


# Projektowanie Graficznych Interfejsów Użytkownika

Robert Szmurło

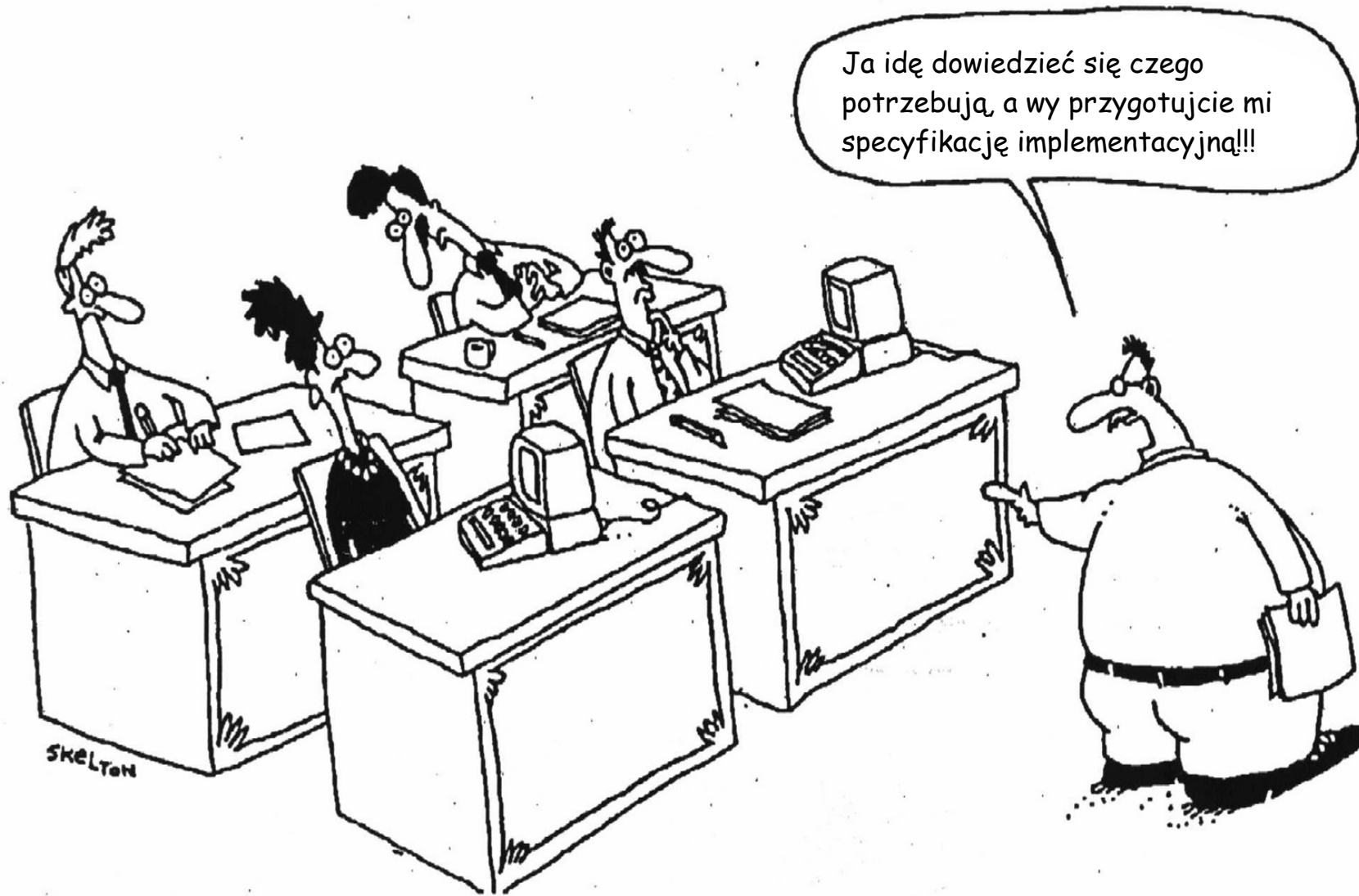


A top-down view of various electronic components and tools scattered on a light-colored wooden surface. The items include a purple motherboard, a yellow coiled cable, a black hard drive, two green RAM modules, a small black fan, a red circuit board, a blue circuit board, a blue screwdriver, a red multimeter, a pair of blue-handled pliers, and several pieces of tape and ribbon cables.

