

Odejścia pracowników i ich czynniki

Specyfikacja



4 marca 2023

GRZEGORZ BORKOWSKI

Wersja 3.0

Odejścia pracowników i ich czynniki

Spis treści

[1. Cel i opis projektu 1](#_Toc131025456)

[2. Kluczowe wymagania klienta 2](#_Toc131025457)

[3. Przeprowadzone analizy na podstawie danych ogólnodostępnych 2](#_Toc131025458)

[4. Obecna architektura klienta 3](#_Toc131025459)

[5. Data dictionary (słownik danych) 4](#_Toc131025460)

[6. Decision logs (dzienniki decyzji) – uzgodnione z klientem 5](#_Toc131025461)

[7. Medallion architecture (architektura medalionu) 7](#_Toc131025462)

[8. Zaproponowana logiczna architektura 8](#_Toc131025463)

[9. Product backlog (Backlog produktu) 9](#_Toc131025464)

[10. Wykres Gantta z podziałem na zadania i epici 12](#_Toc131025465)

[11. Katalog produktów 13](#_Toc131025466)

[12. Makieta wizualizacji 14](#_Toc131025467)

[13. Proof of Concept (Sprawdzenie koncepcji) 15](#_Toc131025468)

[13.1. Serwisy w Microsoft Azure Cloud 15](#_Toc131025469)

[13.2. Przygotowanie i konfiguracja narzędzi 22](#_Toc131025470)

[13.3. Proces ELTL 31](#_Toc131025471)

[14. Czas trwania oraz koszt projektu 35](#_Toc131025472)

# Cel i opis projektu

Celem projektu jest wielowymiarowa analiza czynników i motywacji pracowników odchodzących z firmy klienta, jak również zestawienie tych danych z publicznie dostępnymi informacjami w tej tematyce w celu zrozumienia faktorów na to wpływających, jak również zmniejszeniu ilości odejść tychże pracowników. Jako pracowników odchodzących, klient kwalifikuje:

1. Pracowników, którzy odeszli z pracy przed końcem umowy
2. Pracowników, którzy odeszli z pracy po zakończeniu umowy (pomimo otrzymania propozycji przedłużenia)

Zgodnie z informacjami od klienta, analizie **nie będą** podlegać pracownicy, którzy zostali zwolnieni, jak również pracownicy, którzy byli na stażu.

# Kluczowe wymagania klienta

Klient jest elastyczny pod względem metryk oraz wniosków, które będą wynikiem analizy, aczkolwiek MVP (Minimum Viable Product) jest:

1. Analiza odejść pracowników na podstawie wieku pracownika
2. Analiza odejść pracowników na podstawie departamentu pracownika
3. Analiza odejść pracowników na podstawie sumarycznego stażu pracy
4. Analiza odejść pracowników na podstawie stażu pracy u klienta
5. Analiza odejść pracowników na podstawie płci

Jako narzędzie do wizualizacji danych, klient wybrał narzędzie **Power BI**. Klient nie posiada subskrypcji do tego narzędzia, dlatego też ich koszt zostanie uwzględniony w części kosztowej

Do celów integracji danych, przepływu danych, transformacji, przechowywania danych oraz zarządzania danymi (m.in. Data Governance), klient zdecydował się na środowisko Azure. Klient otrzymał wstępne wyceny, poznał mocne i słabe strony rozwiązań chmurowych od różnych dostawców oraz rozwiązań on-premise, stąd też nie będą proponowane inne rozwiązania. Szczegółowe informacje znajdą swoje miejsce w decision logach w dalszej części dokumentu.

# Przeprowadzone analizy na podstawie danych ogólnodostępnych

W celu lepszej analizy danych, korzystając z danych, statystyk oraz informacji dostępnych publicznie, stworzyłem listę wniosków, które będą przede wszystkim brane pod uwagę przy tworzeniu tego projektu. Należy pamiętać, że każda firma posiada dane w innym formacie oraz analizuje inne typy danych, dlatego też nie wszystkie z podanych wniosków będą miały zastosowanie lub będą mogły zostać użyte opierając się na danych otrzymanych od klienta.

Do najważniejszych faktorów zaliczamy:

* Przepracowanie
* Nieadekwatne zarobki
* Brak możliwości wzrostu i rozwoju
* Brak innowacji lub poczucia celu
* Złe zarządzanie
* Brak elastyczności w pracy
* Niewystarczające oferty pracy (brak ofert dla osób bardziej doświadczonych, brak możliwości awansu)
* Złe środowisko pracy
* Brak motywacji
* Zły balans między życiem zawodowym a osobistym
* Brak poczucia przynależności do zespołu
* Osobiste motywacje
* Problemy w pracy

Źródła danych:

