi zgłoszeń zdefiniowany jest w Aplikacji BPMS.</i></p> <p><i>Po stronie Aplikacji BPMS identyfikuje się:</i></p> <ul style="list-style-type: none"> <li>• <i>ILF dla instancji procesu obsługi zgłoszenia,</i></li> <li>• <i>ILF dla konfiguracji procesu obsługi zgłoszenia,</i></li> <li>• <i>EI dla utworzenia instancji procesu obsługi zgłoszenia,</i></li> <li>• <i>EI dla aktualizacji instancji procesu obsługi zgłoszenia,</i></li> <li>• <i>EI dla usunięcia instancji procesu obsługi zgłoszenia,</i></li> </ul>

- *EQ pobierająca stan instancji procesu obsługi zgłoszenia.*

*Po stronie Aplikacji kancelaryjnej identyfikuje się:*

- *EI rejestracji zgłoszenia,*
- *EI wycofania zgłoszenia,*
- *EQ podglądu zgłoszenia,*
- *EIF dla utworzenia instancji procesu obsługi zgłoszenia, bo korzysta z niego rejestracja zgłoszenia,*
- *EIF dla usunięcia instancji procesu obsługi zgłoszenia, bo korzysta z niego wycofanie zgłoszenia,*
- *EIF pobierająca stan instancji procesu obsługi zgłoszenia, bo korzysta z niego podgląd zgłoszenia.*

*Po stronie Aplikacji kontrolingowej identyfikuje się:*

- *EI wprowadzania danych zgłoszenia,*
- *EQ podglądu danych zgłoszenia,*
- *EI skierowania do poprawy zgłoszenia,*
- *EI anulowania zgłoszenia,*
- *EO przetwarzania zgłoszenia,*
- *EI potwierdzenie zgłoszenia,*
- *EIF dla aktualizacji instancji procesu obsługi zgłoszenia, bo korzysta z niego wprowadzanie, skierowanie do poprawy, przetwarzanie, potwierdzenie zgłoszenia,*
- *EIF dla usunięcia instancji procesu obsługi zgłoszenia, bo korzysta z niego anulowanie zgłoszenia,*
- *EIF pobierająca stan instancji procesu obsługi zgłoszenia, bo korzysta z niego podgląd danych zgłoszenia.*

## 6 WYMAGANIA TECHNOLOGICZNE

Wymagania, które polegają na wprowadzeniu lub zmianie ograniczeń dla danej funkcjonalności i wiążą się z ulepszeniem lub przebudową sposobu realizacji (refaktoringiem) istniejącej funkcjonalności, poprzez zastosowanie określonego rozwiązania technicznego, wzorca projektowego, specyfikacji technicznej lub zasad projektowych, będą nazywane dalej Wymaganiami Technologicznymi.

Rozwiązania, praktyki lub standardy, których wprowadzenie jest przedmiotem Wymagania Technologicznego muszą należeć do jednej ze zdefiniowanych niżej kategorii, nazywanych dalej Warstwami Technologicznymi:

- środowisko programistyczne,
- architektura logiczna, wzorce, wytyczne,
- UX i interfejs użytkownika.

„Środowisko programistyczne” obejmuje zastosowanie określonego oprogramowania standardowego pełniącego funkcję środowiska wykonawczego w rozumieniu pojęcia ExecutionEnvironment ze specyfikacji UML 2.X czy oprogramowania typu middleware np. serwera bazy danych, serwera aplikacyjnego, silnika bazy danych, zastosowanie określonych frameworków lub bibliotek programistycznych, API, gotowych komponentów oprogramowania, oprogramowania firm trzecich.

„Architektura logiczna, wzorce, wytyczne” obejmuje stosowanie określonych wzorców projektowych lub architektonicznych, algorytmów, standardów i technologii w rozumieniu specyfikacji lecz nie gotowych rozwiązań, wytycznych dotyczących projektowania rozwiązań, dekompozycji, organizacji kodu źródłowego (np. wydzielenia określonej warstwy), zależności pomiędzy komponentami, itp. Przykładami Standardów Technologicznych należących do tej warstwy są: stosowanie architektury opartej o mikrousługi, Blockchain, SSO (single sign-on), SPA (single-page application), itp.

„UX i interfejs użytkownika” dotyczy zmian związanych z interfejsem użytkownika, wygodą użytkownika, wyglądem elementów, sposobem interakcji, np.: stosowanie określonej kolorystyki interfejsu użytkownika, stosowanie paska postępu przy przesyłaniu dużych ilości danych, miejsce wyświetlania komunikatów, znikanie komunikatów, możliwość wykonania czynności „przeciągnij i upuść” na określonych elementach interfejsu użytkownika.

Koszt realizacji Wymagań Technologicznych jest określany na podstawie wzoru:

$$RF_T = \sum_{i=1}^W RF_i \cdot w_i$$

gdzie:

$RF_T$  – rozmiar Wymagań Technologicznych,

$W$  – liczba Warstw Technologicznych,

$RFi$  – rozmiar Elementarnych Procesów reprezentujących funkcjonalność, którą ulepszono lub przebudowano w zakresie  $i$ -tej warstwy technologicznej, zidentyfikowanej zgodnie z niniejszym Podręcznikiem,

$w_i$  – współczynnik wpływu  $i$ -tej warstwy technologicznej, określony w Tabeli 7. Współczynnik wpływu warstwy technologicznej.

**Tabela 7. Współczynnik wpływu warstwy technologicznej.**

Warstwa Technologiczna	Współczynnik wpływu ( $w_i$ )
środowisko programistyczne	0,3
architektura logiczna, wzorce, wytyczne	0,3
UX i interfejs użytkownika	0,3

### Przykład 6.A:

Zdefiniowano następujące Wymagania Technologiczne:

1. Zmiana interakcji obsługi formularzy wprowadzania wniosków. Rozmiar Elementarnych Procesów związanych z wprowadzaniem wniosków wynosi 100 PF.
2. Zmiana obsługi paska postępu w funkcjonalności obejmującej import plików xml. Rozmiar Elementarnych Procesów obejmujących import plików xml wynosi 80 PF, przy czym 40 PF spośród tej funkcjonalności związana jest z obsługą formularzy wprowadzania wniosków.
3. Zastosowanie biblioteki React.JS do funkcjonalności związanej z wprowadzaniem wniosków (tych samych co w pkt.1).

Wymagania 1 i 2 odnoszą się do tej samej Warstwy Technologicznej (UX i interfejs użytkownika) i dotyczą sumarycznie funkcjonalności o rozmiarze 140 PF. Funkcjonalność o rozmiarze 40 PF jednocześnie jest funkcjonalnością wprowadzania wniosków jak i funkcjonalnością obejmującą import plików, a zatem dotyczą jej zastosowanie rozwiązań zdefiniowanych zarówno w wymaganiu 1 jak i 2, jednak nie wpływa to na rozmiar wymagania, do wyznaczenia rozmiaru wystarczy określenie unikatowej funkcjonalności, której zmiany odnoszą się do danej warstwy.

Wymaganie 3 odnosi się do innej warstwy (środowisko programistyczne) i dotyczy funkcjonalności wprowadzania o rozmiarze 100 PF.

Dla warstwy (architektura logiczna, wzorce, wytyczne) nie zdefiniowano żadnych wymagań, Standardy Technologiczne w ramach tej warstwy nie będą miały być zastosowane w żadnej funkcjonalności (0 PF).

Rozmiar wymagań 1, 2 i 3 wynosi zatem  $140 * 0,3 + 100 * 0,3 = 42 + 30 = 72$  PF.

**Uwaga:** Realizację Wymagania Technologicznego należy rozumieć jako częściową, ponowną implementację istniejącej funkcjonalności, Wymagania Technologiczne nie odnoszą się do zastosowania określonych rozwiązań w nowej lub zmienianej funkcjonalności. Koszt związany z zastosowaniem określonych rozwiązań w nowej lub zmienianej funkcjonalności uznaje się za zawarty w koszcie wytworzenia lub zmiany tej funkcjonalności i co za tym idzie wliczony w cenę Punktu Funkcyjnego.

## 7 DOKUMENTOWANIE WYCENY



## 7.1 DEFINICJE

**21prim** – aplikacja służąca do wyznaczania rozmiaru funkcjonalnego oprogramowania na podstawie specyfikacji wymagań, dostępna pod adresem 21prim.com; aplikacja pozwala na dokumentowanie, weryfikację oraz uzgadnianie pomiaru wielkości oprogramowania w wybranej metodzie punktów funkcyjnych.

**Lokalizacja Zmiany** – Przestrzeń Robocza, w której powinna być umieszczana Specyfikacja oraz Pomiary dla danej Zmiany.

