



---

# POLITECHNIKA POZNAŃSKA

---

WYDZIAŁ AUTOMATYKI, ROBOTYKI I ELEKTROTECHNIKI  
Instytut Robotyki i Inteligencji Maszynowej



Praca dyplomowa inżynierska

## AUTOMATYZACJA PROCESU PLANOWANIA WYDATKÓW NA USŁUGI IT W VW POZNAŃ Z WYKORZYSTANIEM MICROSOFT POWER PLATFORM

Remigiusz Wolniak, 151192

Michał Gajdzis, 151066

Promotor  
dr hab. inż. Piotr Kaczmarek

POZNAŃ 2025



## **Zastrzeżenie dotyczące treści pracy dyplomowej**

Niniejsza praca inżynierska zawiera treści, informacje itp. udostępnione przez spółkę Volkswa-  
gen Poznań Sp. z o.o. z siedzibą przy ulicy Warszawskiej 349, 61-060 w Poznaniu, mogące stanowić  
tajemnice przedsiębiorstwa tej spółki i mogące być wykorzystane wyłącznie dla potrzeb napisania  
niniejszej pracy. Wobec powyższego niedozwolone jest wykorzystywanie całości lub części niniejszej  
pracy, a także udostępnianie całości lub części pracy komukolwiek jak również kopiowanie, powie-  
lanie, publikowanie itp. bez pisemnej zgody spółki Volkswagen Poznań Sp. z o.o. – zastrzeżenie to  
nie ma zastosowania do przypadku udostępnienia niniejszej pracy nauczycielom akademickim w  
celu oceny, recenzji i obrony ww. pracy. Podmioty, które naruszą powyższy zakaz ponoszą odpo-  
wiedzialność odszkodowawczą wobec spółki Volkswagen Poznań Sp. z o.o.

Tutaj będzie skan karty pracy dyplomowej.

# Spis treści

<b>1</b>	<b>Wstęp</b>	<b>1</b>
<b>2</b>	<b>Podstawy teoretyczne</b>	<b>3</b>
2.1	Struktura procesu . . . . .	3
2.1.1	Gromadzenie danych dotyczących ofert usługodawców . . . . .	3
2.1.2	Przygotowanie danych . . . . .	3
2.1.3	Przebieg Iteracji . . . . .	4
2.2	Wykorzystane technologie . . . . .	5
2.2.1	Skrypty pakietu Office . . . . .	5
2.2.2	SharePoint . . . . .	5
2.2.3	Power Automate . . . . .	6
2.2.4	Power Apps . . . . .	6
<b>3</b>	<b>Architektura rozwiązania</b>	<b>9</b>
3.1	Założenia projektowe . . . . .	9
3.1.1	Systematyzacja danych . . . . .	9
3.1.2	Archiwizacja danych . . . . .	9
3.1.3	Interfejs przyjazny dla użytkownika . . . . .	10
3.1.4	Użycie pakietu Microsoft 365 . . . . .	10
3.1.5	Optymalizacja . . . . .	10
<b>4</b>	<b>Rozwinięcie</b>	<b>11</b>
<b>5</b>	<b>Zakończenie</b>	<b>12</b>
	<b>Literatura</b>	<b>13</b>
<b>A</b>	<b>Składanie dokumentu w systemie L<sup>A</sup>T<sub>E</sub>X</b>	<b>14</b>
A.1	Struktura dokumentu . . . . .	14
A.2	Akapity i znaki specjalne . . . . .	14
A.3	Wypunktowania . . . . .	14
A.4	Polecenia pakietu <code>ppcreefthesis</code> . . . . .	15
A.5	Rysunki . . . . .	15
A.5.1	Tablice . . . . .	16
A.5.2	Przydatne uwagi . . . . .	16
A.6	Literatura i materiały dodatkowe . . . . .	17

# Rozdział 1

## Wstęp

Współczesny świat biznesu stawia coraz większe wymagania wobec przedsiębiorstw, zarówno w zakresie wydajności procesów, jak i precyzji podejmowanych działań. Powszechne metody zarządzania i przetwarzania danych, oparte są na manualnej pracy z wykorzystaniem mało efektywnych narzędzi oraz wymianie informacji w sposób niustandaryzowany. Stają się one niewystarczające w przypadku rosnącej skali operacji oraz wymagań co do szybkości i niezawodności podejmowanych decyzji. W obliczu tych wyzwań coraz większą rolę odgrywają rozwiązania z zakresu automatyzacji biurowej, które umożliwiają oszczędność czasu i zasobów oraz pozwalają na usprawnienie kluczowych procesów organizacyjnych, minimalizując ryzyko błędów ludzkich.

Jednym z obszarów, w którym automatyzacja znajduje zastosowanie, jest zarządzanie usługami IT i powiązanymi kosztami. W dużych organizacjach o rozbudowanej strukturze, konieczność gromadzenia, analizy oraz weryfikacji danych finansowych stanowi poważne wyzwanie. Dzięki wdrożeniu odpowiednich narzędzi, procesy te mogą być prowadzone w sposób uporządkowany i efektywny, umożliwiając jednocześnie bieżącą kontrolę nad wydatkami oraz lepsze planowanie budżetowe.

Ustandaryzowany i zautomatyzowany przepływ informacji ogranicza ryzyko powielania błędów i pozwala na skrócenie czasu potrzebnego na wykonanie poszczególnych zadań. Dodatkowo, wdrożenie automatyzacji zapewnia większą przejrzystość i ułatwia dostęp do informacji każdemu uczestnikowi procesu.