# **Projektowanie Interfejsu w Oparciu o Komponenty Użytkownika.**

**Inżynieria oprogramowania, ale z perspektywy tworzenia interfejsów użytkownika.**

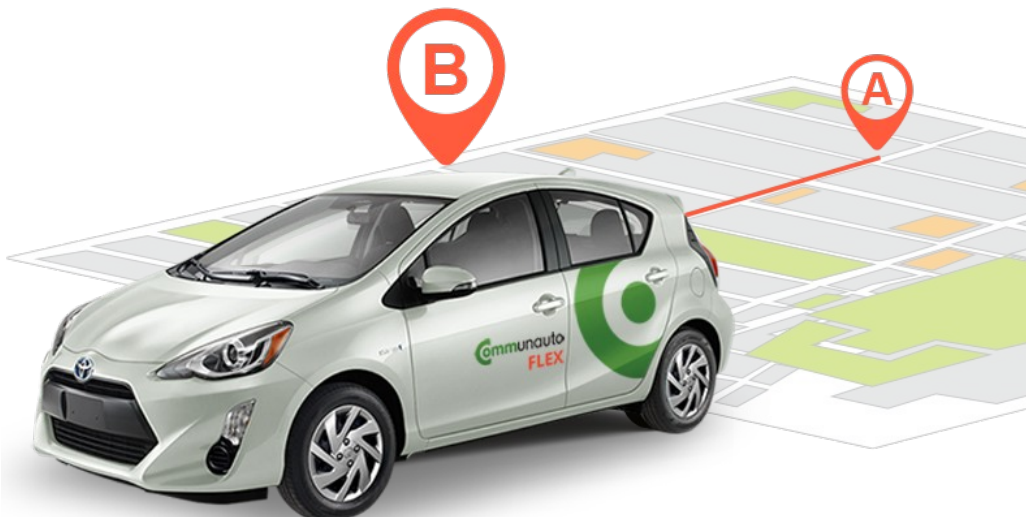
# Tradycyjny „proces” wytwarzania oprogramowania