**Metoda ARiMR** – metoda wymiarowania oprogramowania wspierana przez 21prim zgodna z niniejszym Podręcznikiem.

**Pomiar (ang. Measurement)** – udokumentowany pomiar oprogramowania wykonany na podstawie określonej Specyfikacji przy użyciu aplikacji 21prim zgodnie z jedną z dostępnych w 21prim metod punktów funkcyjnych

**Pomiar Pierwotny** – Pomiar wykonany na podstawie pierwszej wersji Specyfikacji dla której w ogóle wykonano Pomiar.

**Pomiar Różnicowy** – Pomiar wykonany na podstawie różnic pomiędzy dwoma kolejnymi wersjami Specyfikacji, który modyfikuje Pomiar wykonany dla poprzedniej wersji Specyfikacji, w taki sposób, że staje się on Pomiarom dla nowej wersji Specyfikacji.

**Przestrzeń Robocza (ang. Workspace)** – zasób dostępny z poziomu 21prim, w którym umieszczane są Specyfikacje i odpowiadające im Pomiary; dostęp do określonej Przestrzeni Roboczej jest przyznawany tylko określonym Użytkownikom.

## 7.2 OGÓLNE ZASADY

Warunkiem akceptacji wyceny Zmiany jest udokumentowanie tej Zmiany przy pomocy Pomiaru wykonanego zgodnie z Metodą ARiMR.

Przekazane do akceptacji Pomiary Zmian nie mogą być ani zmieniane ani usuwane, będą dostępne w 21prim dla uprawnionych użytkowników przez cały okres trwania Umowy.

Podstawą dla Pomiaru powinna być Specyfikacja Zmiany, której dotyczy przedstawiona przez Wykonawcę wycena Zmiany wyznaczona na podstawie danego Pomiaru, w postaci dokumentu w formacie PDF umieszczonego w Lokalizacji Zmiany.

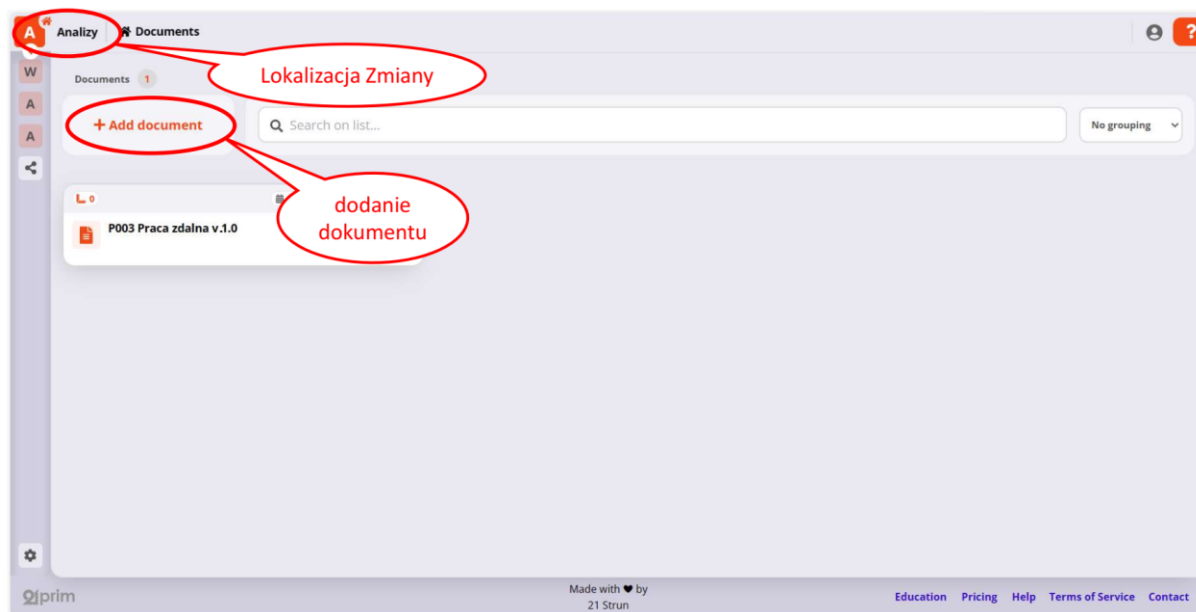
Lokalizacja Zmiany musi być uzgodniona pomiędzy Stronami przed przekazaniem Pomiaru do akceptacji.

Uprawnienia dostępu do Lokalizacji Zmiany nadaje ARiMR w ramach Umowy.

Wprowadzenie do użytkowania 21prim jest dostępne pod adresem <https://21prim.com/help-approx>.

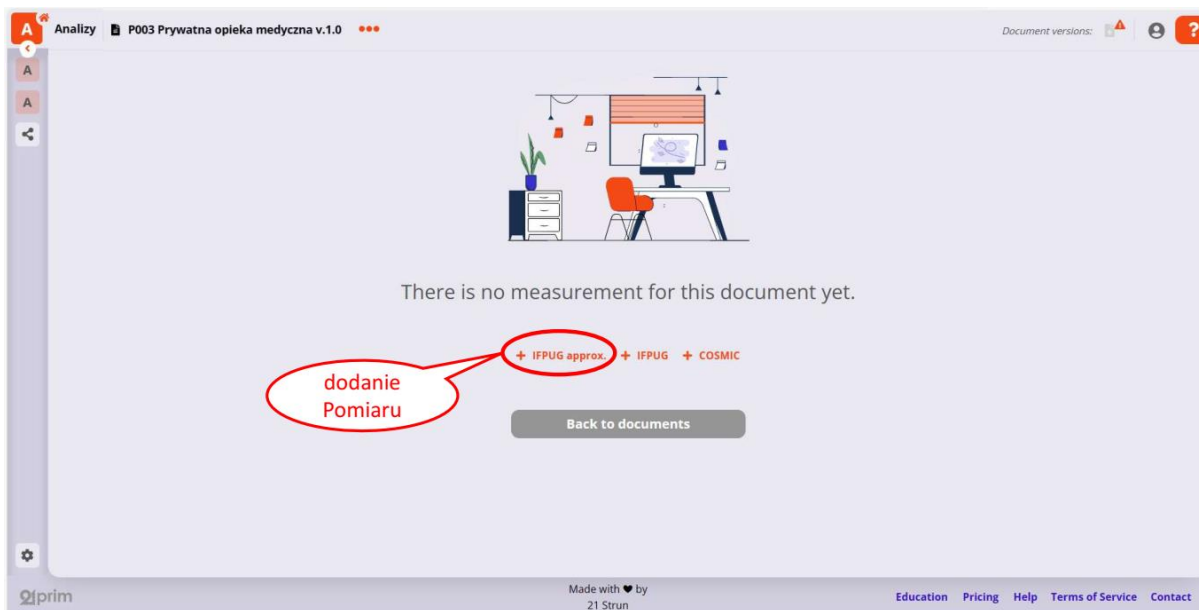
## 7.3 UTWORZENIE I PRZYGOTOWANIE POMIARU DLA ZMIANY

W celu utworzenia Pomiaru dla Zmiany należy na podstawie zatwierdzonej Specyfikacji Zmiany utworzyć plik w formacie PDF, a następnie umieścić go w Lokalizacji Zmiany (opcja *Add document* - Rysunek 3. Dodanie Specyfikacji w 21prim). Nazwa Specyfikacji nadawana w 21prim powinna zawierać identyfikator, nazwę oraz wersję Specyfikacji Zmiany np. *P003 Obsługa wniosku urlopowego v.1.0.*



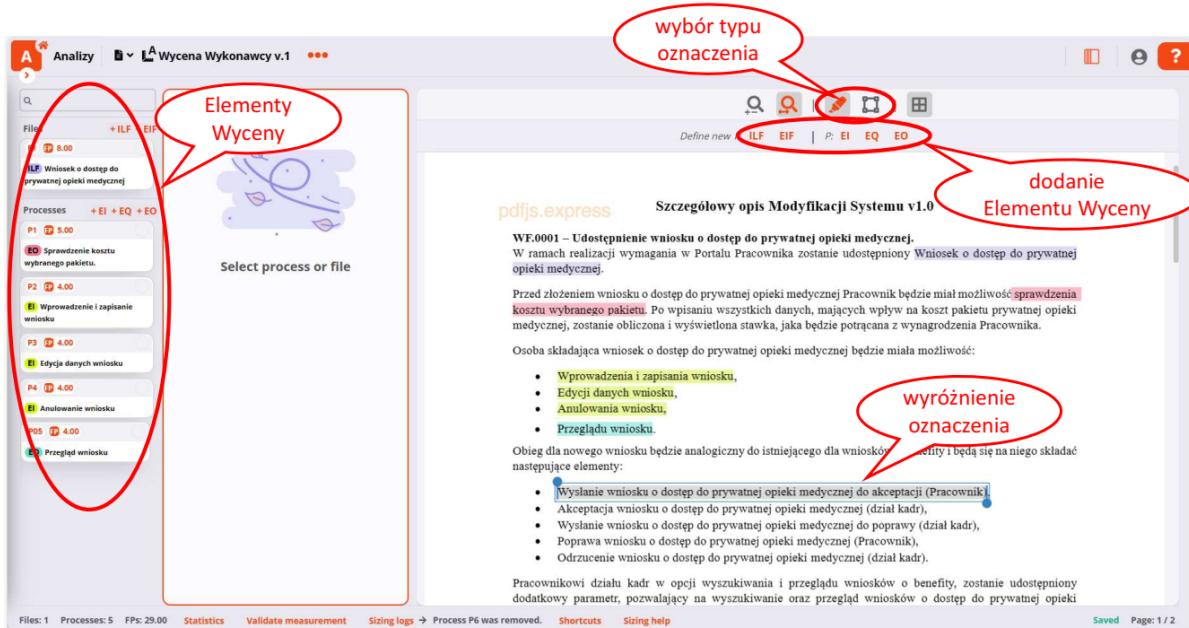
**Rysunek 3. Dodanie Specyfikacji w 21prim**