W dobie intensywnej cyfryzacji przedsiębiorstw oraz dynamicznego rozwoju technologii, automatyzacja biurowa staje się konieczna, aby sprostać wymaganiom współczesnego rynku. Odpowiednio zaprojektowane systemy i narzędzia wspierają nie tylko wydajność operacyjną, ale także strategiczne zarządzanie zasobami, umożliwiając rozwój w innych obszarach swojej działalności.

Celem pracy jest opracowanie aplikacji, usprawniającej proces podejmowania decyzji dotyczących zakupu usług IT<sup>1</sup> na najbliższy rok kalendarzowy. Praca została wykonana z wykorzystaniem *Power Platform* oraz *SharePoint*, które są integralną częścią pakietu *Microsoft 365*. Zdecydowano się na wybór tego rozwiązania, ponieważ pozwala ono na prostą integrację między programami wchodzącymi w skład pakietu. Ponadto, każdy z uczestników procesu ma dostęp do wspomnianych serwisów, co pozwala uniknąć dodatkowych kosztów.

**DOPISAC:**

**Struktura pracy jest następująca. W rozdziale 2 przedstawiono przegląd literatury na temat ...**

---

<sup>1</sup> Usługi IT należy rozumieć jako licencje oraz klucze dostępu do używanych systemów informatycznych.

Rozdział 3 jest poświęcony ... (kilka zdań).

Rozdział 4 zawiera ... (kilka zdań) ... itd.

Rozdział X stanowi podsumowanie pracy.

W przypadku prac inżynierskich zespołowych lub magisterskich 2-osobowych, po tych dwóch w/w akapitach musi w pracy znaleźć się akapit, w którym będzie opisany udział w pracy poszczególnych członków zespołu. Na przykład:

Jan Kowalski w ramach niniejszej pracy wykonał projekt tego i tego, opracował ... Grzegorz Bręczyszczykiewicz wykonał ..., itd.

## Rozdział 2

# Podstawy teoretyczne

### 2.1 Struktura procesu

Przedmiotem omawianego procesu jest podjęcie decyzji dotyczących zakupu usług IT w zakładzie Volkswagen Poznań. Proces ten polega na wielokrotnej wymianie uwag dotyczących wcześniej używanego lub nowego oprogramowania między oddziałem Volkswagena w Poznaniu a zakładem z siedzibą w Wolfsburgu.

W wyniku wymiany zdań zapada decyzja o zakupie lub rezygnacji z wybranego produktu. Procedura, zazwyczaj podzielona na cztery indykacje<sup>1</sup>, rozpoczyna się wraz z początkiem czerwca i trwa do końca roku.

Efektem podejmowanych działań jest nabycie odpowiedniej ilości potrzebnych uprawnień licencyjnych. Przy podejmowaniu decyzji kluczowymi aspektami są:

- liczba użytkowników danego oprogramowania,
- cena zakupu w porównaniu z rokiem poprzednim,
- określenie, czy dana usługa zostanie w pełni wykorzystana biorąc pod uwagę poprzednie kryteria.

Dotychczas analiza i przetwarzanie danych odbywały się przy użyciu arkuszy kalkulacyjnych programu Excel, a wymiana informacji między jednostkami była realizowana za pomocą wiadomości e-mail.

#### 2.1.1 Gromadzenie danych dotyczących ofert usługodawców

Informacje na temat serwisów są zbierane na początku roku, przed rozpoczęciem cyklu procesu. W tym czasie, prowadzone są rozmowy między menadżerami odpowiedzialnymi za dane rozwiązanie (*BSM*, ang. *Business Service Manager*) a firmami świadczącymi usługi, w celu otrzymania zaaktualizowanych wiadomości związanych z ich produktami. Na podstawie danych od usługodawców oraz menadżerów, powstaje arkusz, który jest przekazywany do zakładu w Poznaniu.

#### 2.1.2 Przygotowanie danych

Otrzymany arkusz kalkulacyjny, zawiera tabelę o strukturze kolumn podobnej do tabeli 2.1. Brakuje w nim jednak informacji kluczowych do rozpoczęcia cyklu. Dlatego pierwszym krokiem

---

<sup>1</sup>indykacja - wstępne głosowanie

jest przygotowanie danych przez osobę nadzorującą proces ze strony oddziału w Poznaniu. Jej zadaniem jest manualne przypisanie numeru określającego miejsce powstawania kosztów, wewnętrznie nazywanego *MPK*. Numer ten definiuje konkretną jednostkę należącą do obszaru IT, która decyduje o zakupie danego produktu. Ponadto dodawana jest kolumna, w której znajduje się wyliczona różnica cen między rokiem obecnym a poprzednim, w celu określenia czy koszt wzrósł lub zmalał. Tak przetworzony plik zostaje umieszczony we wspólnej przestrzeni dyskowej, co umożliwia pozostałym uczestnikom procesu przystąpienie do analizy oraz dalszego przetwarzania zawartych w nim informacji.

