## 1. Architektura obecna

* **APEX na Oracle 19c**
  + APEX przechowuje wyniki uzgodnień danych, walidacje, zapytania SQL.
  + 22 tabów które posiadają rózne funkcjonalności oraz w 6 przypadkach (upload danych). Dane pochodzą od Veenalasa Natarajan.
  + 46 tabów (20 tabów omówilismy z Priya i znamy wstępną logikę) które posiadają rózne funkcjonalności oraz w 6 przypadkach (upload danych). Dane pochodzą od Mojego zepołu.
  + Brak repozytorium kodu, brak CI/CD pipeline.
  + Wbudowane mechanizmy wysyłki e-mail.
  + Brak testów jednostkowych, testów funkcjonalnych, testów wydajności.

## 2. Architektura docelowa

* **Brak APEX** – cała logika zostaje przeniesiona do:
  + **Nowej bazy danych** (Oracle 19c na nowym serwerze ).
  + **API Spring Boot** – nowe endpointy REST (dla każdego taba z APEX).
  + **GraphQL API** – nowe endpointy które bazuja na pierwszej warstwie punktów końcowych Java Spring Boot – niebędne do implmentacji w Service Bench.
  + **Custom UI** – dodanie brakujących wszytakich tabów w liczbie minimum 22 (46 w przypadku potwierdzenia z Priya) wraz z fitrami, sortowaniem oraz w 6 przypakach dodaniem drop down list oraz mozliwości aktualizacji rekordów w tabelach (dodatkowe słowniki).
* **Nowe elementy obowiązkowe:**
  + Nowe Repozytorium kodu dla obiektów PL/SQL.
  + Nowe Repozytorium kodu dla obiektów API Java Spring Boot.
  + Nowe Repozytorium kodu dla testów integracji API Java Spring Boot.
  + Nowe Repozytorium kodu dla obiektów API GraphQL.
  + Nowe Pipeliny dla bazy danych i API (Java spring Boot oraz GraphQL).
  + Dodatkowe role (uprawniania) do uwtorzenia w SC-IDP
  + Testy jednostkowe i integracyjne dla każdej nowej funkcjonalności która zostanie utworzona w Java spring Boot oraz GraphQL oraz bazie danych
  + Utworzenie osobnych funkcjonlaności w aplikacji ETL dla dodawnia danych do bazy danych które aktualziauja sześć tabel zródłowych.
  + Dodatkowe testy w aplikacji ETL dla sześciu modeli obiektów oraz testy jednostkowe i fukcjonlane dla przeyłu danych do bazy danych.

## 3. Zespół i realna pracochłonność

Moj zespół:

* 2x UI Dev (Senior - Indie, Mid - Indie)
* 2x Java Dev (Senior - Polska, Mid - Indie)
* 2x PL/SQL Dev (Senior - Polska, Mid - Indie)
* 4x Business Analysts (Dwóch w Polsce i dwoch w Indiach )
* 2x DevOps (Senior - Polska, Mid - Indie)

Relane czas który drużyna może poswiecić na prace :

* Spotkania oraz przewy na lunch = około 2h dziennie → 6h produktywne.
* Bankowe procedury (zmniejszają efektywność ~30%).
* Drużyna jest miedzynarodowa należy wziąc pod uwagę urlopy oraz swieta.

## 4. Ryzyka i zależności

* **Brak repozytoriów kodu** → czasochłonne odtwarzanie logiki.
* **Skomplikowane procesy bankowe** → wydłużają każdy release/test/ przygotowanie zmia oraz uzyskanie aprobat do wdrożenia .
* Kazdy release wraz z przygotowaniem kodu to trzy CHG reqesty (Service Bench UI, SC-IDP oraz jeden dla bazy danych oraz API )
* **Możliwa migracja DB na nowy serwer** → dodatkowy test performance + walidacja oraz zaangażowanie drużyny DBA Oracle w cały proces. Wraz z migracją oraz przeniesieniem bazy danych na nowy serwer będzie potzreba dodakowa wycena tego przestrzeni która będzie wykorzystywana oraz utworzonie nowych schamatów do przetrzymywania danych.
* **Duplikacja funkcji z APEX „PRET” oraz obecnego rozwiązania „ARRT”** → ryzyko robienia „tego samego 2x”.
* należy doliczyć 30% czasu więcej jako ryzko związane z zalezściami których nie znamy.

Estymacja została wykonana dla 22 tabów czas to kooło 1707 h. Nazlzy zrobić estymacje dla 22 tabów uwględniając brakujace rzeczy które tutaj wyszczególniłem oraz estymacje dla 46 tabów.