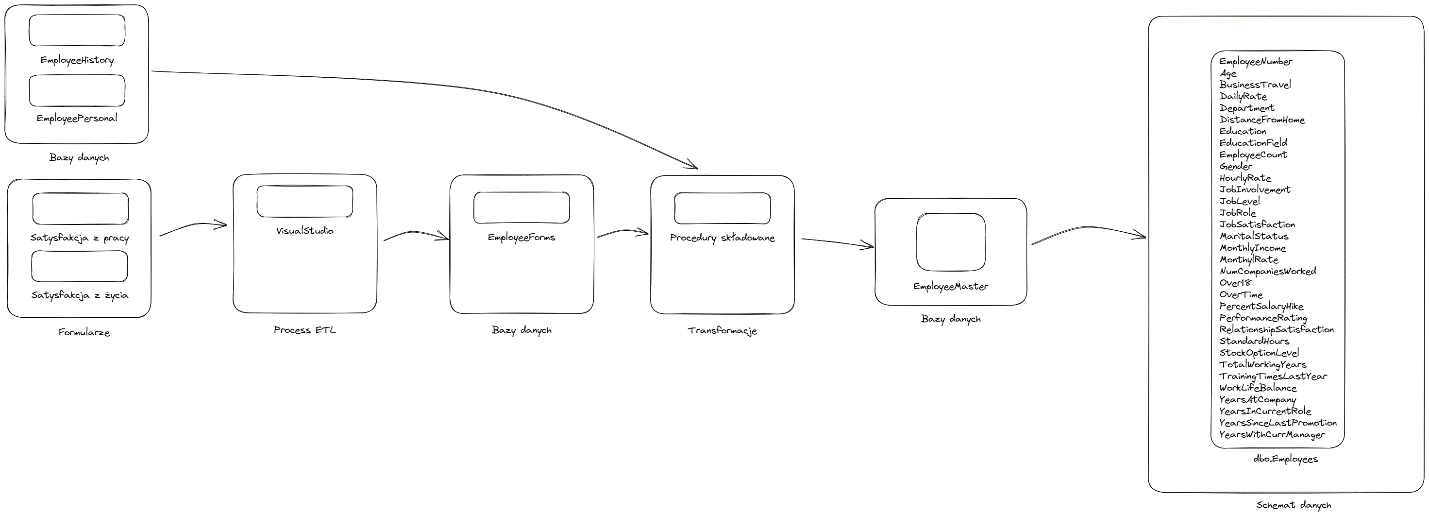
* <https://splashbi.com/employee-attrition-definition-types-best-practices>
* <https://www.betterup.com/blog/employee-attrition>
* https://www.spiceworks.com/hr/engagement-retention/articles/what-is-attrition-complete-guide/

# Obecna architektura klienta

Klient posiada swoje dane w architekturze on-premise w środowisku bazodanowym SQL Server 2016. Klient korzysta z następujących baz danych:

* EmployeeHistory
* EmployeePersonal
* EmployeeMaster

Dane są pobierane z baz danych EmployeeHistory i EmployeePersonal oraz z formularzy: Satysfakcja z pracy oraz Satysfakcja z życia.



Rysunek 1 - Obecna architektura klienta (większa rozdzielczość w pliku obecna\_architektura.png)

Zgodnie z ustaleniami, klient nie chce udostępniać dostępu do baz danych EmployeePersonal oraz EmployeeHistory z uwagi na wysoce wrażliwe dane (np. numer PESEL, numer dowodu osobistego), dlatego też dane będą wyciągane bezpośrednio z bazy EmployeeMaster. Klient w przeszłości próbował używać tabeli dbo.Employees w rozwiązaniach Big Data, dlatego też tabela ma postać tabeli zdenormalizowanej.

# Data dictionary (słownik danych)

Na spotkaniu przedwdrożeniowym klient udostępnił słownik pojęć, który definiuje nazwę kolumny oraz opis kolumny.

|  |  |
| --- | --- |
| Nazwa kolumny | Opis |
| Age | Wiek |
| Attrition | Flaga czy pracownik jest zwolniony (0 - nie, 1 - tak) |
| BusinessTravel | Jak często pracownik podróżuje w ramach pracy |
| DailyRate | Stawka dzienna |
| Department | Oddział |
| DistanceFromHome | Odległość od domu do miejsca pracy |
| Education | Stopień edukacji (1-5) |
| EducationField | Ścieżka edukacji |
| EmployeeCount | Ilość pracowników, używane do sprawdzania duplikatów |
| EmployeeNumber | Numer pracownika |
| EnvironmentSatisfaction | Satysfakcja z miejsca pracy (1-5) |
| Gender | Płeć |
| HourlyRate | Stawka godzinowa |
| JobInvolvement | Zaangażowanie w pracę (1-5) |
| JobLevel | Poziom pracownika (1-5) |
| JobRole | Nazwa roli |
| JobSatisfaction | Satysfakcja z pracy (1-5) |
| MaritalStatus | Stan cywilny |
| MonthlyIncome | Miesięczny dochód (po uwzględnieniu podatków) |
| MonthlyRate | Stawka miesięczna (bez uwzględnienia podatków) |
| NumCompaniesWorked | Ilość miejsc pracy, w których pracownik pracował wcześniej |
| Over18 | Czy pracownik jest pełnoletni (0 - nie, 1 - tak) |
| OverTime | Czy pracownik bierze nadgodziny (No - nie, Yes - tak) |
| PercentSalaryHike | Procentowa podwyżka wynagrodzenia |
| PerformanceRating | Wydajność pracownika (1-5) |
| RelationshipSatisfaction | Satysfakcja w związku (1-5) |
| StandardHours | Średnia ilość godzin pracy |
| StockOptionLevel | Poziom posiadania akcji firmy (0-5) |
| TotalWorkingYears | Staż pracownika w latach |
| TrainingTimesLastYear | Ilość szkoleń w ostatnim roku |
| WorkLifeBalance | Poziom równowagi między pracą, a życiem prywatnym (1-5) |
| YearsAtCompany | Staż pracownika w obecnej firmie w latach |
| YearsInCurrentRole | Ilość lat na obecnym stanowisku |
| YearsSinceLastPromotion | Ilość lat od ostatniego awansu |
| YearsWithCurrManager | Ilość lat z obecnym menadżerem |