Dla Specyfikacji Zmiany należy utworzyć Pomiar zgodny z Metodą ARiMR (opcja *IFPUG approx.* - Rysunek 4. Dodanie Pomiaru dla Specyfikacji w 21prim) o nazwie *Wycena Wykonawcy v.[n]*, gdzie *[n]* oznacza kolejny numer wersji Pomiaru dla danej wersji Specyfikacji (patrz rozdział 7.6)



#### Rysunek 4. Dodanie Pomiaru dla Specyfikacji w 21prim

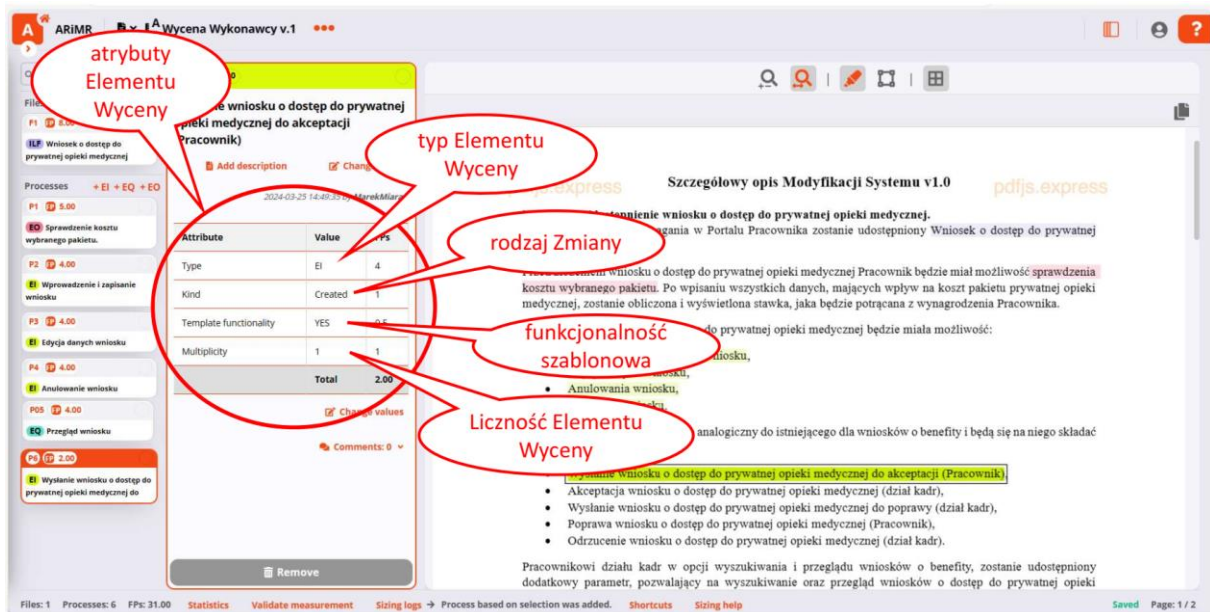
Po utworzeniu Pomiaru ARiMR aplikacja wyświetli okno, w którym widoczna będzie Specyfikacja. W Specyfikacji należy odpowiednio oznaczyć fragmenty opisu wymagań poświadczające zasadność występowania Elementów Wyceny przy pomocy funkcjonalności 21prim (Rysunek 5. Przygotowanie Pomiaru poprzez oznaczanie odpowiednich fragmentów w Specyfikacji). Wszystkie Elementy Wyceny będące podstawą Rozmiaru Funkcjonalnego Zmiany muszą być oznaczone w Specyfikacji.



#### Rysunek 5. Przygotowanie Pomiaru poprzez oznaczanie odpowiednich fragmentów w Specyfikacji

Dla każdego Elementu Wyceny należy określić (Rysunek 6. Atrybuty Elementu Wyceny):

- typ Elementarnego Procesu lub Pliku Danych, który reprezentuje,
- czy reprezentuje dodanie, modyfikację, czy usunięcie Elementarnego Procesu lub Pliku Danych,
- czy reprezentuje funkcjonalność szablonową (reguła IFPUG015),
- licznosc Elementarnego Procesu lub Pliku Danych - liczbę wystąpień odrębnych, jednak charakteryzujących się takimi samymi parametrami (z punktu widzenia IFPUG approximation) Elementów Wyceny, pozwalająca zastąpić dodawanie każdego Elementu Wyceny z osobna.



## Rysunek 6. Atrybuty Elementu Wyceny

Atrybuty Elementów Wyceny muszą być określone spójnie z opisem Wymagania, który poświadcza jego występowanie.

Przykładowy demonstracyjny Pomiar Zmiany jest dostępny pod adresem

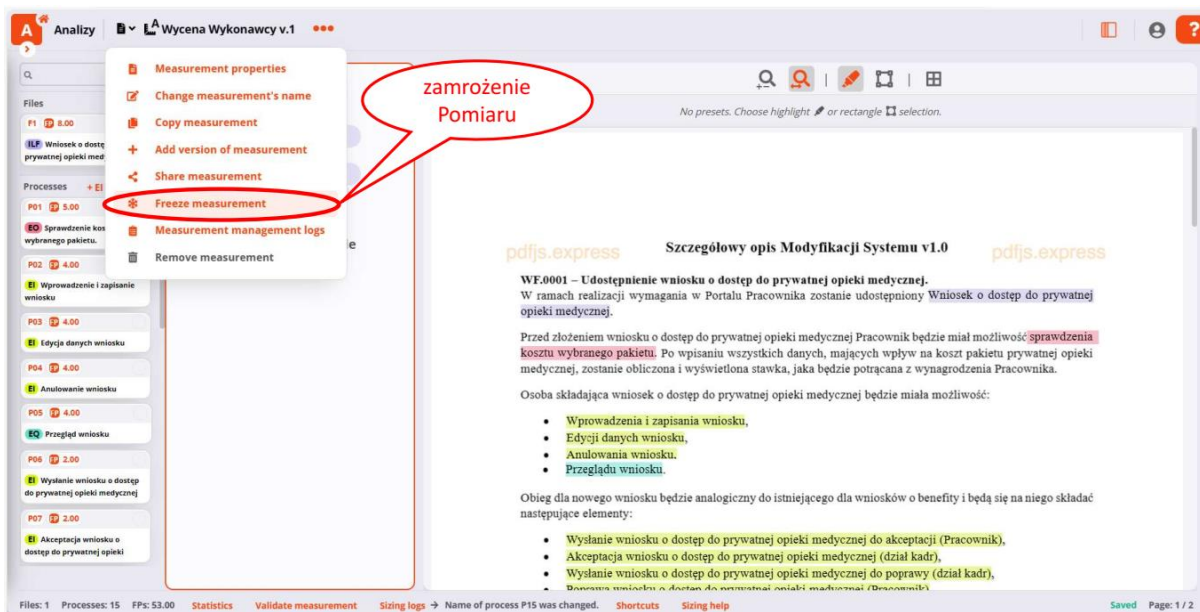
<https://21prim.com/link/demo-measurement>

Prace nad przygotowaniem Pomiaru mogą być prowadzone przez kilku użytkowników, jednak w tym samym czasie Pomiar może edytować tylko jeden użytkownik.