Podział godzinowy musi być rozbity na role w zespole w szczególach.

Prace muszą być podzielone na miesiace.

Koncowa rekomendacja ma być ze nie powinniśmy przenośić PRET ze wględfu na dy=uż pracochłonnośc oraz mozliwoiśc zdublowania tego dsamego co robimy w **„ARRT”.**

**Plan „B” powienien zostać określony jako plan na przeniesienie całej funkcjonalnosci z jednego serwera na drugi pod warunkiem nie tworzenia UI oraz API. PIM Recon Team powinien dalej pracowac na tej samej bazie danych jeżeli chce nadal wykonywac jakieś zadania zwiazane z manualnym uzgodnieneim danych a baza danych będzie tylko wsparcime w ich procesie. Po naszej stronie wesprzemy PIM Recon Team w utworzeniu repozytoriuów kodu pipeline, procesie ąłdowania danych do bazay danych – ETL oraz w procesie tworzenia zmain na bazie danych.**

**Utwurz dogłebną analizę biorąc pod uwage załacozne zdjecia oraz załeozenia które tutaj napisałem.**

### ------------------------

### **Dogłębna Analiza i Rewaluacja Estymacji Projektu Migracji Systemu APEX**

**Data:** 27.08.2025 **Wersja:** 1.0

### 1. Streszczenie Zarządcze (Executive Summary)

Niniejszy dokument przedstawia kompleksową analizę i rewaluację pracochłonności projektu migracji systemu z obecnej technologii Oracle APEX do docelowej architektury opartej o mikroserwisy (Java Spring Boot, GraphQL) oraz dedykowany interfejs użytkownika (UI).

Pierwotna estymacja, wynosząca **1,707 godzin** dla migracji 22 funkcjonalności ("zakładek"), została zidentyfikowana jako znacząco niedoszacowana. Nie uwzględniała ona kluczowych prac (np. DevOps, CI/CD, ETL), realnej produktywności zespołu obniżonej przez wewnętrzne procedury (~30%), ani krytycznego ryzyka związanego z nieprecyzyjnie zdefiniowanym zakresem (różne widoki dla profili Admin, Super Admin, User).

Po uwzględnieniu wszystkich czynników, nowe estymacje kształtują się następująco:

* **Zrewidowana estymacja dla 22 zakładek: ~3,134 godziny.**
* **Zrewidowana estymacja dla 46 zakładek: ~6,461 godzin.**

**Rekomendacja:** Zdecydowanie odradza się realizację migracji w pełnym zakresie 46 zakładek ("PRET"). Koszt projektu (~6,500 godzin) jest nieproporcjonalnie wysoki w stosunku do potencjalnych korzyści, a ryzyko powielenia istniejących funkcjonalności z systemu "ARRT" jest zbyt duże.

**Proponowana alternatywa (Plan "B"):** Rekomenduje się wdrożenie rozwiązania strategicznego, polegającego na migracji samej bazy danych na nowy serwer i wsparciu zespołu PIM Recon Team w implementacji dobrych praktyk (repozytoria kodu, CI/CD dla bazy danych, usprawnienie procesów ETL) bez budowania nowego UI i API. Jest to rozwiązanie znacznie szybsze, tańsze i niosące mniejsze ryzyko.

### 2. Analiza Architektury i Założeń

#### 2.1. Architektura Obecna (AS-IS)

* System oparty o **APEX na bazie danych Oracle 19c**.
* Aplikacja przechowuje i przetwarza wyniki uzgodnień danych, walidacje i zapytania SQL.
* Zakres funkcjonalny obejmuje od **22 do 46 zakładek** o różnej logice, w tym 6 funkcjonalności do manualnego wgrywania danych (upload).
* **Brak centralnego repozytorium kodu** dla logiki biznesowej.
* **Brak zautomatyzowanych procesów CI/CD** (Continuous Integration/Continuous Delivery).
* Brak wbudowanych mechanizmów wysyłki e-mail.
* **Brak jakichkolwiek testów**: jednostkowych, funkcjonalnych, wydajnościowych.

#### 2.2. Architektura Docelowa (TO-BE)

* **Całkowita eliminacja APEX**. Logika biznesowa zostanie przeniesiona do nowej architektury.
* **Nowa baza danych:** Oracle 19c na nowym, dedykowanym serwerze.
* **Warstwa API w Java Spring Boot:** Nowe punkty końcowe REST API dla każdej funkcjonalności przenoszonej z APEX.
* **Warstwa API w GraphQL:** Nowe punkty końcowe, bazujące na warstwie Spring Boot, w celu integracji z platformą Service Bench.
* **Dedykowany Interfejs Użytkownika (Custom UI):** Implementacja od zera wszystkich zakładek (minimum 22, potencjalnie 46), wraz z filtrowaniem, sortowaniem oraz mechanizmami aktualizacji danych.