TABELA 2.1: Nagłówki kolumn z arkusza kalkulacyjnego z roku 2022

Service group	Service main group	Service sub group	Business Service	ID	Business Service Manager	Unit of Measurement	PL70 2022 PLAN EUR w KVA	QTY	PL71 2023 PLAN EUR w KVA	QTY
---------------	--------------------	-------------------	------------------	----	--------------------------	---------------------	--------------------------	-----	--------------------------	-----

### 2.1.3 Przebieg Iteracji

W trakcie trwania iteracji rozpatrywane są kluczowe informacje takie jak:

- **Unit of Measurement** - trzeba jakoś wytłumaczyć o co z tym chodzi,
- decyzja podjęta w roku poprzednim.
- cena oraz ilość użytkowników w roku obecnym,
- cena oraz ilość użytkowników w roku przyszłym,

Po analizie i porównaniu danych z wcześniejszych lat, w arkuszu powstają kolejne kolumny. Ich struktura nie jest określona przez żaden standard, ale zazwyczaj zawierają one:

- Komentarz wewnętrzny,
- Status,
- Komentarz klienta.

*Komentarz wewnętrzny* nie jest wymagany dla każdego serwisu. Jest on zapisywany w celu skonsultowania decyzji ze współpracownikami.

*Status* określa wstępną, wymaganą decyzję (Zaakceptowany/Niezaakceptowany).

*Komentarz klienta* zawiera uzasadnienie podjętej decyzji ze strony Volkswagen Poznań.

Tak uzupełniony arkusz zostaje przekazany pośrednio przez zakład w Wolfsburgu, do zarządu firmy.

Kolejnym etapem jest analiza tych informacji przez wcześniej wymienione podmioty. Ich zadaniem jest konfrontacja podjętej decyzji. Dodawane są kolejne kolumny:

- Komentarz BSM,
- Komentarz K-DES.

*Komentarz BSM* jest to odpowiedź ze strony menadżera usługi.

*Komentarz K-DES* (tutaj by się przydało rozszyfrować co to K-DES z niemieckiego) natomiast jest odpowiedzią międzynarodowego zarządu firmy.



Zaaktualizowany plik powraca do Volkswagen Poznań, rozpoczynając tym samym kolejną iterację procesu.

Jak wcześniej wspomniano, proces składa się zazwyczaj z czterech iteracji. Etapem kończącym cykl jest sporządzenie wymaganych dokumentów oraz faktur.

## 2.2 Wykorzystane technologie

Aby usprawnić przebieg procesu, zabezpieczyć go przed błędami i usystematyzować dane, zdecydowano się na stworzenie aplikacji do jego obsługi. Głównym kryterium przy doborze technologii była powszechna dostępność do powstałego systemu wśród pracowników. Dlatego też zdecydowano się na wykorzystanie komponentów pakietu *Microsoft 365*. Pakiet ten jest bardzo rozbudowany i powszechnie wykorzystywany w firmie Volkswagen. Zawiera on programy pozwalające na stworzenie kompletnego systemu bez konieczności użycia dodatkowych serwisów.

### 2.2.1 Skrypty pakietu Office

Skrypty pakietu Office pozwalają na automatyzację zadań w arkuszach kalkulacyjnych Excel. Jedną z dostępnych opcji jest funkcja *Action Recorder*, która daje możliwość "nagrania" sekwencji kroków wykonanych przez użytkownika, a następnie przekształca je na skrypt wielokrotnego użytku.

Skrypty pakietu Office dysponują również wbudowanym *edytorem kodu* (ang. *Code Editor*), opartym na języku *TypeScript*, będącym odmianą *JavaScript*. Sam edytor, choć stosunkowo ograniczony, umożliwia zastosowanie konstrukcji niedostępnych w Action Recorder, takich jak instrukcje warunkowe czy pętle.

Ponadto, program Excel pozwala na zapis skryptu w skrószycie. Oznacza to, że każdy użytkownik dysponujący dostępem do pliku uzyskuje również możliwość uruchomienia kodu powiązanego ze skrószycem, do którego jest on przypisany.

### 2.2.2 SharePoint

SharePoint to platforma należąca do pakietu Microsoft Microsoft 365, umożliwiająca tworzenie aplikacji webowych, takich jak witryny i strony internetowe. Jej głównym celem jest usprawnienie współpracy zespołowej poprzez dostarczanie narzędzi do publikowania informacji i raportów, które mogą być skierowane do określonych grup odbiorców.

Jednym z kluczowych zastosowań SharePointa jest zarządzanie danymi. Platforma oferuje przestrzeń do przechowywania różnego rodzaju plików, dokumentów i informacji, pełniąc funkcję serwera danych. Dzięki dostępności wbudowanych konektorów<sup>2</sup> (ang. *connectors*), umożliwia również wykorzystanie przechowywanych danych w procesie tworzenia aplikacji czy witryn.

Istotnym elementem środowiska SharePoint jest możliwość tworzenia list, często nazywanych *listami sharepointowymi*. Listy te mogą być wykorzystywane jako proste bazy danych, które umożliwiają dynamiczne aktualizowanie i synchronizowanie danych w czasie rzeczywistym.

SharePoint oferuje zaawansowane zarządzanie uprawnieniami. Administratorzy mogą precyzyjnie definiować dostęp użytkowników do poszczególnych zasobów witryny co pozwala na skuteczne zabezpieczenie wrażliwych informacji.

---

<sup>2</sup> *Konektor* (ang. *connector*) – moduł umożliwiający integrację aplikacji z usługami lub źródłami danych w celu wymiany informacji i synchronizacji systemów.

Platforma jest silnie zintegrowana z innymi usługami pakietu Microsoft 365, takimi jak Teams, Outlook czy OneDrive. Dzięki temu użytkownicy mogą współdzielić dane, pracować nad nimi w czasie rzeczywistym i korzystać z jednego spójnego środowiska pracy.