# Decision logs (dzienniki decyzji) – uzgodnione z klientem

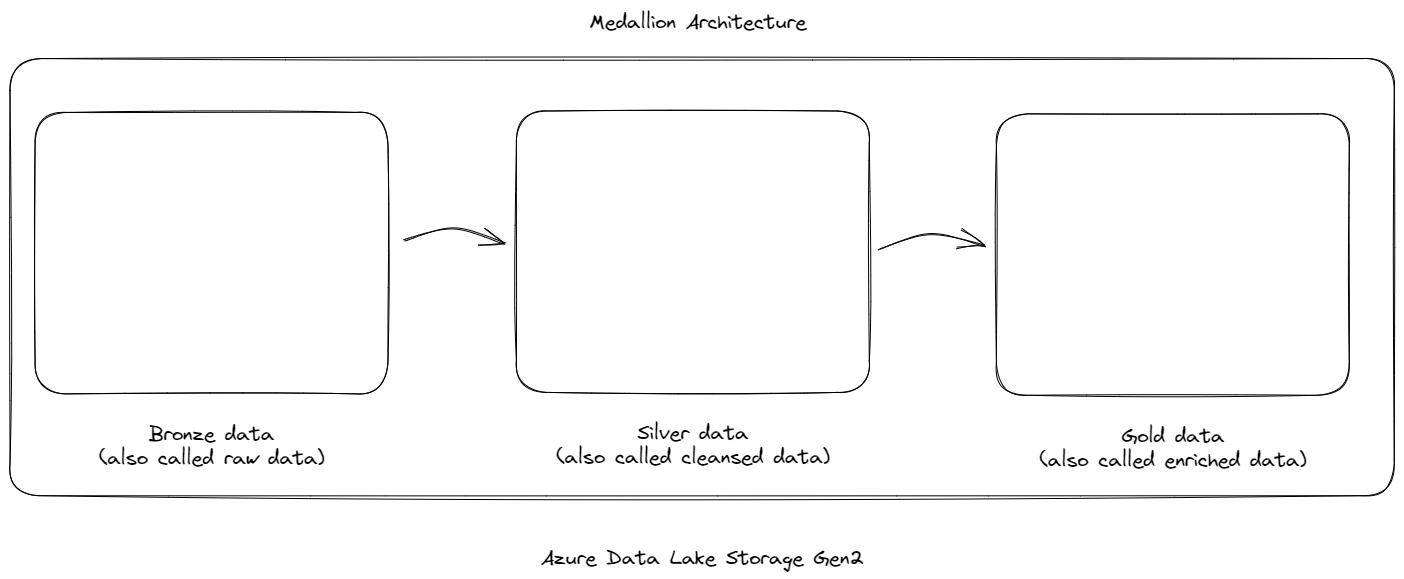
Decision logs przedstawiają decyzje, które zostały przygotowane i opracowane przez Grzegorza Borkowskiego, a zostały podjęte przez osobę decyzyjną ze strony klienta (Krzysztofa Drużkowskiego).

|  |  |  |  |
| --- | --- | --- | --- |
| Nazwa | Opis | Proponowane opcje | Podjęta decyzja |
| Hurtownia danych | Zdefiniowanie lokalizacji hurtowni danych, w której będą przechowywane dane używane do wizualizacji | * On-premise (SQL Server) * Azure (Azure SQL Database) * Azure (Azure Synapse) | Azure (Azure SQL Database) |
| Narzędzie ETL | Narzędzie używane do integracji, przenoszenia danych pomiędzy środowiskiem klientem a hurtownią danych oraz do transformacji danych | * On-premise (Apache Airflow) * Azure (Azure Data Factory) + Azure Databricks * Azure (Azure Databricks) * Azure (Azure Synapse) | Azure (Azure Data Factory) + Azure Databricks |
| Wizualizacja danych | Narzędzie do wizualizacji danych | * On-premise (Power BI Report Server) * Microsoft Power BI * QlikView * Tableau | Microsoft Power BI |
| Pośrednia warstwa przechowywania danych | Pośrednia warstwa przechowywania danych jest używana jako tymczasowe miejsce, w którym dane będą przechowywane w celu zapewnienia wysoko wydajnego storage’ a do dalszej transformacji danych | * On-premise (dyski NAS/NTFS) * Azure (Azure Data Lake Storage Gen2) * Azure (Azure File Shares) * Azure (Azure Blob Storage) | Azure (Azure Data Lake Storage Gen2) |
| Retencja danych | Ilość dni, po których dane mają zostać usunięte lub przeniesione do innej warstwy | N/A | Dane mają zostać:   * Przeniesione z warstwy hot do warstwy cold po 30 dniach * Przeniesione z warstwy cold do warstwy archive po 180 dniach * Usunięte po 300 dniach od daty utworzenia |
| Historia wersji danych | Historia wersji danych jest istotnym mechanizmem przy przechowywaniu danych. Dzięki niej możemy porównywać dane z danymi z przeszłości celem zestawienia oraz dalszej analizy, np. aby sprawdzić, czy zmniejsza się ilość pracowników odchodzących z pracy) | N/A | Będziemy tworzyć snapshot danych codziennie, natomiast ostatniego dnia miesiąca będziemy zachowywać tylko dane z pierwszego dnia miesiąca (reszta będzie usuwana). Klient będzie chciał porównać maksymalnie 6 miesięcy |
| Data Governance | Data Governance jest bardzo szerokim obszarem, którego celem jest łatwiejsze zarządzanie danymi, klasyfikacja danych, zarządzanie uprawnieniami do danych, sprawdzanie jakości danych i inne. | * On-premise (IBM InfoSphere) * Azure (Azure Purview) * Azure (Databricks Unity Catalog) | Azure (Databricks Unity Catalog) |
| Schemat hurtowni danych | Logiczny model hurtowni danych | * Kimball * Data Vault * 3NF | Kimball |
| Architektura rozwiązania | Architektura odnosi się do logicznego planu wykonania, relacjami między danymi oraz między zespołami, jak również do używania wytycznych określonych w ramach danej architektury | * MDA (Klasyczny proces) * MDA (Data Mesh) | MDA (Klasyczny proces) |
| Kontrola wersji | Kontrola wersji służy do synchronicznej pracy nad kodem, jak również pozwala śledzić zmiany, łączyć zmiany i testować wrzucony kod celem utrzymania polityki czystego kodu | * GitHub * Azure DevOps * GitLab | Azure DevOps |
| Narzędzie do zarządzania projektem | W celu zarządzania projektu konieczne jest narzędzie, które pozwoli na dodawanie zadań, kontrolę postępów oraz kontrolę estymowanych czasów | * Jira * Azure DevOps * Trello | Azure DevOps |
| Architektura danych | Architektura danych określa logiczną organizację danych (w naszym przypadku w lakehouse, czyli ADLS) | * Medallion architecture * Data Lake Architecture | Medallion architecture |