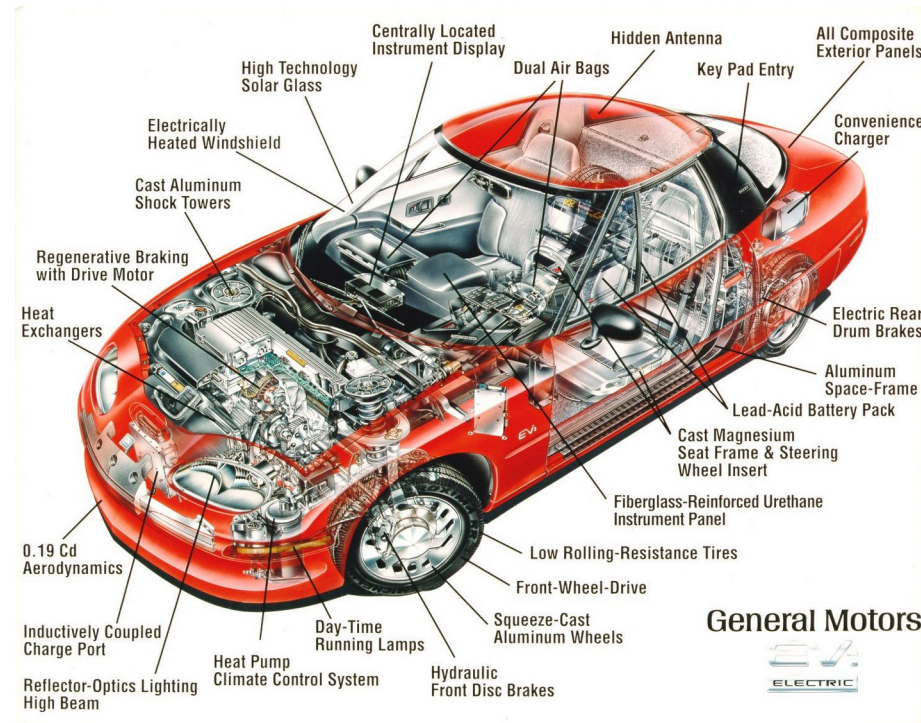


# Model pojęciowy samochodu

## 1) Model oczami użytkownika



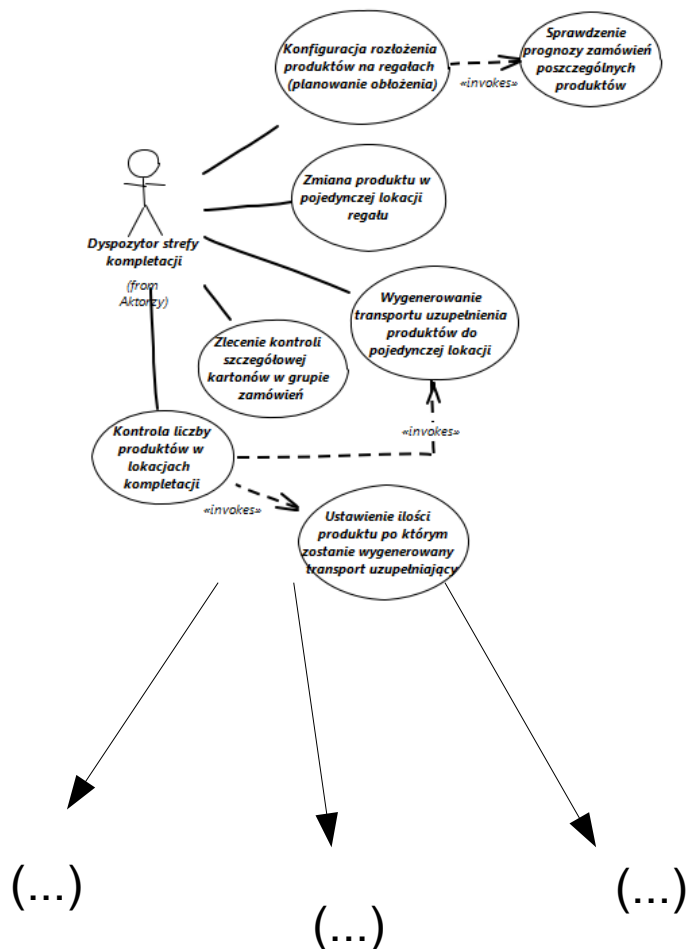
## 2) Model oczami mechanika



# Model pojęciowy a proces wytwarzania: UCD vs MCD

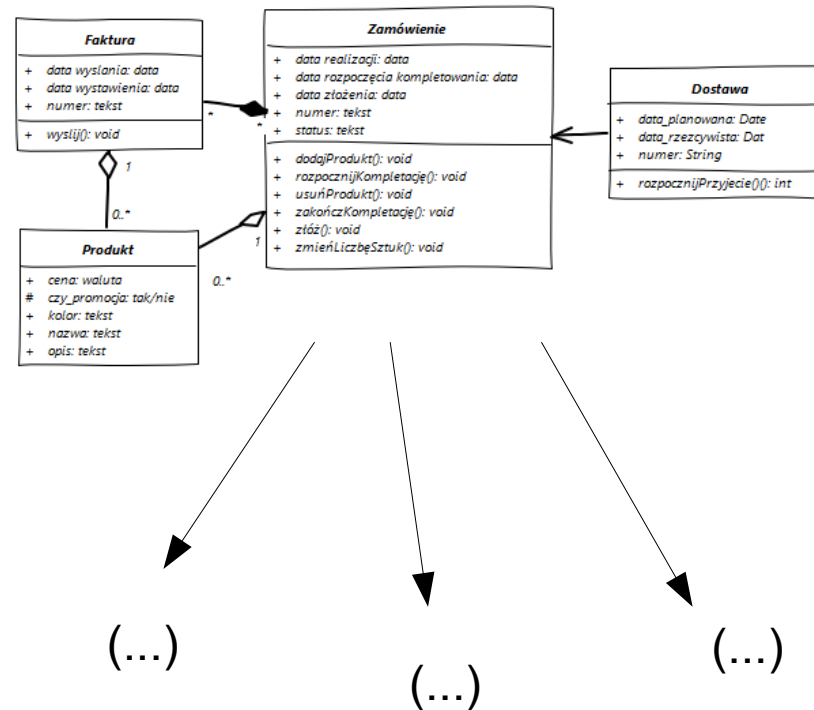
## 1) Pojęcie: **User Centered Design**

– Projekt Skoncentrowany na Użytkowniku (punktem wyjściowym jest funkcja potrzebna do realizacji zadania)