## 7.4 PRZEKAZANIE POMIARU DO AKCEPTACJI

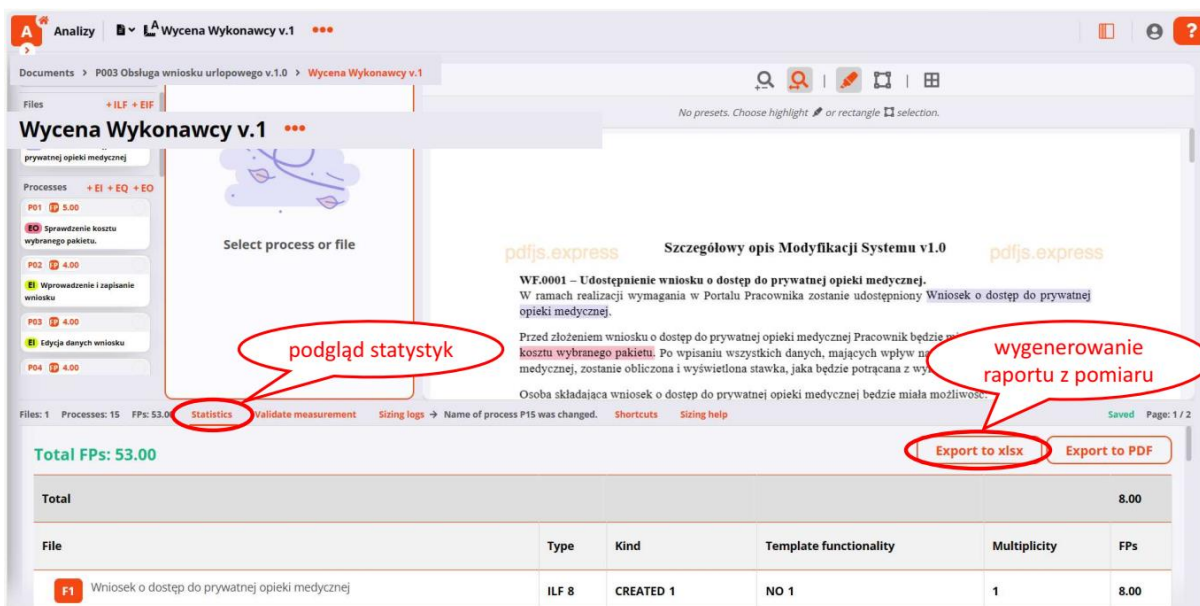
Przekazanie Pomiaru do akceptacji ARiMR wymaga:

- zamrożenia Pomiaru (opcja *Freeze Measurement* - Rysunek 7. Zamrożenie pomiaru),



Rysunek 7. Zamrożenie pomiaru

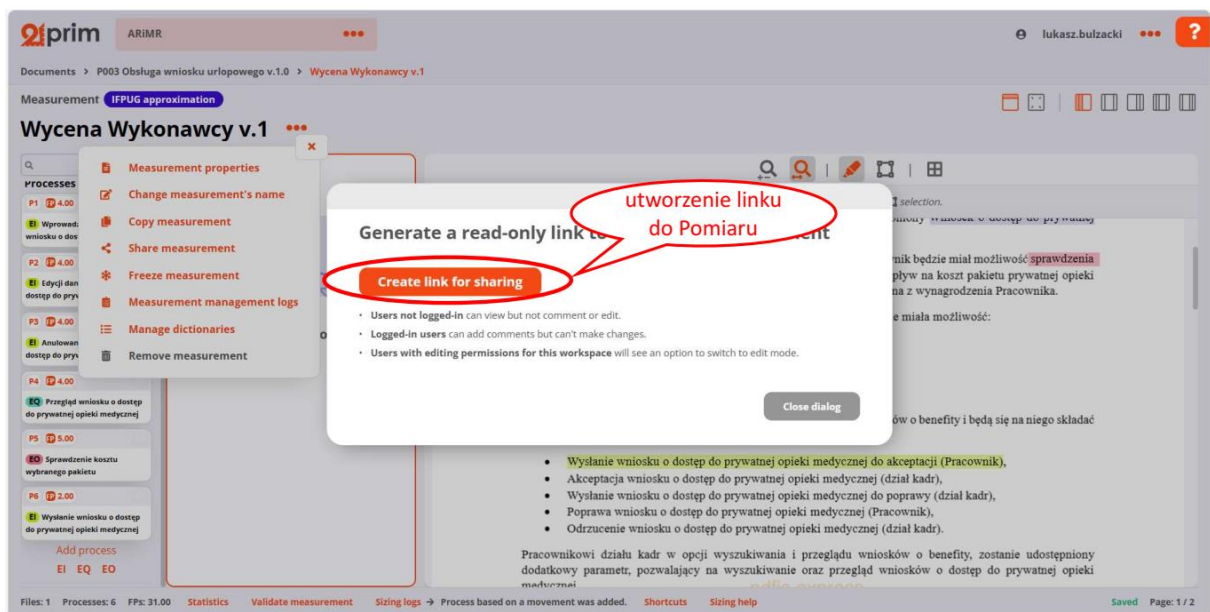
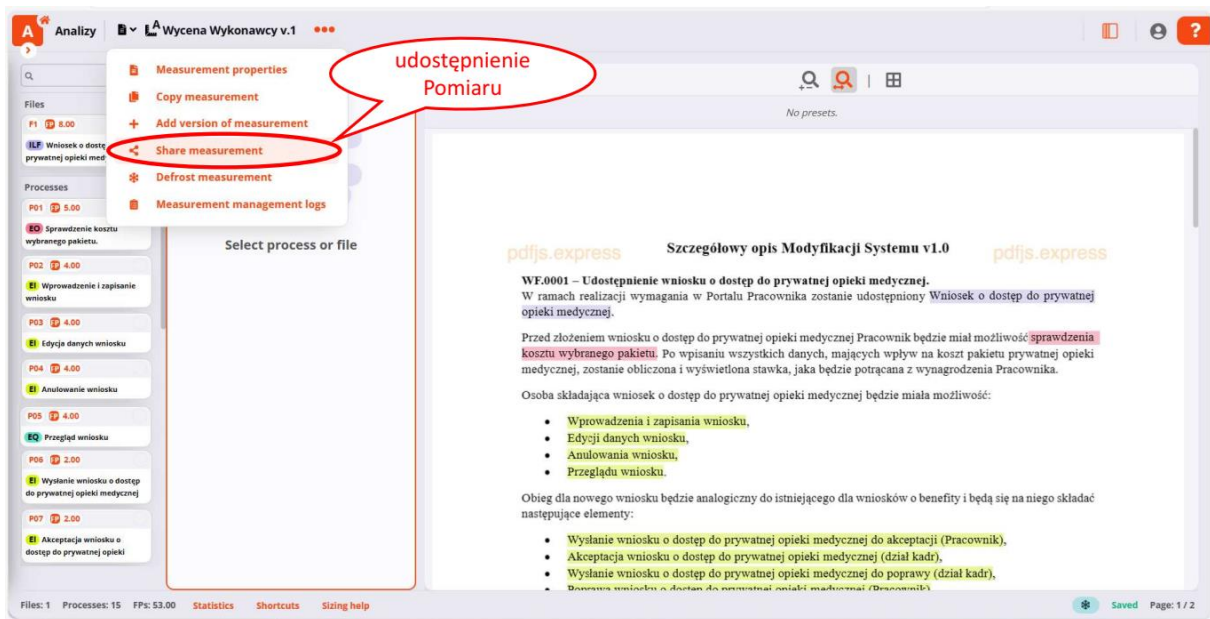
- wygenerowania końcowego raportu z Pomiaru, w postaci pliku xlsx (opcja *Export to xlsx* - Rysunek 8. Wygenerowanie końcowego raportu z Pomiaru),

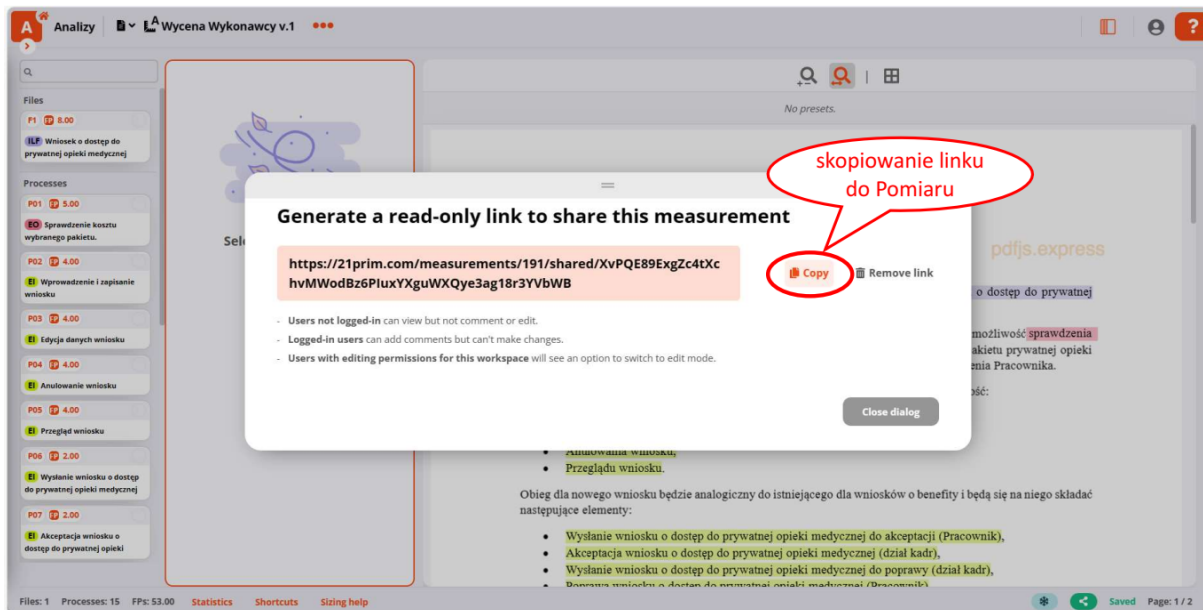


Rysunek 8. Wygenerowanie końcowego raportu z Pomiaru

- pobrania linku do Pomiaru (opcja: *Share/Copy Link* - Rysunek 9. Pobranie linku do Pomiaru).