# Medallion architecture (architektura medalionu)

Architektura medalionu polega na logicznej organizacji danych w kontenerach ADLS Gen2. W architekturze tej wyróżniamy 3 warstwy:

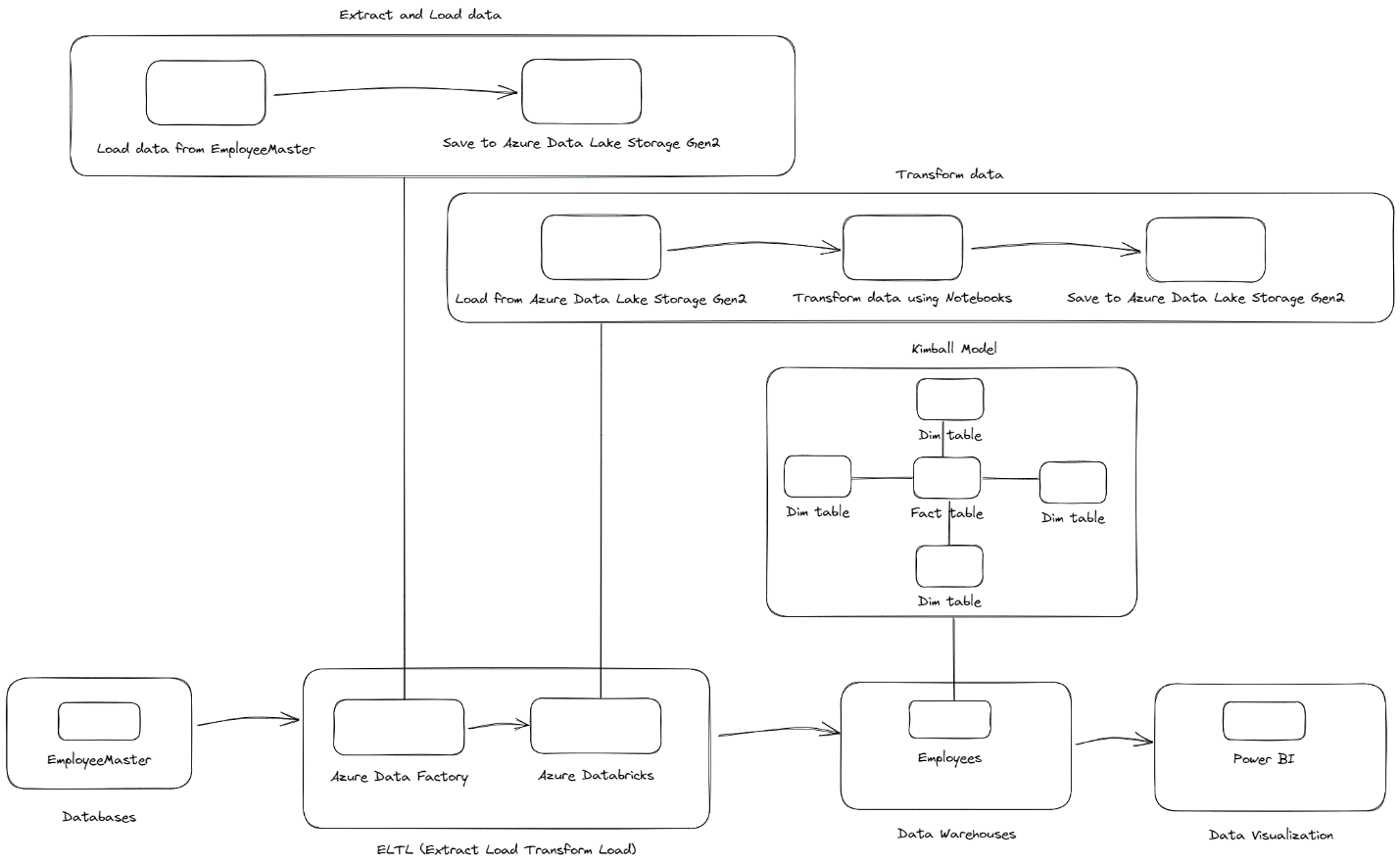
1. Bronze – jest to warstwa, do której trafiają dane z systemów źródłowych. W naszym przypadku będą to dane z bazy danych EmployeeMaster z tabeli Employees
2. Silver – jest to warstwa danych wyczyszczonych i gotowych na dalsze obróbki. W warstwie tej powinny się znajdować dane przefiltrowane, zwalidowane i pozbawione złych rekordów
3. Gold – jest to ostatnia warstwa, w której wyczyszczone dane są agregowane, łączone ze sobą i budują w ten sposób dane gotowe dla biznesu. W naszym przypadku w warstwie tej będziemy normalizować dane i tworzyć wymiary i fakty.



Rysunek 2 - Architektura medalionu (pełne zdjęcie dostępne w załączniku medallion\_architecture.png)

# Zaproponowana logiczna architektura

Mając na uwadze posiadane dane przez klienta, jak również oczekiwania i wymagania, Grzegorz Borkowski przygotował i opracował architekturę danych. Architektura została omówiona z Krzysztofem Drużkowski i jego zespołem i została zaakceptowana.