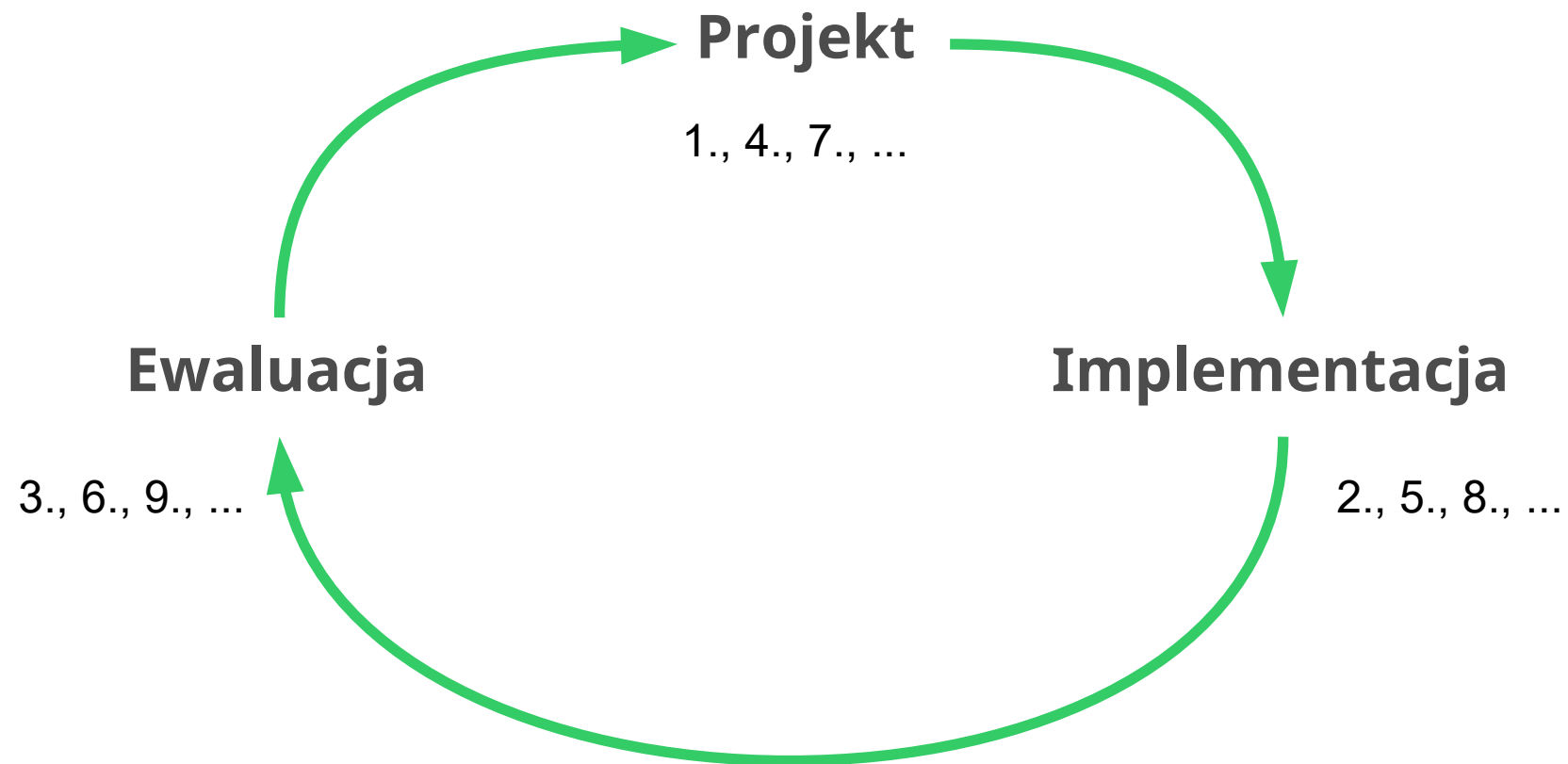
## 2) Pojęcie: **Model Centered Design**

– Projekt Oparty na Modelach (danych dziedzinowych, modelach systemu, strukturze bazy danych, itp..)