#### 2.3. Nowe Wymagania Obowiązkowe

* **Repozytoria Kodu:** Utworzenie 5 dedykowanych repozytoriów dla: obiektów PL/SQL, API Java Spring Boot, testów integracyjnych API, obiektów API GraphQL.
* **Pipeline CI/CD:** Budowa i wdrożenie w pełni zautomatyzowanych pipeline'ów dla bazy danych oraz wszystkich warstw API.
* **Zarządzanie Uprawnieniami:** Utworzenie nowych ról i uprawnień w systemie SC-IDP.
* **Testy:** Implementacja testów jednostkowych i integracyjnych dla każdej nowej funkcjonalności (Java, GraphQL, PL/SQL).
* **Procesy ETL:** Utworzenie oddzielnych funkcjonalności w aplikacji ETL do zasilania 6 tabel źródłowych oraz implementacja testów dla tych procesów.

### 3. Zespół i Realna Pracochłonność

#### 3.1. Skład Zespołu

* **UI Dev:** 2 (Senior - Indie, Mid - Indie)
* **Java Dev:** 2 (Senior - Polska, Mid - Indie)
* **PL/SQL Dev:** 2 (Senior - Polska, Mid - Indie)
* **Business Analyst:** 4 (2 - Polska, 2 - Indie)
* **DevOps:** 2 (Senior - Polska, Mid - Indie)

#### 3.2. Ograniczenia Produktywności

* **Efektywny czas pracy:** Z 8-godzinnego dnia pracy, realny czas produktywny wynosi **6 godzin** (2 godziny przeznaczane są na spotkania i przerwy).
* **Procedury bankowe:** Wewnętrzne procesy i procedury zmniejszają ogólną efektywność zespołu o około **30%**.
* **Zespół międzynarodowy:** Należy uwzględnić różnice w dniach wolnych, świętach oraz okresach urlopowych w Polsce i Indiach.

### 4. Analiza i Rewaluacja Estymacji Pracochłonności

#### 4.1. Kluczowe, Nieuwzględnione Ryzyko: Zmienny Zakres

Pierwotna estymacja nie uwzględniła fundamentalnego faktu: **istnienie różnych profili użytkowników (Admin, Super Admin, User)**. Oznacza to, że realna liczba unikalnych widoków i komponentów do zaimplementowania jest nieznana i prawdopodobnie znacznie większa niż 46. Jeden moduł biznesowy może wymagać stworzenia 2-3 różnych interfejsów, co multiplikuje pracę, a w szczególności złożoność testowania.

### 5. Główne Ryzyka i Zależności

* **Brak repozytoriów kodu:** Znacząco wydłuża czasochłonną fazę odtwarzania logiki biznesowej.
* **Skomplikowane procesy bankowe:** Wydłużają każdy etap projektu: testy, przygotowanie zmian, uzyskiwanie zgód na wdrożenie.
* **Wysoki narzut administracyjny:** Każde wdrożenie wymaga trzech osobnych zgłoszeń (CHG requests), co komplikuje proces.
* **Migracja bazy danych:** Ryzyko dodatkowych prac (testy wydajnościowe, walidacja, wycena nowej infrastruktury) i konieczność zaangażowania zespołu DBA.
* **Duplikacja funkcji:** Ryzyko, że migrowane funkcje z "PRET" powielają logikę z istniejącego rozwiązania "ARRT", co oznacza marnotrawstwo zasobów.
* **Nieznane zależności:** Zidentyfikowano potrzebę dodania 30% buforu czasowego na ryzyka, które nie są obecnie znane.

### 6. Rekomendacja Końcowa

**Zdecydowanie odradza się realizację migracji funkcjonalności "PRET" w rozszerzonym zakresie (scenariusz 46 zakładek).**

**Uzasadnienie:**

1. **Nieproporcjonalnie wysoki koszt:** Szacowana pracochłonność na poziomie **~6,500 godzin** jest inwestycją o bardzo wysokim koszcie i długim czasie realizacji (ponad 6 miesięcy).
2. **Krytyczne ryzyko zakresu:** Brak jasności co do realnej liczby widoków wynikającej z profili użytkowników sprawia, że nawet 6,500 godzin może być estymacją optymistyczną.
3. **Wysokie ryzyko duplikacji pracy:** Istnieje uzasadniona obawa, że projekt będzie odtwarzał funkcjonalności, które już istnieją lub są planowane w systemie "ARRT", co podważa sens biznesowy całej migracji.