Rysunek 3 - Zaproponowana logiczna architektura (większa rozdzielczość dostępna w pliku zaproponowana\_logiczna\_architektura.png)

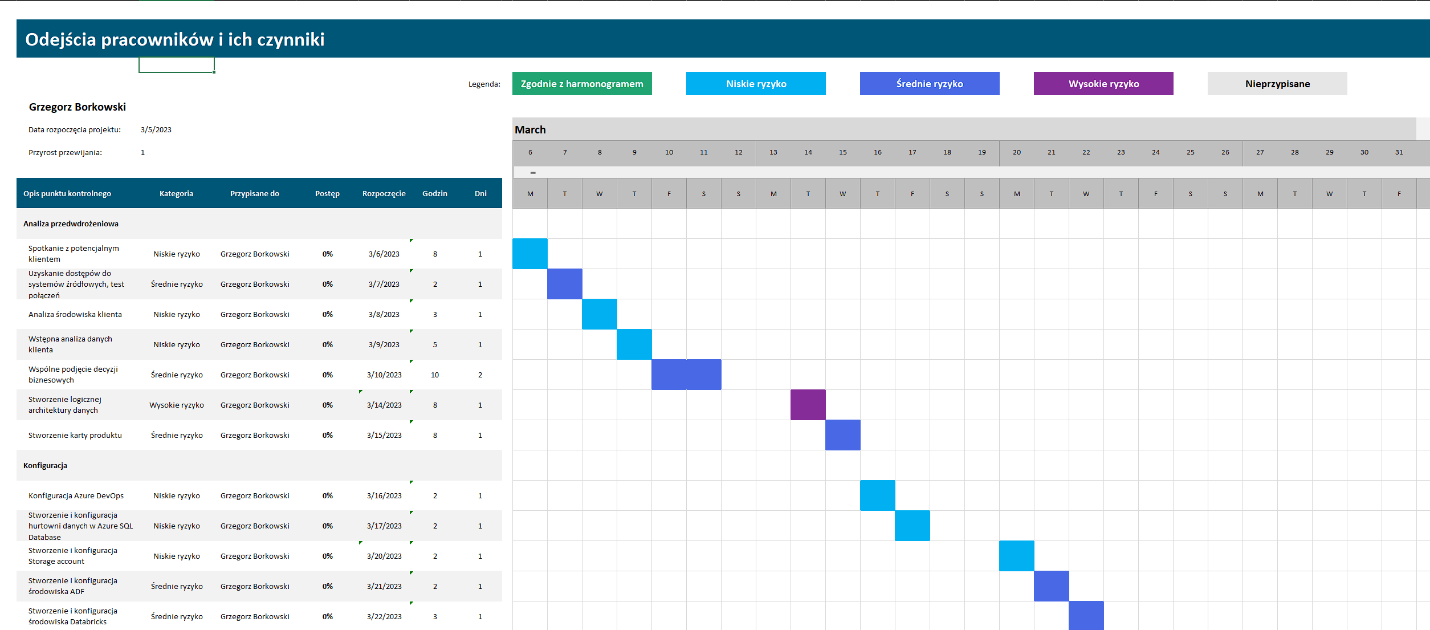
# Product backlog (Backlog produktu)

Backlog produktu listuje oraz opisuje zadania, które należy wykonać celem osiągnięcia zakładanego przez klienta rezultatu. Zadania zostały opracowane oraz wyestymowane przez Grzegorza Borkowskiego i przedstawione poniżej w postaci tabeli z podzieleniem na epici.

|  |  |  |  |
| --- | --- | --- | --- |
| Nazwa zadania | Opis | Epic | Estymowany czas |
| Spotkanie z potencjalnym klientem | Spotkanie w siedzibie klienta lub zdalnie, zebranie wymagań, celu projektu oraz poznanie obecnego środowiska | Analiza przedwdrożeniowa | 8h |
| Uzyskanie dostępów do systemów źródłowych, test połączeń | Otrzymanie oraz przetestowanie dostępów | Analiza przedwdrożeniowa | 2h |
| Analiza środowiska klienta | Analiza pod względem wydajnościowym (jak np. ilość pamięci, ilość procesorów) jak również poznanie systemów klienta | Analiza przedwdrożeniowa | 3h |
| Wstępna analiza danych klienta | Analiza danych, które mają być użyte do stworzenia aplikacji | Analiza przedwdrożeniowa | 5h |
| Wspólne podjęcie decyzji biznesowych | Opracowanie decision logów, rekomendacja rozwiązań oraz wspólne podjęcie decyzji z klientem | Analiza przedwdrożeniowa | 10h |
| Stworzenie logicznej architektury danych | Opracowanie i stworzenie logicznej architektury, która będzie zgodna z decision logami oraz wymaganiami klienta | Analiza przedwdrożeniowa | 8h |
| Stworzenie karty produktu | Zebranie wszystkich informacji, oszacowanie czasu wdrożenia, utworzenie backlogu produktu, estymacja kosztów | Analiza przedwdrożeniowa | 8h |
| Konfiguracja Azure DevOps | Konfiguracja kontroli wersji oraz narzędzia do zarządzania projektami | Konfiguracja | 2h |
| Stworzenie i konfiguracja hurtowni danych w Azure SQL Database | Dodanie nowej hurtowni danych, ustawienie odpowiedniej wielkości | Konfiguracja | 2h |
| Stworzenie i konfiguracja Storage account | Konfiguracja storage account, na którym będzie postawiony Azure Data Lake Storage Gen2 | Konfiguracja | 2h |
| Stworzenie i konfiguracja środowiska ADF | Skonfigurowanie całego środowiska Azure Data Factory, w tym zestawienie połączeń (Linked Services), podpięcie do Gita | Konfiguracja | 2h |
| Stworzenie i konfiguracja środowiska Databricks | Skonfigurowanie środowiska Databricks, w tym utworzenie mount do ADLS, skonfigurowanie Gita | Konfiguracja | 3h |
| Ustawienie lifecycle management w Storage Account | Określenie po jakim czasie dane mają być przenoszone lub usuwane | Konfiguracja | 2h |
| Stworzenie przepływów danych w ADF | Utworzenie przepływów danych, dodanie datasetów | Development | 4h |
| Utworzenie transformacji danych w Databricks | Utworzenie transformacji danych, w tym: czyszczenie danych, normalizacja danych, stworzenie wymiarów i faktów, stworzenie procesu do usuwania danych | Development | 12h |
| Konfiguracja Unity Catalog | Zdefiniowanie uprawnień do danych, nadanie dostępów | Development | 6h |
| Stworzenie modelu danych w Azure SQL Database | Utworzenie modelu danych zgodnego z modelem stworzonym w Databricks | Development | 3h |
| Test rozwiązania, wdrożenie ewentualnych poprawek | Przetestowanie całego procesu od pobierania danych do przeniesienia danych do hurtowni danych | Development | 4h |
| Konfiguracja środowiska Power BI | Konfiguracja środowiska, w tym m.in. zdefiniowanie połączenia z danymi, zaciągniecie danych, stworzenie modelu danych | Wizualizacje | 4h |
| Konfiguracja środowiska Power BI Service | Zakup potrzebnych licencji, ustawienie poprawnych uprawnień | Wizualizacje | 4h |
| Stworzenie raportów (do rozdzielenia) | Stworzenie raportów, które będą zgodne z wymaganiami klienta | Wizualizacje | 24h |
| Weryfikacja poprawności danych | Weryfikacja poprawności danych wraz z klientem oraz wewnętrznie | Oddanie projektu | 8h |
| Warsztat z używania narzędzia | Warsztat na którym zostaną przedstawione raporty oraz zostanie zaprezentowana funkcjonalność raportów | Oddanie projektu | 8h |
| Support powdrożeniowy | Support celem odpowiedzi na pytania klienta, sprawdzenie poprawności danych, przeszkolenie klienta | Oddanie projektu | 6h |