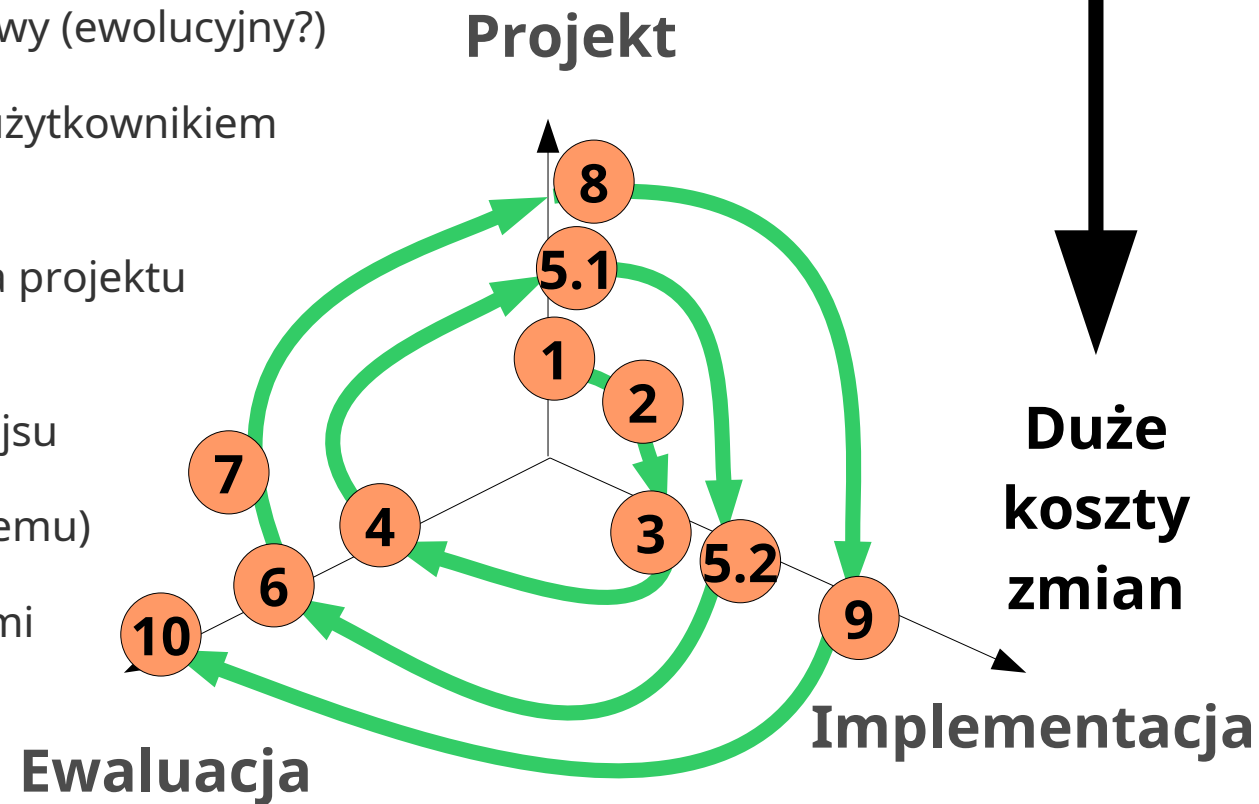
# Spiralny proces wytwarzania oprogramowania

Aby po raz n-ty nie przedstawiać modelu Kaskadowego, od razu przejdźmy do ewolucyjnego, jakim jest Spiralny proces...