Ważnym aspektem SharePointa jest możliwość dostosowania wyglądu i funkcjonalności witryn do potrzeb użytkowników. Personalizacja obejmuje m.in. konfigurację interfejsu, dodawanie aplikacji webowych czy tworzenie dedykowanych formularzy.

W kontekście współczesnych modeli pracy, takich jak praca hybrydowa czy zdalna, SharePoint oferuje wsparcie dla użytkowników korzystających z różnorodnych urządzeń. Dostęp do danych jest możliwy za pośrednictwem przeglądarki internetowej oraz aplikacji mobilnych.

### 2.2.3 Power Automate

Power Automate to narzędzie wchodzące w skład pakietu Microsoft 365, które pozwala na automatyzację procesów biznesowych (*RPA*, ang. *Robotic Process Automation*), ograniczając potrzebę wykonywania powtarzalnych czynności. Umożliwia ono tworzenie przepływów pracy nazywanych *flow* (ang. *przepływ*), opartych na zdarzeniach, które umożliwiają spójną pracę różnych środowisk.

Flow w Power Automate jest odpowiednikiem funkcji w standardowych językach programowania, przy czym jest ono tworzone w wizualnym środowisku Low-Code i działa na zasadzie logicznego ciągu akcji wyzwalanych po sobie przez określone instrukcje.

Za pomocą flow można tworzyć własne procesy, które przy odpowiedniej implementacji, dorównują tym znanym z pełnych środowisk kodowych pod względem logiki i efektywności. Do dyspozycji są instrukcje warunkowe, pętle, zmienne, operacje na danych czy integracje z API poprzez konektory.

### 2.2.4 Power Apps

Power Apps to kolejne środowisko Low-Code, wchodzące w skład pakietu Microsoft 365, które jest dedykowanym rozwiązaniem do tworzenia aplikacji biznesowych. Dzięki intuicyjnemu interfejsowi graficznemu daje możliwość prostej implementacji mechanizmu działania nawet przez osoby bez zaawansowanej wiedzy programistycznej. Jest ona zintegrowana z innymi usługami pakietu Microsoft 365, takimi jak SharePoint czy Power Automate, co rozszerza możliwości stworzonych aplikacji.

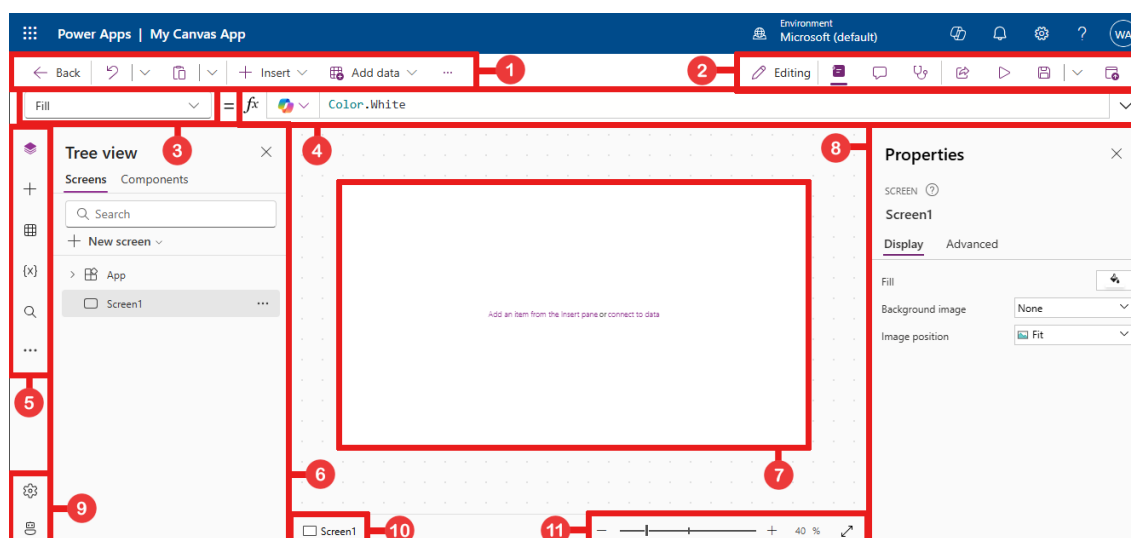
Power Apps pozwala na stworzenie spersonalizowanej aplikacji, dostosowanej do motywu organizacji, a przy połączeniu z innymi serwisami daje możliwość tworzenia zaawansowanych rozwiązań, minimalizując przy tym czas potrzebny na ich zaimplementowanie.

Ekran aplikacji, komponowane za pomocą tego rozwiązania, porównywalne są z tymi, które można stworzyć w standardowych środowiskach programistycznych (jak np. JavaScript czy .NET), jednak proces ich tworzenia jest prostszy, ze względu na obecność edytora wizualnego. Umożliwia on korzystanie z gotowych komponentów w aplikacji, takich jak przyciski, pola danych wejściowych, listy, tabele, grafiki etc.

Dodawanie elementów do ekranów aplikacji odbywa się poprzez przeciąganie ich z biblioteki i upuszczanie w wybranym miejscu. Każdy komponent, może zostać skonfigurowany według potrzeb użytkownika poprzez edycje *właściwości*. Możemy określić między innymi wypełnienie czy pozycję  $X$  i  $Y$  na ekranie, ale niektóre obiekty mają też unikalne właściwości takie jak *OnSelect*<sup>3</sup> dla przycisku.