# Wykres Gantta z podziałem na zadania i epici

W wykresie Gantta zostały dodane zadania z backlogu produktu z podziałem na epici oraz z podziałem na odpowiedzialność i szacowany czas zakończenia każdego z etapu. Zespół będzie pracował w 2 tygodniowych sprintach.



Rysunek 4 - Wykres Gantta (pełny dokument dostępny pod nazwą Wykres\_Gantta.xlsx)

# Katalog produktów

Katalog produktów ma za zadanie wylistować oraz opisać produkty, które będą używane w zaproponowanej architekturze, jak również przedstawia estymowaną ilość zasobów do sprawnego i efektywnego działania.

|  |  |  |
| --- | --- | --- |
| Nazwa produktu | Opis produktu | Estymowane zasoby |
| Azure Data Factory | Narzędzie używane do procesu ELTL. Jego zadaniem w tym projekcie będzie pobieranie danych z bazy danych klienta, ładowanie tych danych do warstwy bronze oraz uruchamianie notebooków z Azure Databricks | 100 000 zapytań miesięcznie |
| Azure Databricks | Będzie używany do transformacji danych, jak również do Data Governance za pomocą Unity Catalog | All-Purpose Compute workload w Tier Premium. Średnio 1 wirtualna maszyna pracująca przez 30 godzin w miesiącu na najmniejszym możliwym klastrze (D4a) |
| Unity Catalog | Narzędzie wchodzące w Azure Databricks. Będzie używane do nadawania dostępów do danych | Nie dotyczy |
| Storage Account | Azure Data Lake Storage Gen2 na Storage Account będzie tymczasowym miejscem do przechowywania danych. To właśnie w nim będzie używana architektura medalionu (bronze -> silver -> gold) | 50GB danych w warstwie Hot oraz w Redundancy LRS |
| Azure SQL Database | Baza danych, która w przypadku tego projektu będzie używana jako hurtownia danych | 50GB wolnego miejsca, płacone per DTU |
| Microsoft Power BI | Narzędzie do wizualizacji danych | 5 subskrypcji |

# Makieta wizualizacji

Makieta wizualizacji przedstawia schemat graficzny raportów Power BI, które będą dostępne w warstwie wizualizacji. Standardem makiety jest:

* Tytuł strony w górnej części raportu
* Co najmniej 3 wizualizacje, które będą się znajdowały poniżej tytułu strony
* Każda z wizualizacji będzie posiadała tytuł wyrównany do lewej strony oraz legendę (jeżeli wybrany typ wizualizacji na to pozwala)

Poniżej znajduje się przykładowa wizualizacja stworzona na podstawie zrandomizowanych danych.

Obraz zawierający wykres

Opis wygenerowany automatycznie

Rysunek 5- Makieta wizualizacji w Power BI (pełne zdjęcie widoczne w Makieta\_wizualizacji.png)

# Proof of Concept (Sprawdzenie koncepcji)

Proof of Concept (w skrócie PoC) ma za zadanie sprawdzić opisane założenia na bardzo wysokim poziomie aby sprawdzić oraz potwierdzić możliwość wykonania zaproponowanego scenariusza. PoC nie jest całym i gotowym produktem, dlatego też głównie jego rolą jest sprawdzenie połączeń, stworzenie prostych transformacji oraz wizualizacji w celu przetestowania połączenia pomiędzy różnymi usługami i potwierdzenia postawionych założeń.