## Rysunek 9. Pobranie linku do Pomiaru

Raport z Pomiaru oraz link do Pomiaru przekazywany jest do ARiMR zgodnie z procedurą przekazania wyceny Zmiany określoną w Umowie.

Zamrożenie Pomiaru powoduje, że Pomiar nie może być od tej pory zmieniany, ale jest dostępny do odczytu i może być opiniowany.

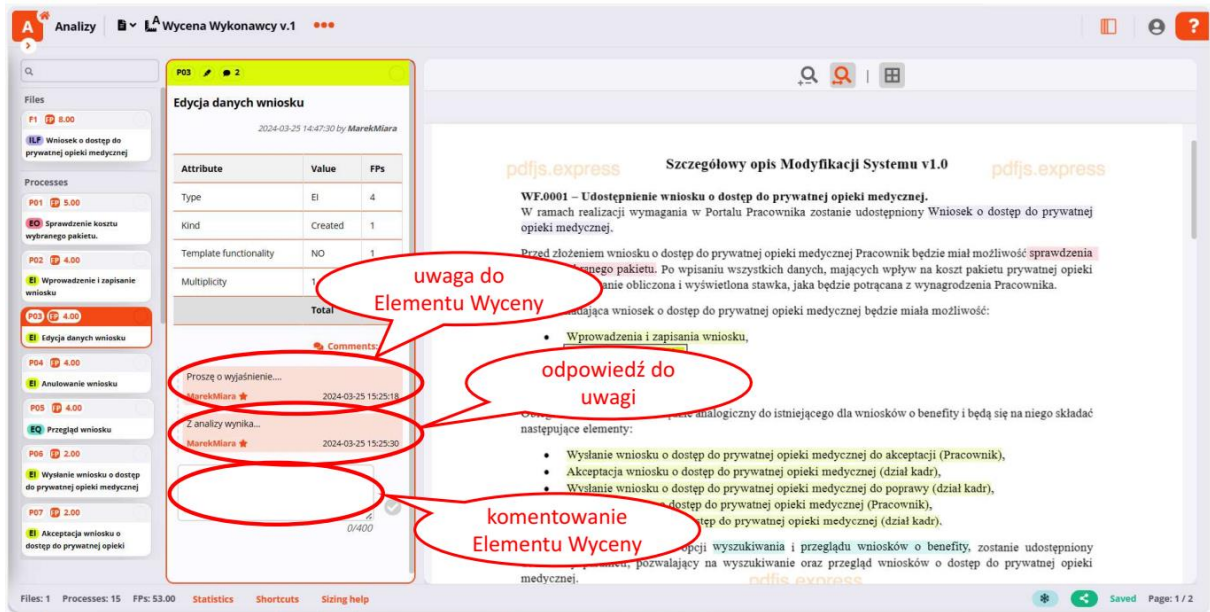
## 7.5 OPINIOWANIE I UZGADNIANIE WYCENY

Opiniowanie wyceny, w zakresie konkretnych Elementów Wyceny, której podstawą jest przekazany Pomiar, będzie się odbywać z użyciem 21prim.

Niezależnie od zasad opiniowania wyceny Zmiany określonych w Umowie, uwagi do konkretnych Elementów Wyceny powinny być umieszczane, a ich rozwiązania uzgadniane, bezpośrednio w danym Pomiarze.

Uwaga do Elementu Wyceny powinna mieć postać komentarza do Elementu Wyceny (Rysunek 10. Opiniowanie wyceny - komentowanie Elementów Wyceny).

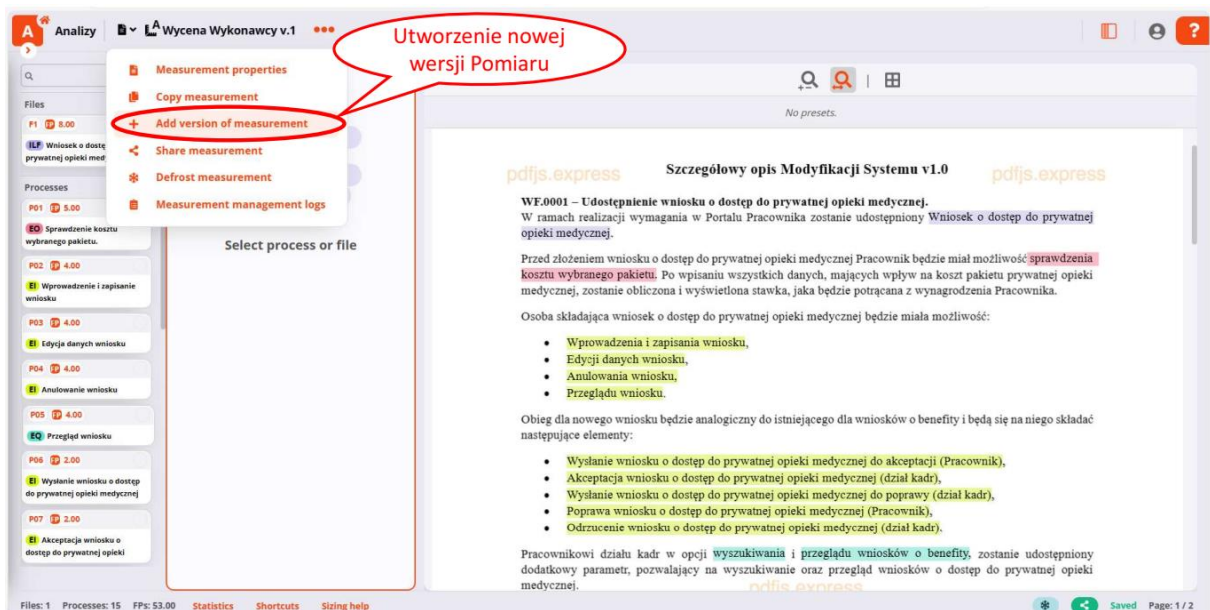
Uzgardnianie rozwiązania uwagi powinno mieć postać odpowiedzi do w/w komentarza w kolejnych komentarzach do danego Elementu Wyceny.



Rysunek 10. Opiniowanie wyceny - komentowanie Elementów Wyceny

## 7.6 Utworzenie nowej wersji pomiaru dla zmiany

W przypadku konieczności modyfikacji Pomiaru (np. w celu uwzględnienia uwag) przekazanego do akceptacji (zamrożonego) powinna zostać utworzona nowa wersja Pomiaru (opcja: New Version of Measurement). Nazwa nowej wersji Pomiaru powinna być nadawana zgodnie z zasadami opisanymi w rozdziale 7.3 np. *Wycena Wykonawcy v.2*



Rysunek 11 Utworzenie nowej wersji Pomiaru

Kolejna wersja Pomiaru początkowo jest kopią ostatniej wersji Pomiaru przekazanej do akceptacji, która może być modyfikowana do momentu ponownego jej przekazania do akceptacji.