---

<sup>3</sup>OnSelect – określa akcje, które zostaną wykonane po naciśnięciu elementu



RYSUNEK 2.1: Edytor Power Apps

LINK DO OBRAZKA I OPISU ELEMENTÓW: <https://learn.microsoft.com/en-us/power-apps/maker/canvas-apps/power-apps-studio>

Rysunek 2.1 przedstawia edytor programu. Zawiera on następujące elementy:

1. **Pasek poleceń:** wyświetla inny zestaw poleceń w zależności od wybranego kontrolki.
2. **Akcje aplikacji:** Opcje wyświetlania właściwości, dodawania komentarzy, sprawdzania błędów, udostępniania, podglądu, zapisu lub publikowania aplikacji.
3. **Lista właściwości:** Lista właściwości wybranego obiektu.
4. **Pasek formuł:** Tworzenie lub edycja formuły dla wybranej właściwości z użyciem jednej lub więcej funkcji.
5. **Menu tworzenia aplikacji:** Panel wyboru umożliwiający przełączanie się między źródłami danych oraz wstawianie dodatkowych opcji.
6. **Lista elementów aplikacji:** Pokazuje elementy obecne na ekranie w postaci drzewa.
7. **Płótno/ekran:** Główne płótno do komponowania struktury aplikacji.
8. **Panel właściwości:** Lista właściwości wybranego obiektu.
9. **Ustawienia i wirtualny agent:** Ustawienia aplikacji lub uzyskanie pomocy od wirtualnego agenta.
10. **Selektor ekranu:** Przełączanie się między różnymi ekranami w aplikacji.
11. **Zmiana rozmiaru płótna:** Zmienianie rozmiaru wyświetlanego płótna podczas tworzenia aplikacji.

TABELA 2.2:

	2022	2023	2024
Nazwy kolumn na przestrzeni lat	Service group	Service group	Service group
	Service main group	Service main group	Service main group
	Service sub group	Service sub group	Service sub group
	Business Service	Business Service	Business Service
	ID	ID	ID
	Business Service Manager	Business Service Manager	Business Service Manager
	Unit of Measurement	Unit of Measurement	Resource Unit
		Settlementtype	Settlementtype
	PL71 2023 PLAN EUR w KVA	PL71 2023 PLAN EUR w KVA	PL72 2024 PLAN EUR w KVA
	QTY	QTY	QTY
	PL71 2023 PLAN EUR w KVA	PL72 2024 PLAN EUR w KVA	PL73 2025 PLAN EUR w KVA
	QTY	QTY	QTY

## Rozdział 3

# Architektura rozwiązania

Niniejszy rozdział przedstawia architekturę rozwiązania, na którą składają się założenia projektowe oraz implementacja systemu. Założenia projektowe określają podstawowe wymagania oraz wytyczne, które stanowią fundament opracowywanego rozwiązania. Implementacja natomiast opisuje techniczne aspekty realizacji przyjętych założeń, w tym wykorzystane narzędzia, technologie oraz metodykę pracy.

### 3.1 Założenia projektowe

Założenia projektowe stanowią zbiór wytycznych, które określają funkcjonalność oraz wymagania techniczne, tworzonego rozwiązania.

#### 3.1.1 Systematyzacja danych

Jedną z najważniejszych funkcji, jakie pełni omawiana aplikacja, jest systematyzacja danych. Jak wcześniej wspomniano, arkusze kalkulacyjne przesyłane przez oddział w Wolfsburgu, nie mają ustalonej struktury. Utrudniało to znacznie czytelność, ponieważ użytkownik musiał poświęcić więcej czasu na przeanalizowanie zawartych informacji. Dodatkowo uniemożliwia to stworzenie jednolitej bazy danych a co za tym idzie, wykorzystania ich w systemie automatyzacji biznesowej.

#### 3.1.2 Archiwizacja danych

Utworzenie bazy danych zawierającej dane historyczne stanowi kluczowy element projektowanego systemu, mający na celu zapewnienie ciągłości procesów decyzyjnych oraz wsparcie użytkowników w analizie i podejmowaniu świadomych decyzji.

Archiwizacja danych umożliwia nowym użytkownikom szybkie zapoznanie się z procesem, jego historią oraz podejmowanymi wcześniej decyzjami. Dzięki dostępowi do danych historycznych możliwe jest zrozumienie kontekstu wcześniejszych działań, co znacząco skraca czas potrzebny na wdrożenie się do pracy z systemem.

Zarchiwizowane dane stanowią podstawę do tworzenia raportów dotyczących budżetu oraz kosztów usług IT na przestrzeni lat. Analiza trendów, porównanie wyników oraz identyfikacja obszarów wymagających optymalizacji stają się możliwe dzięki dostępowi do historycznych informacji.

Dane archiwalne pozwalają na obiektywne podejmowanie decyzji, opartych na analizie wcześniejszych wyników. Dzięki temu użytkownicy mogą unikać powtarzania błędów oraz skutecznie przewidywać potencjalne konsekwencje podejmowanych działań.

### 3.1.3 Interfejs przyjazny dla użytkownika

Stworzenie dedykowanego narzędzia z prostym i intuicyjnym interfejsem pozwala na łatwą nawigację po bazie danych oraz umożliwia koncentrację na konkretnej usłudze, minimalizując ryzyko błędów, takich jak pomyłki między wierszami w arkuszu kalkulacyjnym.

### 3.1.4 Użycie pakietu Microsoft 365

Wykorzystanie platformy Power<sup>1</sup> w połączeniu z Sharepoint, pozwala na utworzenie w pełni funkcjonalnego rozwiązania, zachowując spójność danych dzięki integracji poszczególnych składników pakietu.