## Serwisy w Microsoft Azure Cloud

Na potrzeby PoC, został stworzony nowy Resource group w Portalu Azure (https://portal.azure.com/ ) o nazwie **odejscia\_pracownikow\_rg** w regionie **(Europe) UK South**

Obraz zawierający tekst, zrzut ekranu, monitor, ekran

Opis wygenerowany automatycznie

Rysunek 6 - Stworzenie resource group

Kolejnym krokiem było stworzenie potrzebnych serwisów:

1. Azure Data Factory (W poprzednio utworzonej Resource group, o nazwie **odejscia-pracownikow-adf**, w regionie **UK South** w wersji **V2**Obraz zawierający tekst, zrzut ekranu, monitor, ekran

   Opis wygenerowany automatycznie

Rysunek 7 - Stworzenie Azure Data Factory

1. Storage Account (W poprzednio utworzonej Resource group, o nazwie **odejsciapracownikowadls**, w regionie **(Europe) UK South**, w **standardowym** tierze oraz w Redundancy **Locally-redundant storage (LRS)**. Obraz zawierający tekst

   Opis wygenerowany automatycznie

Rysunek 8 - Stworzenie storage account

Dodatkowo została zaznaczona opcja **Enable hierarchical namespace** celem używania Azure Data Lake Storage Gen2  
**Obraz zawierający tekst

Opis wygenerowany automatycznie**

Rysunek 9 - Stworzenie storage account

1. Azure Databricks (W poprzednio utworzonej Resource group, o nazwie **odejscia\_pracownikow\_adb**, w regionie **UK South** oraz w Pricing Tier **Premium (+ Role-based access controls)**Obraz zawierający tekst

   Opis wygenerowany automatycznie

Rysunek 10 - Stworzenie Azure Databricks workspace

1. SQL Database Server (W poprzednio utworzonej Resource group, o nazwie **odejscia-pracownikow-sqlserver.database.windows.net**, w regionie **(Europe) UK South**, z autentykacją **zarówno** **za pomocą SQL, jak i Azure AD.** W celu autentykacji SQL zostało utworzone konto admina.  
   Obraz zawierający tekst

   Opis wygenerowany automatycznie

Rysunek 11 - Stworzenie SQL Database Server

1. Azure SQL Database (W poprzednio utworzonej Resource group, o nazwie **odejscia\_pracownikow\_db** oraz **odejscia\_pracownikow\_dwh**, na poprzednio utworzonym SQL Serverze, z bazą danych w Service tier **Basic (For less demanding workloads)**, z 2GB miejsca na dane.

Obraz zawierający tekst, zrzut ekranu, monitor, ekran

Opis wygenerowany automatycznie

Rysunek 12 - Stworzenie Azure SQL Database (źródło danych)

Obraz zawierający tekst, zrzut ekranu, monitor, ekran

Opis wygenerowany automatycznie

Rysunek 13 – Stworzenie Azure SQL Database (docelowa hurtownia danych)

## Przygotowanie i konfiguracja narzędzi

Po utworzeniu powyższych serwisów, do bazy danych w Azure SQL Database zostały załadowane dane dotyczące pracowników.   
Obraz zawierający stół

Opis wygenerowany automatycznie

Rysunek 14 - Dane załadowane w Azure SQL Database

Następnie, na podstawie struktury danych ze źródła została stworzona tabeli w hurtowni danych  
Obraz zawierający tekst

Opis wygenerowany automatycznie

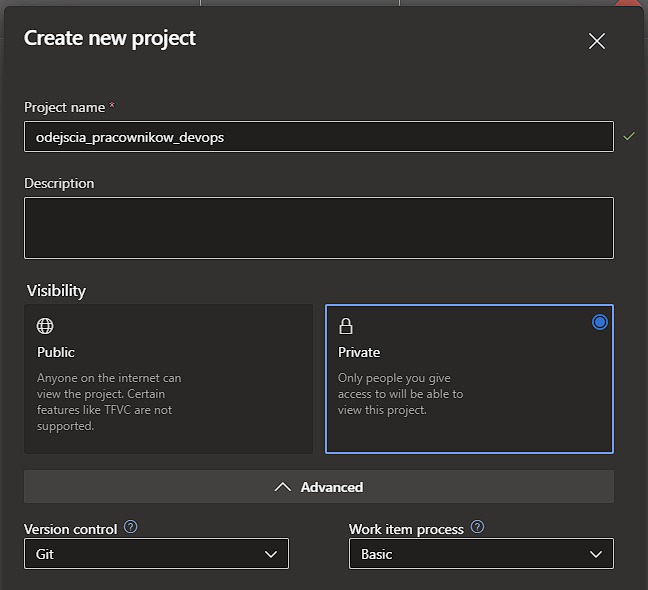
Rysunek 15 - Stworzenie pustej tabeli w hurtowni danych

Następnie, w Storage account został utworzony nowy kontener w Azure Data Lake Storage Gen2 o nazwie **employee-data** z folderami o nazwach **bronze, silver oraz gold**  
Obraz zawierający tekst, zrzut ekranu, w pomieszczeniu, czarne

Opis wygenerowany automatycznie

Rysunek 16 - Utworzony kontener i foldery w Azure Data Lake Storage Gen2

Kolejnym krokiem było utworzenie projektu w Azure DevOps o nazwie **odejscia\_pracownikow\_devops**, do którego będą trafiać zadania, jak również będzie obsługiwana kontrola wersji w Git.