### 7. Plan "B" – Alternatywa Strategiczna

Jako rozwiązanie o znacznie lepszym stosunku kosztu do wartości, proponuje się wdrożenie Planu B.

* **Cel:** Zapewnienie zespołowi PIM Recon Team nowoczesnego i stabilnego środowiska do pracy oraz wdrożenie dobrych praktyk deweloperskich przy minimalizacji kosztów i ryzyka projektowego.
* **Elementy w zakresie:**
  1. **Wsparcie w migracji bazy danych:** Aktywna pomoc zespołowi DBA w przeniesieniu bazy danych Oracle na nowy serwer, wraz z testami i walidacją.
  2. **Wdrożenie repozytorium kodu:** Utworzenie repozytorium Git dla wszystkich obiektów PL/SQL i wsparcie zespołu w jego użytkowaniu.
  3. **Budowa CI/CD dla bazy danych:** Stworzenie zautomatyzowanego pipeline'u do wdrażania zmian na bazie danych.
  4. **Usprawnienie ETL:** Pomoc w optymalizacji i wdrożeniu zautomatyzowanych testów dla procesów zasilających 6 kluczowych tabel.
* **Elementy poza zakresem:**
  1. **Brak budowy nowego UI.**
  2. **Brak budowy warstwy API (Java, GraphQL).**
* **Korzyści:**
  1. **Drastyczna redukcja kosztów i czasu realizacji.**
  2. **Minimalizacja ryzyka** związanego z odtwarzaniem logiki biznesowej.
  3. **Natychmiastowa wartość dla zespołu PIM Recon Team**, który otrzymuje stabilne, wersjonowane i zautomatyzowane środowisko pracy.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Kategoria Pracy (Workstream) | Pierwotna Estymacja (1707h dla 22 zakładek) | Zrewidowane Godziny (Scenariusz 22 zakładki) | Zrewidowane Godziny (Scenariusz 46 zakładek) | Komentarz / Uzaszadanienie Zmian |  |  |
| A: Analysis & Discovery | 110 h | ~110 h | ~250 h | Analiza ponad dwukrotnie większej liczby widoków, z uwzględnieniem logiki dla różnych ról, wymaga znacznie więcej czasu. Skalowanie nie jest liniowe. |  |  |
| B: Page Migration (UI) | 439 h | ~700 h (w ramach subtotalu) | ~1,400 h | Implementacja ponad dwukrotnie większej liczby widoków, z których każdy może mieć warianty dla różnych ról (inne przyciski, pola, akcje). |  |  |
| C: Logic Refactor | 256 h | ~750 h (w ramach subtotalu) | ~1,300 h | Więcej endpointów API (Java, GraphQL) i logiki w bazie danych. Różne role często wymagają dedykowanych ścieżek w kodzie. |  |  |
| F: Testing (Unit/SIT/UAT) | 222 h | (w ramach subtotalu) | ~750 h | Krytyczny wzrost złożoności. Testowanie 46 widoków na 3 profilach to wielokrotność przypadków testowych. Zwiększa się ryzyko błędów w logice uprawnień. |  |  |
| Pozostałe prace deweloperskie | 458 h | (w ramach subtotalu) | ~600 h | Pozostałe zadania (migracja, integracje, dokumentacja) również skalują się wraz ze wzrostem złożoności i zakresu projektu. |  |  |
| I: DevOps (CI/CD) | 0 h | 120 h | ~150 h | Prace jednorazowe, ale zarządzanie bardziej złożonym projektem i większą liczbą komponentów wymaga nieco więcej czasu. |  |  |
| J: ETL Development | 0 h | 100 h | 100 h | Prace związane z 6 konkretnymi tabelami, więc ten koszt pozostaje stały. |  |  |
| Subtotal (przed PM/Ryzykiem) | 1,325 h | ~2,207 h | ~4,550 h | Suma powyższych zrewidowanych prac, uwzględniająca nałożony ~30% narzut za brak efektywności wynikający z procedur bankowych. |  |  |
| PM (12%) | 159 h | ~265 h | ~546 h | Zarządzanie znacznie większym i dłuższym projektem wymaga proporcjonalnie więcej czasu. |  |  |
| Ryzyko (30%) | 223 h (15%) | ~662 h | ~1,365 h | Bufor ryzyka jest absolutnie kluczowy. Przy tak dużej liczbie niewiadomych (role, realna liczba widoków) 30% to minimum. |  |  |
| TOTAL | 1,707 h | ~3,134 h | ~6,461 h | Realny szacunek dla 46 zakładek z uwzględnieniem ról jest prawie 4x wyższy od pierwotnej estymacji. |  |  |