Aby korzystanie z aplikacji było możliwe, użytkownicy muszą mieć dostęp do potrzebnych usług oraz licencje. W przypadku omawianego pakietu, każdy z pracowników, ma do niego dostęp. Pozwala to na uniknięcie dodatkowych kosztów.

Niestety użyty pakiet, nie jest dostępny w najbardziej rozbudowanym wariantcie. Wprowadza to pewne ograniczenia, ponieważ brakuje w nim oprogramowania do tworzenia i zarządzania rozbudowanymi bazami danych o złożonej strukturze (takie możliwości daje między innymi *Microsoft Azure*). Sharepoint pozwala jedynie na utworzenie prostej bazy danych opierającej się o wcześniej opisane listy. Głównym problemem było ograniczenie związane z brakiem możliwości tworzenia relacji między kilkoma listami, co znacząco utrudniało zarządzanie danymi o złożonej strukturze.

### 3.1.5 Optymalizacja

Priorytetem implementowanego rozwiązania jest optymalizacja procesu. Oprócz oszczędności czasu poprzez wprowadzenie automatyzacji, ważne jest usprawnienie analizy i przetwarzania danych poprzez użytkowników.

---

<sup>1</sup>Platforma Power (ang. *Power Platform*) – Składowa pakietu Microsoft 365. Zawiera ona takie programy jak Power Apps, Power Automate czy Power BI.

## Rozdział 4

# Rozwinięcie

Rozdziały dokumentujące pracę własną studenta: opisujące ideę, sposób lub metodę rozwiązania postawionego problemu oraz rozdziały opisujące techniczną stronę rozwiązania — dokumentacja techniczna, przeprowadzone testy, badania i uzyskane wyniki.

Praca musi zawierać elementy pracy własnej autora adekwatne do jego wiedzy praktycznej uzyskanej w okresie studiów. Za pracę własną autora można uznać np.: stworzenie aplikacji informatycznej lub jej fragmentu, zaproponowanie algorytmu rozwiązania problemu szczegółowego, przedstawienie projektu np. systemu informatycznego lub sieci komputerowej, analizę i ocenę nowych technologii lub rozwiązań informatycznych wykorzystywanych w przedsiębiorstwach, itp.

Autor powinien zadbać o właściwą dokumentację pracy własnej obejmującą specyfikację założeń i sposób realizacji poszczególnych zadań wraz z ich oceną i opisem napotkanych problemów. W przypadku prac o charakterze projektowo-implementacyjnym, ta część pracy jest zastępowana dokumentacją techniczną i użytkową systemu.

W pracy **nie należy zamieszczać całego kodu źródłowego** opracowanych programów. Kod źródłowy napisanych programów, wszelkie oprogramowanie wytworzone i wykorzystane w pracy, wyniki przeprowadzonych eksperymentów powinny być przekazane promotorowi oraz wgrane wraz z pracą do systemu informatycznego uczelni.

## Styl tekstu

Należy<sup>1</sup> [1] stosować formę bezosobową, tj. *w pracy rozważono .....*, *w ramach pracy zaprojektowano .....*, a nie: *w pracy rozważyłem*, *w ramach pracy zaprojektowałem*. Odwołania do wcześniejszych fragmentów tekstu powinny mieć następującą postać: „Jak wspomniano wcześniej, ....”, „Jak wykazano powyżej ....”. Należy unikać długich zdań.

Niedopuszczalne są zwroty używane w języku potocznym. W pracy należy używać terminologii technicznej, która ma sprecyzowaną treść i znaczenie.

Niedopuszczalne jest pisanie pracy metodą *copy-paste*, bo jest to plagiat i dowód intelektualnej indolencji autora. Dane zagadnienie należy opisać własnymi słowami. Zawsze trzeba powołać się na zewnętrzne źródła.

---

<sup>1</sup>Uwagi o stylu pochodzą częściowo ze stron prof. Macieja Drozdowskiego.

## Rozdział 5

# Zakończenie

Zakończenie pracy zwane również Uwagami końcowymi lub Podsumowaniem powinno zawierać ustosunkowanie się autora do zadań wskazanych we wstępie do pracy, a w szczególności do celu i zakresu pracy oraz porównanie ich z faktycznymi wynikami pracy. Podejście takie umożliwia jasne określenie stopnia realizacji założonych celów oraz zwrócenie uwagi na wyniki osiągnięte przez autora w ramach jego samodzielnej pracy.

Integralną częścią pracy są również dodatki, aneksy i załączniki zawierające stworzone w ramach pracy programy, aplikacje i projekty.



# Literatura

- [1] Maciej Drozdowski. Jak pisać prace dyplomowe – uwagi o formie. [on-line]  
[http://www.cs.put.poznan.pl/mdrozdowski/dyd/txt/jak\\_mgr.html](http://www.cs.put.poznan.pl/mdrozdowski/dyd/txt/jak_mgr.html), 2006.
- [2] Donald E. Knuth. *The  $T_E X$ book*. Computers and Typesetting. Addison-Wesley, Reading, MA, USA, 1986.
- [3] Leslie Lamport.  *$\LaTeX$  — A Document Preparation System — User’s Guide and Reference Manual*. Addison-Wesley, Reading, MA, USA, 1985.

## Dodatek A

# Składanie dokumentu w systemie L<sup>A</sup>T<sub>E</sub>X

W tym rozdziale znajduje się garść informacji o tym, jak poprawnie składać tekst pracy w systemie L<sup>A</sup>T<sub>E</sub>X wraz z przykładami, które mają służyć do przeklejania do własnych dokumentów.