Rysunek 17 - Utworzenie nowego projektu w Azure DevOps

Następnym krokiem było połączenia usługi Git z Azure DevOps z Azure Databricks  
Obraz zawierający tekst

Opis wygenerowany automatycznie

Rysunek 18 - Integracja Gita z Azure Databricks

Podobnie jak w przypadku Azure Databricks, usług Azure Data Factory również została połączona z usługą Git

Obraz zawierający stół

Opis wygenerowany automatycznie

Rysunek 19 - Integracja Git z Azure Data Factory

W Azure Data Factory zostały dodane Linked services, które zestawiają połączenie z Azure SQL Database, Azure Data Lake Storage Gen2 oraz Azure Databricks  
 Obraz zawierający tekst

Opis wygenerowany automatycznie

Rysunek 20 - Połączenie z Azure SQL Database (źródło danych)

Obraz zawierający tekst

Opis wygenerowany automatycznie

Rysunek 21 - Połączenie z Azure SQL Database (hurtownia danych)

Obraz zawierający tekst

Opis wygenerowany automatycznie

Rysunek 22 - Połączenie z Azure Data Lake Storage Gen2

Obraz zawierający tekst

Opis wygenerowany automatycznie

Rysunek 23 - Połączenie z Azure Databricks

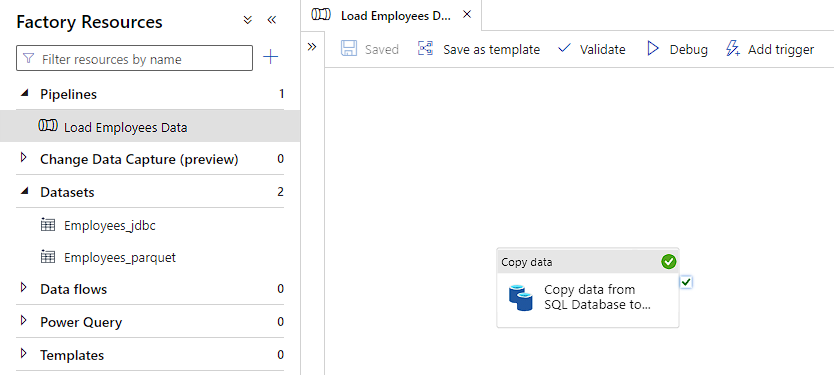
Kolejnym krokiem jest utworzenie nowego klastra w Azure Databricks na którym będą przeprowadzane transformacje  
Obraz zawierający tekst

Opis wygenerowany automatycznie

Rysunek 24 - Utworzenie nowego klastra

## Proces ELTL

Po skonfigurowaniu usług Azure, został stworzony proces ELTL który wygląda następująco:

1. W Azure Data Factory dane są wyciągane z Azure SQL Database, z bazy danych odejscia\_pracownikow\_db z tabeli Employees, a następnie ładowane do Azure Data Lake Storage Gen2 do kontenera employee-data oraz folderu bronze.  
   

Rysunek 25 - Dane załadowane do Azure Data Lake Storage Gen2

1. W Azure Databricks dane są pobierane z Azure Data Lake Storage Gen2, czyszczone, agregowane, wzbogacane i ładowane do folderów silver/gold  
   Obraz zawierający tekst

   Opis wygenerowany automatycznie

Rysunek 26 - Transformacje w Azure Databricks

1. W Azure Data Factory tworzony jest nowy pipeline, który służy jako orkiestrator do systematycznego wykonywania transformacji w Azure Databricks  
   Obraz zawierający tekst

   Opis wygenerowany automatycznie

Rysunek 27 - Utworzenie pipeline do automatycznego wykonywania transformacji w Azure Databricks

1. W Azure Data Factory tworzony jest kolejny pipeline, który ma za zadanie ładować przygotowane dane z warstwy gold do wcześniej utworzonej hurtowni danych  
   Obraz zawierający tekst

   Opis wygenerowany automatycznie

Rysunek 28 - Dane z warstwy gold są ładowane do hurtowni danych

1. Po przetestowaniu pipelineów, tworzony jest jeden zbiorczy pipeline, który zawiera w sobie cały proces ELTL  
   Obraz zawierający tekst

   Opis wygenerowany automatycznie

Rysunek 29 - Stworzenie jednego wspólnego pipeline

1. Dane z hurtowni danych są ładowane do Power BI i na tej podstawie tworzone są wizualizacje oraz raporty  
   Obraz zawierający wykres

   Opis wygenerowany automatycznie

Rysunek 30 - Raport w Power BI stworzony na podstawie danych z hurtowni danych

# Czas trwania oraz koszt projektu

Zgodnie z wykresem Gantta przedstawionym w punkcie 8 (oraz w całości w załączniku Wykres\_Gantta.xlsx), przy dacie rozpoczęcia projektu wynoszącej **5 marca 2023 roku**, projekt zostanie ukończony dnia **12 kwietnia 2023 roku**.

Estymowana ilość godzin do ukończenia projektu to **140h**, gdzie:

1. Analiza przedwdrożeniowa rozpocznie się 6 marca 2023 roku, a zakończy 15 marca 2023 roku **(44h)**
2. Konfiguracja rozpocznie się 16 marca 2023 roku, a zakończy 23 marca 2023 roku **(13h)**
3. Development rozpocznie się 24 marca 2023 roku, a zakończy 31 marca 2023 roku **(29h)**
4. Wizualizacja rozpocznie się 3 kwietnia 2023 roku, a zakończy się 8 kwietnia 2023 roku **(32h)**
5. Oddanie projektu rozpocznie się 10 kwietnia, a zakończy się 12 kwietnia 2023 roku **(22h)**

Stawka godzinowa za roboczogodzinę wynosi **250 zł**. Koszt realizacji całego projektu wyniesie **35 000 z**ł.

Dodatkowo, do powyższej kwoty należy doliczyć miesięczny koszt utrzymania, tj. koszt subskrypcji (5 subskrypcji Power BI Pro) oraz koszt środowiska chmurowego (tj. Storage Account, Azure Data Factory, Azure Databricks, Azure SQL Database). Estymowane koszty wyglądają następująco:

1. Power BI: **68.5 USD/msc** (przy aktualnym koszcie licencji na poziomie 13.7 USD/msc za licencje - <https://powerbi.microsoft.com/en-au/pricing/> )
2. Storage Account: **10 USD/msc**
3. Azure Data Factory: **10 USD/msc**
4. Azure Databricks: **20 USD/msc**
5. Azure SQL Database: **10 USD/msc**

Łączny **miesięczny** koszt utrzymania architektury wynosi **118.5 USD**.