# „Spiralny” User Centered Design (UCD)

- 1 - Analiza projektu + użytkowników + ich zadań
- 2 - Szkice projektu interfejsu („na brudno”)
- 3 - „Papierowy” prototyp (model prototypowania z porzuceniem)
- 4 - Testy wewnętrzne - konsultacje z użytkownikiem (zamawiającym system)
- 5.1 5.2 - Korekta + Prototyp komputerowy (ewolucyjny?)
- 6 - Ewaluacja heurystyczna + testy z użytkownikiem („bal masek”)
- 7 - Analiza raportu ewaluacji i korekta projektu
- 8 - Projekt architektury, projekt szczegółowy implementacji interfejsu
- 9 - Implementacja danej iteracji (systemu)
- 10 - Testy Użyteczności z użytkownikami



# Dokumentacja...

- Powstaje na wszystkich etapach projektu...



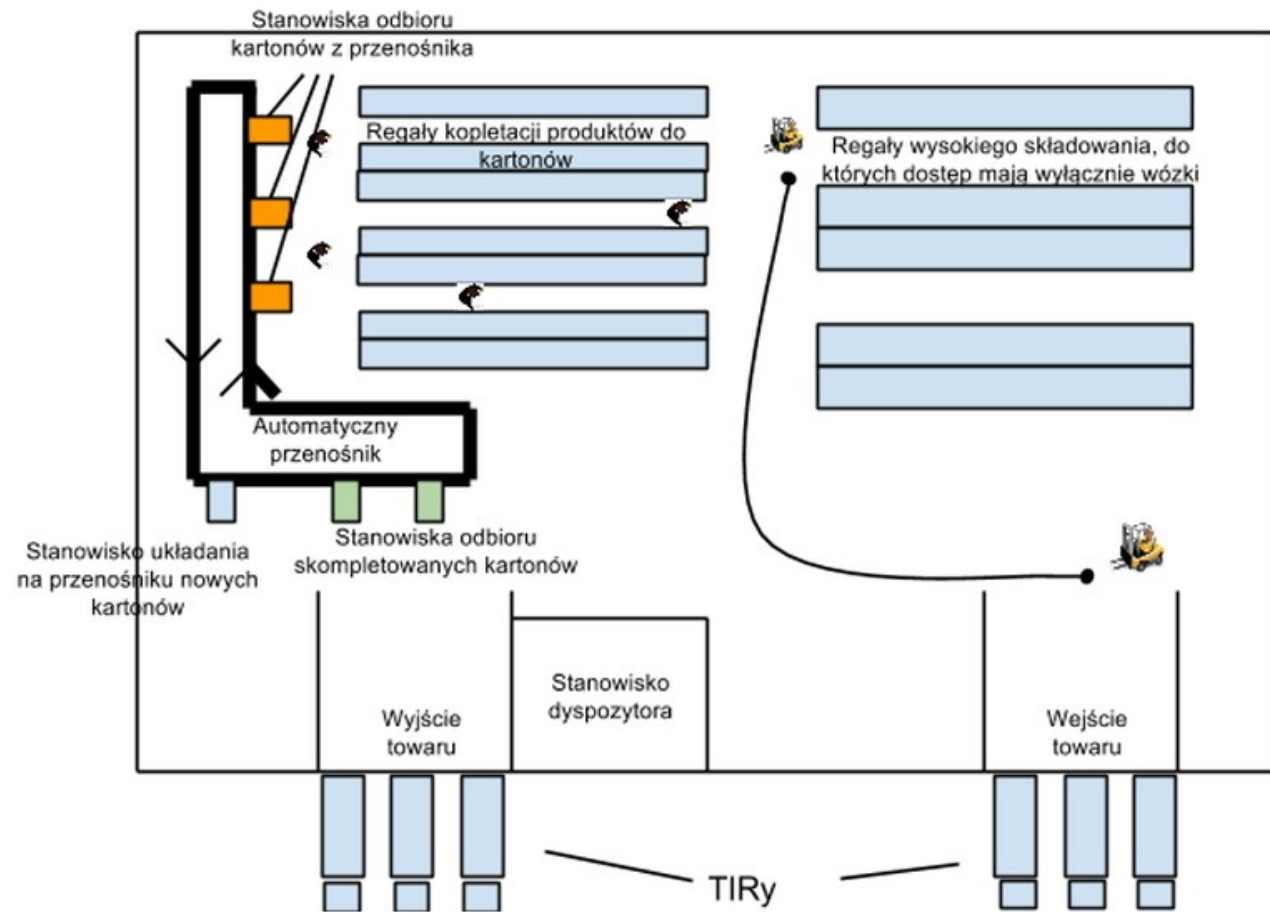
DOKUMENTACJA?? Naprawdę powstaje? Może chociaż notatki po sprintach...