W przypadku kolejnej wersji Pomiaru wygenerowanie Raportu końcowego z Pomiaru umożliwia opcja *Statistics/Summary*. Poza tym, zasady przekazywania i opiniowania nowej wersji Pomiaru są takie same jak opisano to w rozdziale 7.4

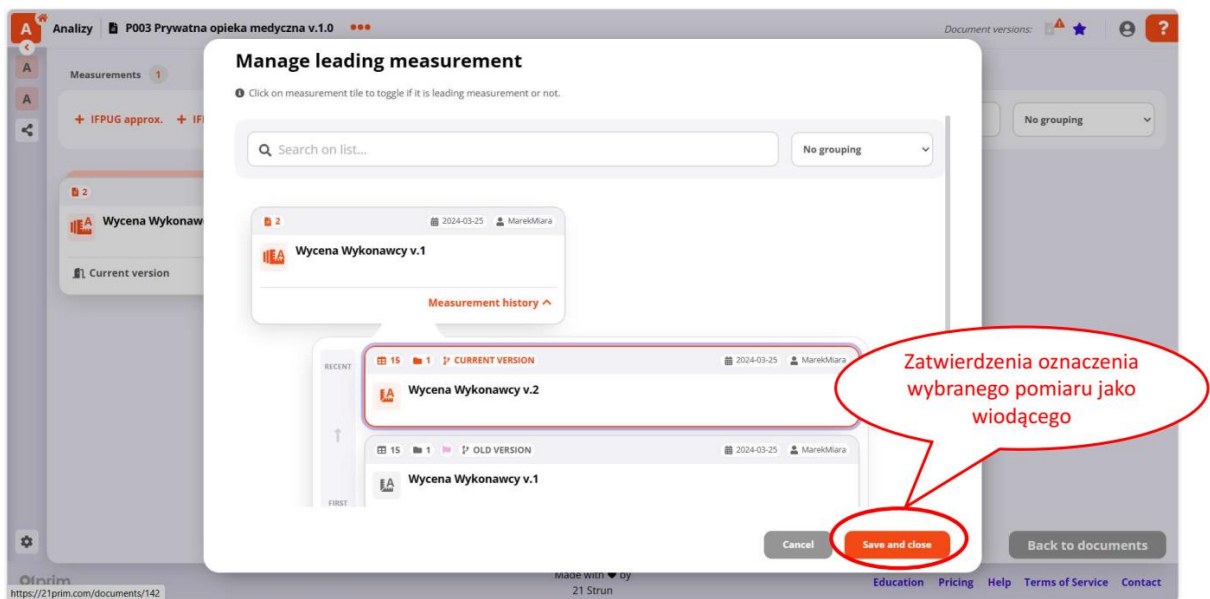
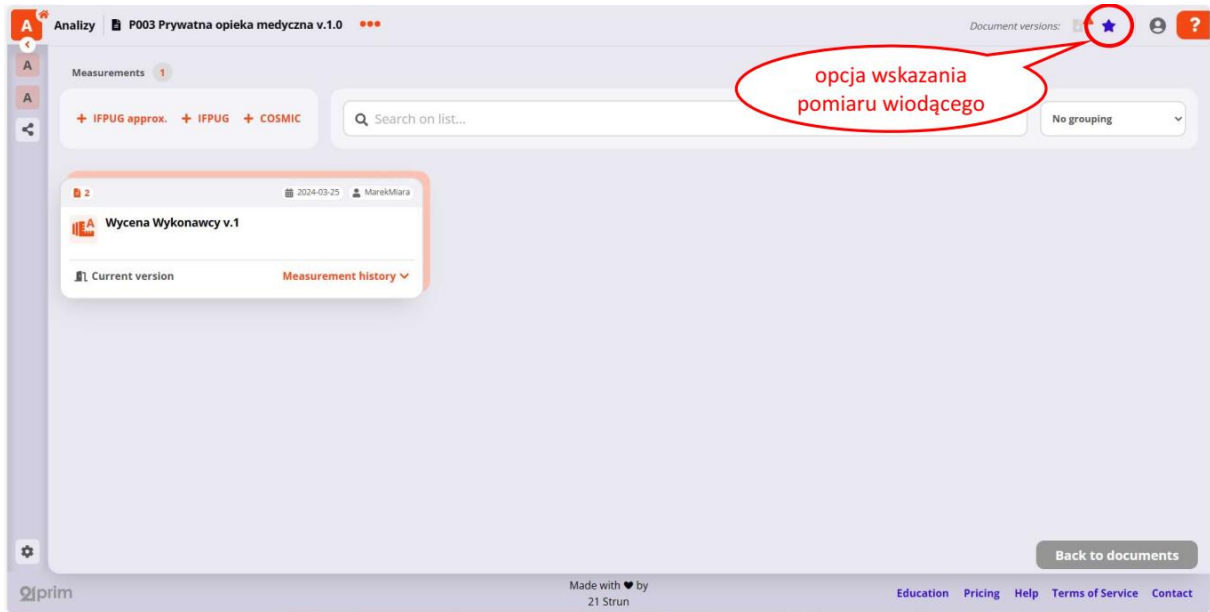
Aplikacja 21prim umożliwia także wyświetlenie porównania kolejnych wersji Pomiaru (opcja *Measurement versions*).

## 7.7 UTWORZENIE POMIARU RÓŻNICOWEGO

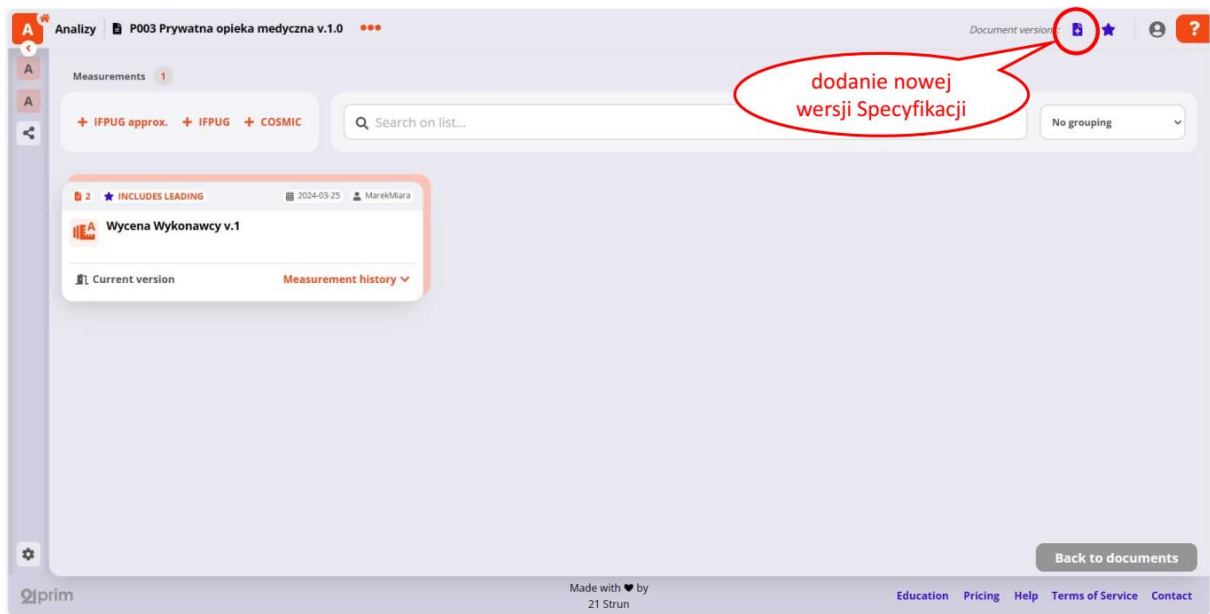
Pomiar Różnicowy umożliwia nam wprowadzenie korekty Pomiaru tylko na podstawie zmian w kolejnej wersji Specyfikacji. Funkcjonalność Pomiaru Różnicowego pozwala uniknąć przeprowadzenia ponownie pełnego Pomiaru w przypadku zmiany treści Specyfikacji.

W przypadku zmiany treści Specyfikacji (np. w wyniku uzgodnień) należy dodać nową wersję Specyfikacji, w której będą wyróżnione różnice treści względem wersji Specyfikacji, dla której utworzono uzgodniony Pomiar. Zmiany treści wprowadzane wcześniej (przed utworzeniem wersji Specyfikacji, dla której utworzono Pomiar i przekazano go do akceptacji) nie powinny być wyróżnione. Dokument nowej wersji Specyfikacji zgodny z powyższym opisem można uzyskać tworząc dokument PDF na podstawie dokumentu Word, w którym są widoczne zmiany treści w trybie rejestracji zmian. Wyżej opisana wersja Specyfikacji posłuży jako podstawa do utworzenia Pomiaru Różnicowego.

W celu dodania nowej wersji Specyfikacji, w pierwszej kolejności należy oznaczyć uzgodniony Pomiar ostatniej wersji Specyfikacji jako wiodący (przez co będzie on modyfikowany przez dany Pomiar Różnicowy) – opcja: *Choose leading measurement* (Rysunek 12. Oznaczenie Pomiaru poprzedniej wersji Specyfikacji jako wiodącego). Do Pomiaru oznaczonego jako wiodący należy dodać nową wersję Specyfikacji (opcja: *Add document version* - Rysunek 13. Dodanie nowej wersji Specyfikacji). Nazwa nowej wersji Specyfikacji powinna różnić się od innych wersji tej samej Specyfikacji wyłącznie numerem wersji Specyfikacji np. *P003 Obsługa wniosku urlopowego v.1.1*.