### A.1 Struktura dokumentu

Praca składa się z rozdziałów (`chapter`) i podrozdziałów (`section`). Ewentualnie można również rozdziały zagnieżdzać (`subsection`, `subsubsection`), jednak nie powinno się wykraczać poza drugi poziom hierarchii (czyli `subsubsection`).

### A.2 Akapity i znaki specjalne

Akapity rozdziela się od siebie przynajmniej jedną pustą linią. Podstawowe instrukcje, które się przydają to *wyróżnienie pewnych słów*. Można również stosować **styl pogrubiony**, choć nie jest to generalnie zalecane.

Należy pamiętać o zasadach polskiej interpunkcji i ortografii. Po spójnikach jednoliterowych warto wstawić znak tyldy (`~`), który jest tak zwaną „twardą spacją” i powoduje, że wyrazy nią połączone nie będą rozdzielane na dwie linie tekstu.

Polskie znaki interpunkcyjne różnią się nieco od angielskich: to jest „polski”, a to jest “angielski”. W kodzie źródłowym tego tekstu będzie widać różnicę.

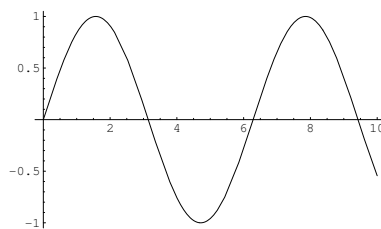
Proszę również zwrócić uwagę na znak myślnika, który może być pauzą „—” lub półpauzą: „-”. Należy stosować je konsekwentnie. Do łączenia wyrazów używamy zwykłego „-” (*północno-wschodni*), do myślników — pauzy lub półpauzy. Inne zasady interpunkcji i typografii można znaleźć w słownikach.

### A.3 Wypunktowania

Wypunktowanie z cyframi:

1. to jest punkt,
2. i to jest punkt,
3. a to jest ostatni punkt.

Po wypunktowaniach czasem nie warto wstawiać wcięcia akapitowego. Wtedy przydatne jest polecenie `noindent`. Wypunktowanie z kropkami (tzw. *bullet list*) wygląda tak:



RYSUNEK A.1: Wykres.

- to jest punkt,
- i to jest punkt,
- a to jest ostatni punkt.

Wypunktowania opisowe właściwie niewiele się różnią:

**elementA** to jest opis,

**elementB** i to jest opis,

**elementC** a to jest ostatni opis.

## A.4 Polecenia pakietu *ppcreefthesis*

Parę poleceń zostało zdefiniowanych aby uspoźnić styl pracy. Są one przedstawione poniżej (oczywiście nie trzeba się do nich stosować).

**Makra zdefiniowane dla języka angielskiego.** Są nimi: `termdef` oraz `acronym`. Przykłady poniżej obrazują ich przewidywane użycie w tekście.

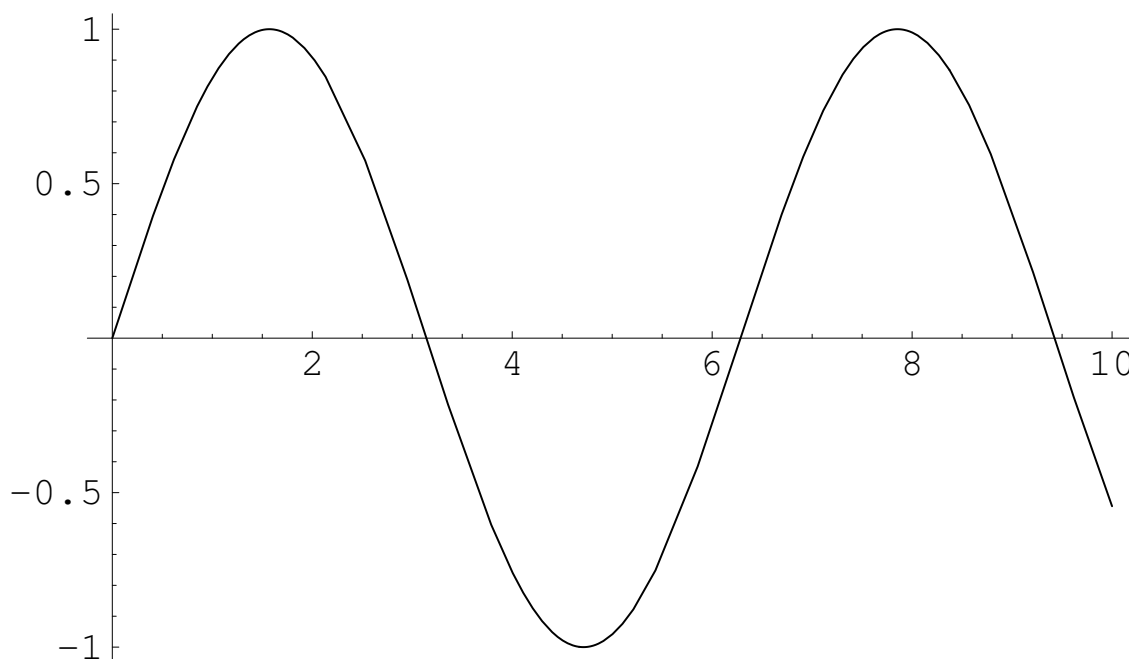
źródło	<code>we call this a \termdef{Database Management System} (\acronym{DBMS})</code>
docelowo	we call this a <i>Database Management System (DBMS)</i>

**Makra zdefiniowane dla języka polskiego.** Podobnie jak dla języka angielskiego zdefiniowano odpowiedniki polskie: `definicja`, `akronim` oraz `english` dla tłumaczeń angielskich terminów. Przykłady poniżej obrazują ich przewidywane użycie w tekście.