# Przykład dziedzinowy omawiany na wykładzie

## Model centrum logistycznego



# Projekt systemu vs Projekt Interfejsu

Procedurę projektowania podzielimy na 5 głównych etapów:

- A) Analiza wymagań (analiza (1)) – wyodrębnienie podstawowych koncepcji systemu jak np. obiektów, relacji i uogólnień, a z punktu widzenia interfejsu zadań realizowanych przez aktorów w formie listy przypadków użycia. (**Specyfikacja biznesowa (dziedzinowa) - BRD**)
- B) Projekt funkcjonalny (semantyczny (2)) – zdefiniowanie scenariuszy przypadków użycia oraz szkieletów ekranów (okna, widoki) i operacji na nich wykonywanych; specyfikacja danych wejściowych i wyjściowych. (**Specyfikacja funkcjonalna - FRD**)
- C) Projekt sekwencyjny (syntaktyczny (2)) – definiuje kolejność operacji wprowadzania danych wejściowych i wyjściowych (np. przez użytkownika do systemu) na konkretnych ekranach interfejsu. Cykl życia i interakcji poszczególnych widoków i okien systemu. (**Specyfikacja implementacyjna - IRD**)
- D) Projekt wizualny (3)– definiuje wizualną reprezentację obiektów za pomocą widoków, projekt okien, kolejność ekranów - „storyboards”, na początku realizowany w postaci szkieletu ekranu, potem uwzględnia kolorystykę i grafikę. (**Specyfikacja funkcjonalna / implementacyjna - IRD**)
- E) Projekt leksykalny (8) – za pomocą jakich kontrolerek graficznych i technologii, frameworków, modeli wdrożenia będą zrealizowane poszczególne elementy interfejsu.



# Liczba Millera a diagramy UML

[https://pl.wikipedia.org/wiki/Liczba\\_Millera](https://pl.wikipedia.org/wiki/Liczba_Millera)

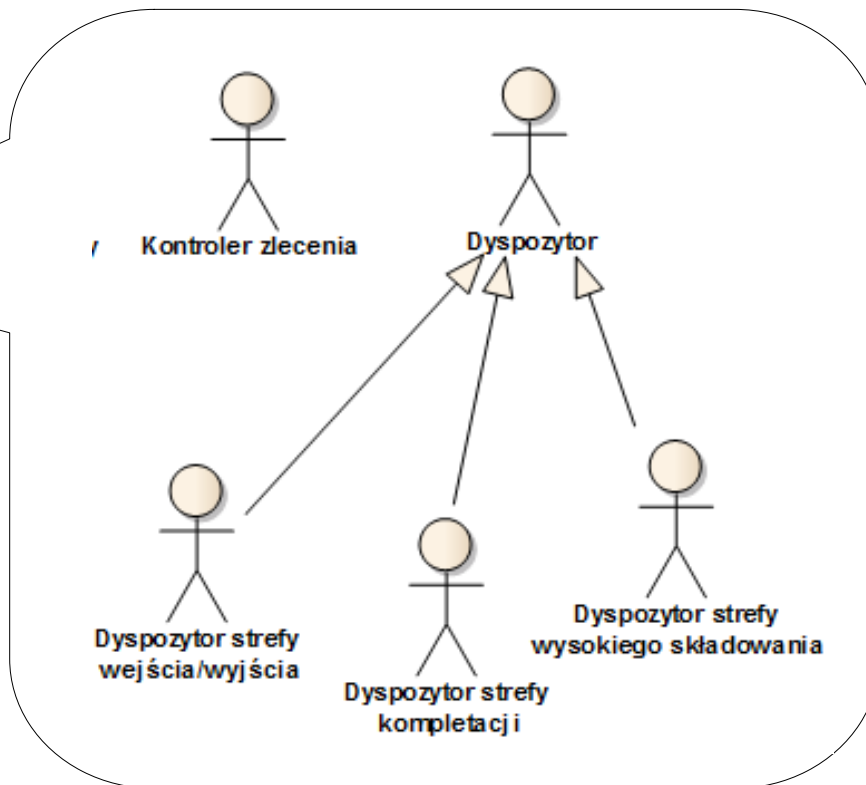
- Magiczne 7 (liczba Millera) – maksymalna liczba informacji jaką człowiek jest w stanie rozróżnić bezpośrednio.
- Liczba ta ustalona została przez George'a Millera w roku 1956 na podstawie badań psychologicznych nad zapamiętywaniem informacji. Liczba ta waha się w zależności od rodzaju informacji do zapamiętania (dźwięk, smak, obraz, liczba), ale zawsze jest to około 7.
- Ze względu na tę różnorodność przyjmuje się rozrzut między 5 a 9, czyli  $7 \pm 2$ .
- Liczba Millera znalazła zastosowanie wszędzie tam, gdzie prezentuje się informacje człowiekowi. Gdy mamy do przekazania użytkownikowi wiele pozycji (menu, produkty etc.), lepiej jest podzielić je na kategorie, tak aby miał ich do wyboru 5-9. Jeśli ten podział nie wystarcza, można dokonać kolejnego podziału na podkategorie – zgodnie z badaniami Millera takich podziałów można dokonać do siódmego poziomu "w głąb" – to z kolei jest ograniczenie pamięci krótkotrwałej.