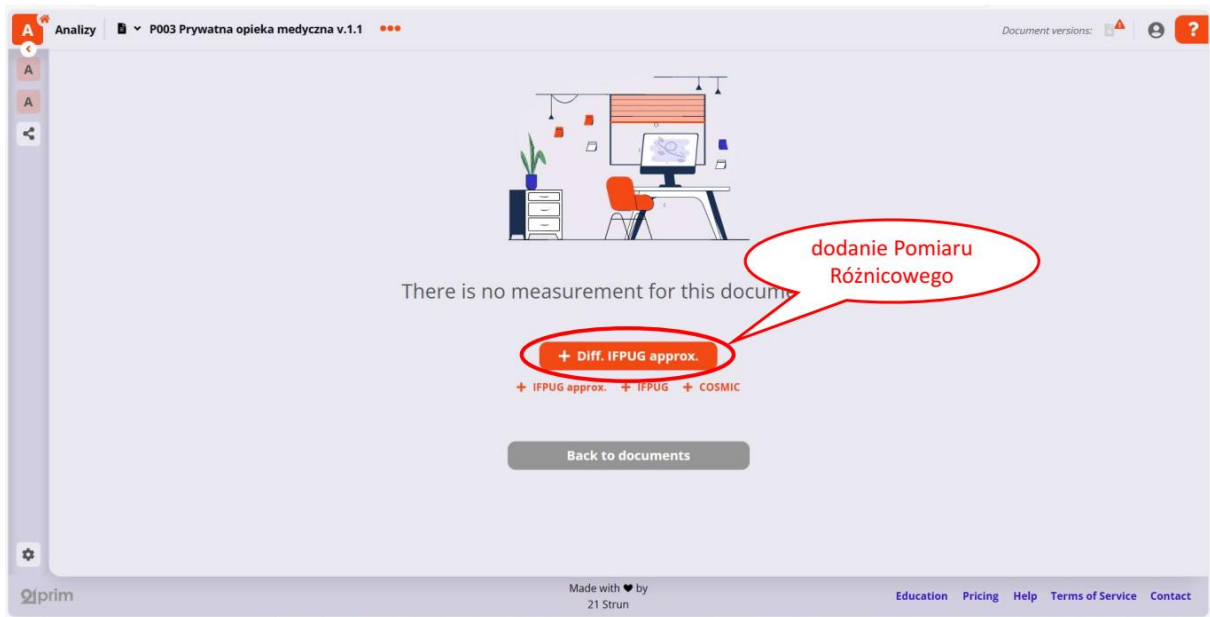
Rysunek 12. Oznaczenie Pomiaru poprzedniej wersji Specyfikacji jako wiodącego



### Rysunek 13. Dodanie nowej wersji Specyfikacji

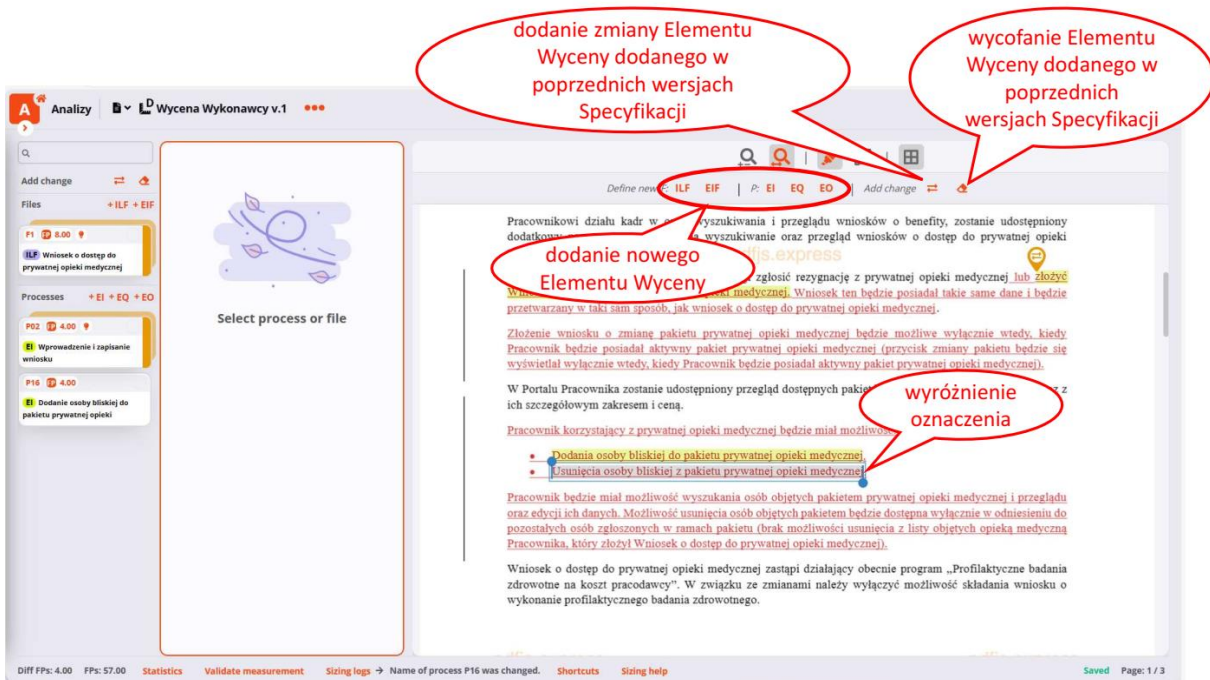
Dodanie nowej wersji Specyfikacji powoduje zablokowanie możliwości wprowadzania zmian w pomiarze wiodącym, dlatego nową wersję Specyfikacji należy dodawać dopiero wtedy, kiedy Pomiar dla ostatniej wymiarowanej wersji Specyfikacji został już uzgodniony i zatwierdzony i nie będzie podlegał dalszym korektom.

Po dodaniu nowej wersji Specyfikacji należy utworzyć dla niej Pomiar Różnicowy (opcja: *Diff. IFPUG approx.* – Rysunek 14. Dodanie Pomiaru Różnicowego dla nowej wersji Specyfikacji w 21prim). Zasady nazywania i wersjonowania Pomiarów Różnicowych dla danej wersji Specyfikacji są takie same jak dla innych Pomiarów i są opisane w rozdziale 7.3.



**Rysunek 14. Dodanie Pomiaru Różnicowego dla nowej wersji Specyfikacji w 21prim**

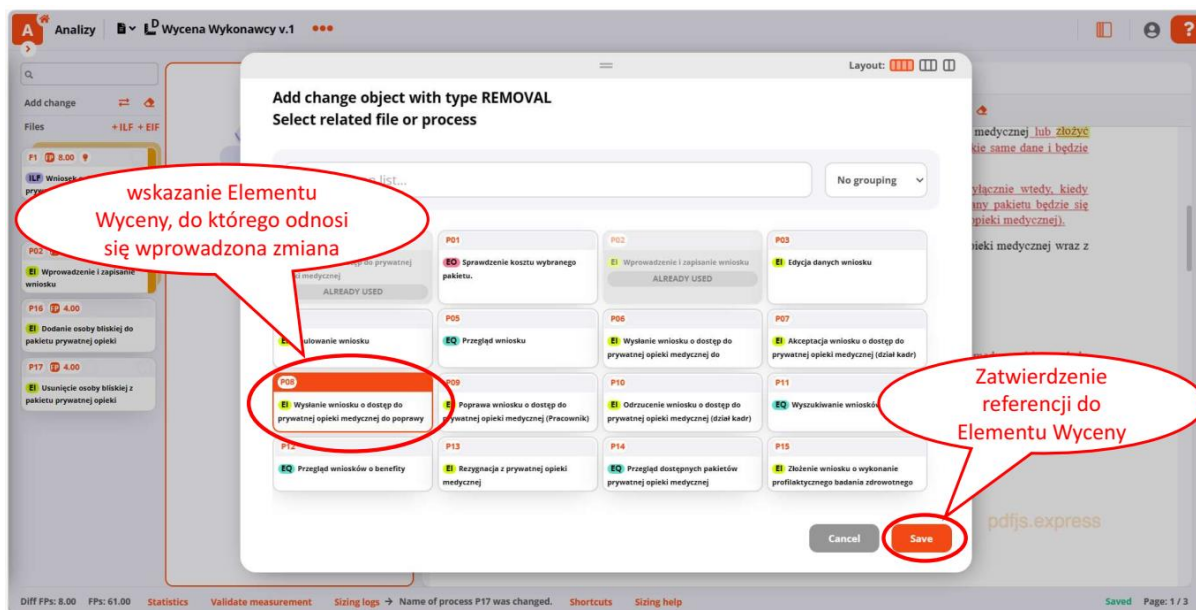
W nowej, dodanej zgodnie z powyższym opisem, wersji Specyfikacji należy odpowiednio oznaczyć fragmenty opisu wymagań poświadczające zasadność występowania Elementów Wyceny, jednak tylko w części wyróżnionej opisu, który będzie różnicą pomiędzy nową, a ostatnią wersją Specyfikacji (Rysunek 15. Przygotowanie Pomiaru Różnicowego poprzez oznaczanie odpowiednich fragmentów w Specyfikacji przy pomocy 21prim).