źródło	<code>nazywamy go \definicja{systemem zarządzania bazą danych} (\akronim{DBMS}, \english{Database Management System})</code>
docelowo	nazywamy go <i>systemem zarządzania bazą danych (DBMS, ang. Database Management System)</i>

## A.5 Rysunki

Wszystkie rysunki (w tym również diagramy, szkice i inne) osadzamy w środowisku `figure` i umieszczamy podpis *pod* rysunkiem, w formie elementu `caption`. Rysunki powinny zostać umieszczone u góry strony (osadzone bezpośrednio w treści strony zwykle utrudniają czytanie tekstu). Rysunek A.1 zawiera przykład pełnego osadzenia rysunku na stronie.



RYSUNEK A.2: Ten sam wykres ale na szerokość tekstu.

### A.5.1 Tablice

Tablice to piękna rzecz, choć akurat ich umiejętne tworzenie w  $\text{\LaTeX}$ u nie jest łatwe. Jeśli tablica jest skomplikowana, to można ją na przykład wykonać w programie OpenOffice, a następnie wyeksportować jako plik *PDF*. W każdym przypadku tablice wstawia się podobnie jak rysunki, tylko że w środowisko `table`. Tradycja typograficzna sugeruje umieszczenie opisu tablicy, a więc elementu `caption` ponad jej treścią (inaczej niż przy rysunkach).

Tablica A.1 pokazuje pełen przykład.

TABELA A.1: Przykładowa tabela. Styl opisu jest zgodny z rysunkami.

artykuł	cena [zł]
bułka	0,4
masło	2,5

### A.5.2 Przydatne uwagi

- Znakiem myślnika jest w  $\text{\LaTeX}$ u dywiz pełen (`—`) albo półpauza (`-`), przykład: A niech to jasna cholera `—` wrzasnąłem.
- Połączenie między wyrazami to zwykły myślnik, przykład: północno-zachodni
- Sprawdź ostrzeżenia o `'overfull'` i `'underful'` boxes. Niektóre z nich można zignorować (spójrz na wynik formatowania), niektóre trzeba poprawić; czasem przeformułować zdanie.
- Przypisy stawia się wewnątrz zdań lub za kropką, przykład: Footnote is added after a comma.<sup>1</sup>

---

<sup>1</sup>Here is a footnote.

- Nie używaj przypisów zbyt często. Zobacz, czy nie lepiej będzie zintegrować przypis z tekstem.
- Tytuły tabel, rysunków powinny kończyć się kropką.
- Nie używaj modyfikatora [h] (here) do rysunków i tabel. Rysunki i tabele powinny być justowane do góry strony lub na stronie osobnej.
- Wyróżnienie w tekście to polecenie *wyraz*, należy unikać **czcionki pogrubionej** i podkreślenia (które wystają wizualnie z tekstu i rozpraszaają).
- Nazwy plików, katalogów, ścieżek, zmiennych środowiskowych, klas i metod formatujemy poleceniem `plik.o_pewnej_nazwie`.
- Po ostatniej zmianie do treści, sprawdź i przenieś wiszące spójniki wstawiając przed nie znak tyldy (twardej spacji), przykład: Ala i kotek nie lubią mleczka, a Stasiu lubi.
- Za i.e. (id est) i e.g. (exempli gratia) stawia się zwyczajowo przecinek w typografii amerykańskiej.
- Przed i za pełną pauza nie ma zwyczajowo spacji w typografii amerykańskiej, przykład: Darn, this looks good—said Mary.
- Zamykający cudzysłów oraz footnote wychodzą za ostatni znak interpunkcji w typografii amerykańskiej, przykłady: It can be called a “curiosity,” but it’s actually normal. Footnote is added after a comma.<sup>2</sup>
- Odwołania do tabel i rysunków zawsze z wielkiej litery, przykład: In Figure A.1 we illustrated XXX and in Table A.1 we show detailed data.

## A.6 Literatura i materiały dodatkowe

Materiałów jest mnóstwo. Oto parę z nich:

- *The Not So Short Introduction...*, która posiada również tłumaczenie w języku polskim.  
<http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf>
- Klasy stylu `memoir` posiadają bardzo wiele informacji o składzie tekstów anglosaskich oraz sposoby dostosowania  $\text{\LaTeX}$ a do własnych potrzeb.  
<http://www.ctan.org/tex-archive/macros/latex/contrib/memoir/memman.pdf>
- Nasza grupa dyskusyjna i repozytorium Git są również dobrym miejscem aby zapytać (lub sprawdzić czy pytanie nie zostało już zadane).  
<https://github.com/politechnika/put-latex>
- Dla łaknących więcej wiedzy o systemie  $\text{\LaTeX}$  podstawowym źródłem informacji jest książka Lamport’a [3]. Prawdziwy *hardcore* to oczywiście *The  $\text{\TeX}$ book* profesora Knutha [2].

---

<sup>2</sup>Here is a footnote.



© 2025 Remigiusz Wolniak, Michał Gajdzis

Instytut Robotyki i Inteligencji Maszynowej

Wydział Automatyki, Robotyki i Elektrotechniki Politechnika Poznańska

Skład przy użyciu systemu  $\text{\LaTeX}$  na platformie Overleaf.