**Rysunek 15. Przygotowanie Pomiaru Różnicowego poprzez oznaczanie odpowiednich fragmentów w Specyfikacji przy pomocy 21prim**

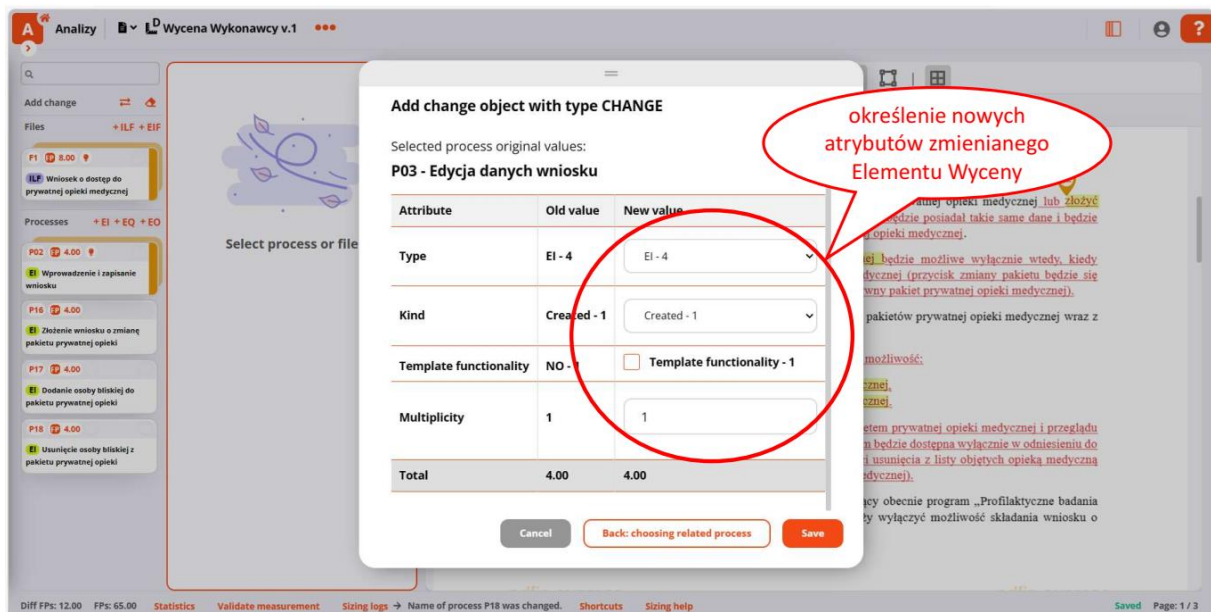
W nowej wersji Specyfikacji część opisu wymagań może zostać wyróżniona jako usunięta. Jeżeli usunięty opis wymagań zawierał fragmenty, które były poświadczeniami Elementów Wyceny we wcześniejszej wersji Pomiaru, to należy w nim oznaczyć fragmenty, które będą poświadczeniami wycofania tych Elementów Wyceny w nowej wersji Pomiaru.

W przypadku oznaczenia wycofania Elementu Wyceny należy wskazać konkretny wycofywany Element Wyceny utworzony we wcześniejszych wersjach Pomiaru.



**Rysunek 16. Wskazanie wycofanego Elementu Wyceny utworzonego we wcześniejszej wersji Pomiaru**

W nowej wersji Specyfikacji część opisu wymagań może dotyczyć zmian Elementów Wyceny zidentyfikowanych w poprzedniej wersji. Te elementy również należy oznaczyć w treści Specyfikacji wybierając właściwy atrybut Elementu Wyceny (opcja: *Add change*) oraz wskazać referencję do Elementu Wyceny dodanego w poprzedniej wersji Pomiaru – analogicznie jak w przypadku referencji do wycofanego Elementu Wyceny. Dodając referencję do zmiany Elementu Wyceny można dokonać edycji przypisanych mu we wcześniejszych wersjach Pomiarów atrybutów (Rysunek 17. Zdefiniowanie na nowo atrybutów zmienianego Elementu Wyceny).



**Rysunek 17. Zdefiniowanie na nowo atrybutów zmienianego Elementu Wyceny**

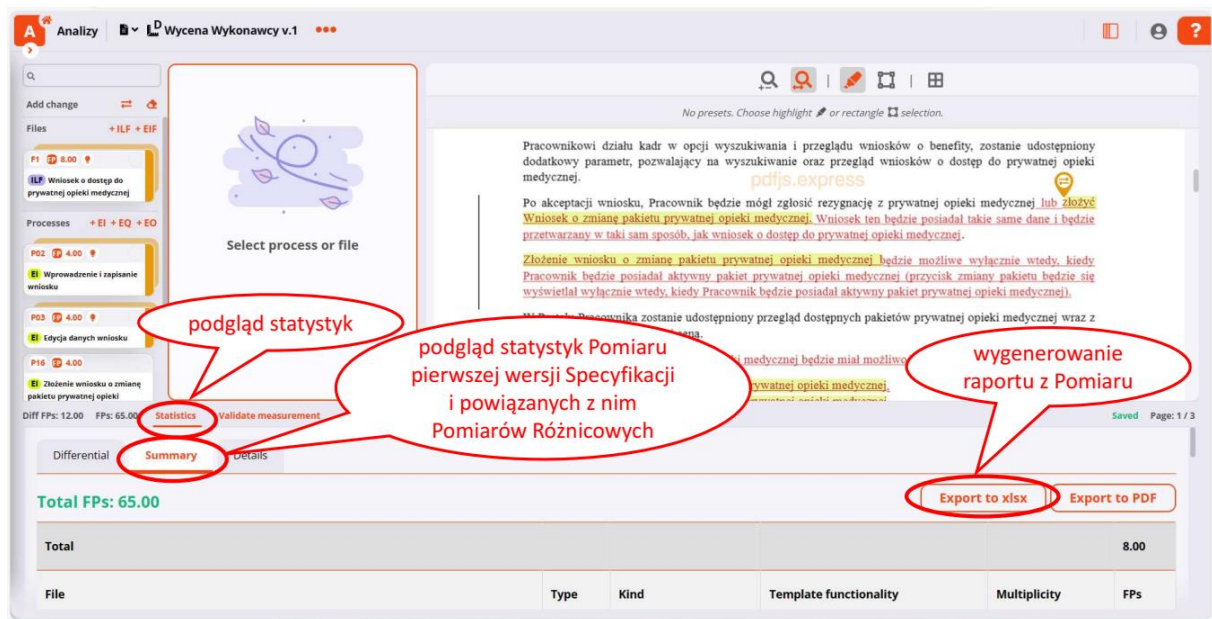
Przykłady kolejnych Pomiarów Różnicowych w aplikacji 21prim są dostępne pod **adresami:**

<https://21prim.com/link/demo-diff-1> <https://21prim.com/link/demo-diff-2>

Przekazywanie do akceptacji, opiniowanie i tworzenie nowej wersji Pomiaru Różnicowego odbywa się tak samo jak w przypadku zwykłego Pomiaru.

Aplikacja 21prim umożliwia wygenerowanie Raportu końcowego z Pomiaru (opcja *Statistics/Summary*), który jest złożeniem ostatniej wersji Pomiaru wykonanego dla pierwszej wersji Specyfikacji i ostatnich wersji wszystkich powiązanych z nim Pomiarów Różnicowych wykonanych dla kolejnych wersji Specyfikacji aż do wersji bieżącej Specyfikacji (Rysunek 18. Generowanie końcowego Raportu dla Pomiaru Różnicowego w 21prim).





Rysunek 18. Generowanie końcowego Raportu dla Pomiaru Różnicowego w 21prim

## 8 ODPSTĘPSTWA OD ZASAD DOKUMENTOWANIA WYCENY

Dopuszcza się stosowanie zmienionych w stosunku do wyżej opisanych, zasad dokumentowania, jeżeli spełnione będą następujące warunki:

- zmienione zasady zostaną szczegółowo opisane,
- zmienione zasady zostaną zaakceptowane przez ARiMR,
- zmienione zasady będą konsekwentnie stosowane w kolejnych Zmianach,
- sposób dokumentowania wyceny przygotowany zgodnie ze zmienionymi zasadami, będzie zawierał co najmniej taką informację jak sposób dokumentowania wyceny zgodny z opisem powyżej, nie nastąpi żadna strata informacji,
- dostępność i łatwość uzyskania każdego rodzaju informacji z dokumentacji wyceny przygotowanej zgodnie ze zmienionymi zasadami, będzie taka sama lub lepsza niż dostępność i łatwość uzyskania tego rodzaju informacji z dokumentacji wyceny zgodnej z opisem powyżej,
- jeżeli przed zmianą zasad była dokumentowana wycena, to dokumentacja ta zostanie przez Wykonawcę bezstratnie przemigrowana do dokumentacji wyceny zgodnej ze zmienionymi zasadami.