# Analiza wymagań (A)

Etap Odkrywania / Analizy:

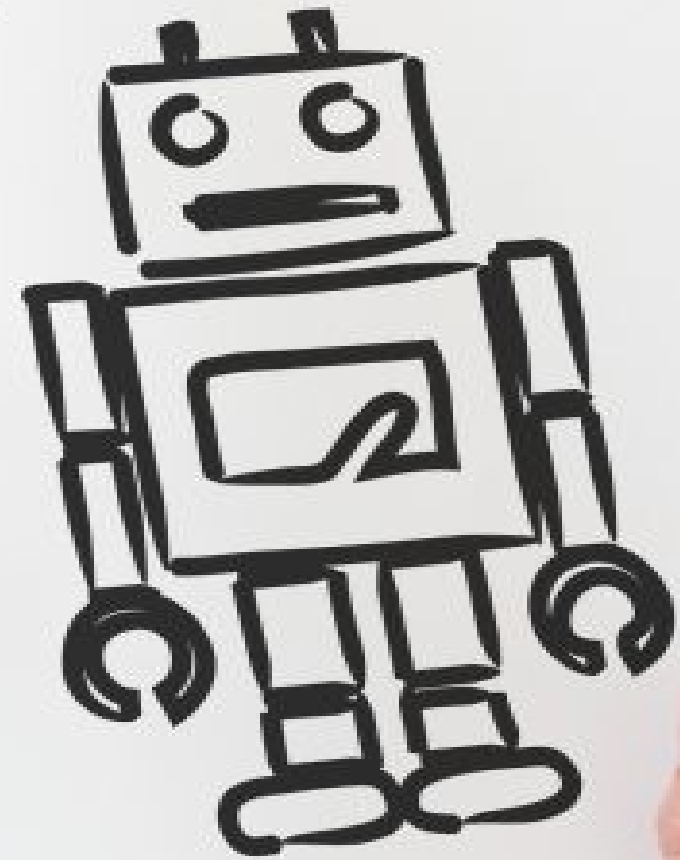
- **1) określ klasy użytkowników (aktorów)**
- 2) określ zadania użytkowników (zadania, oczekiwania)
- 3) scenariusze wykonywania zadań
- 4) zbuduj projekt przypadków użycia





## Poznaj Użytkowników

Określ charakterystykę docelowej grupy użytkowników, która będzie wpływała na wymagania: poziom wykształcenia, możliwości fizyczne (np.: niepełnosprawni), ogólne doświadczenie komputerowe, doświadczenie z podobnymi aplikacjami, umiejętności (szybkie pisanie, czytanie), doświadczenie związane z dziedziną pracy, wymagania dotyczące szkolenia (np.: czy duża rotacja pracowników w firmie)

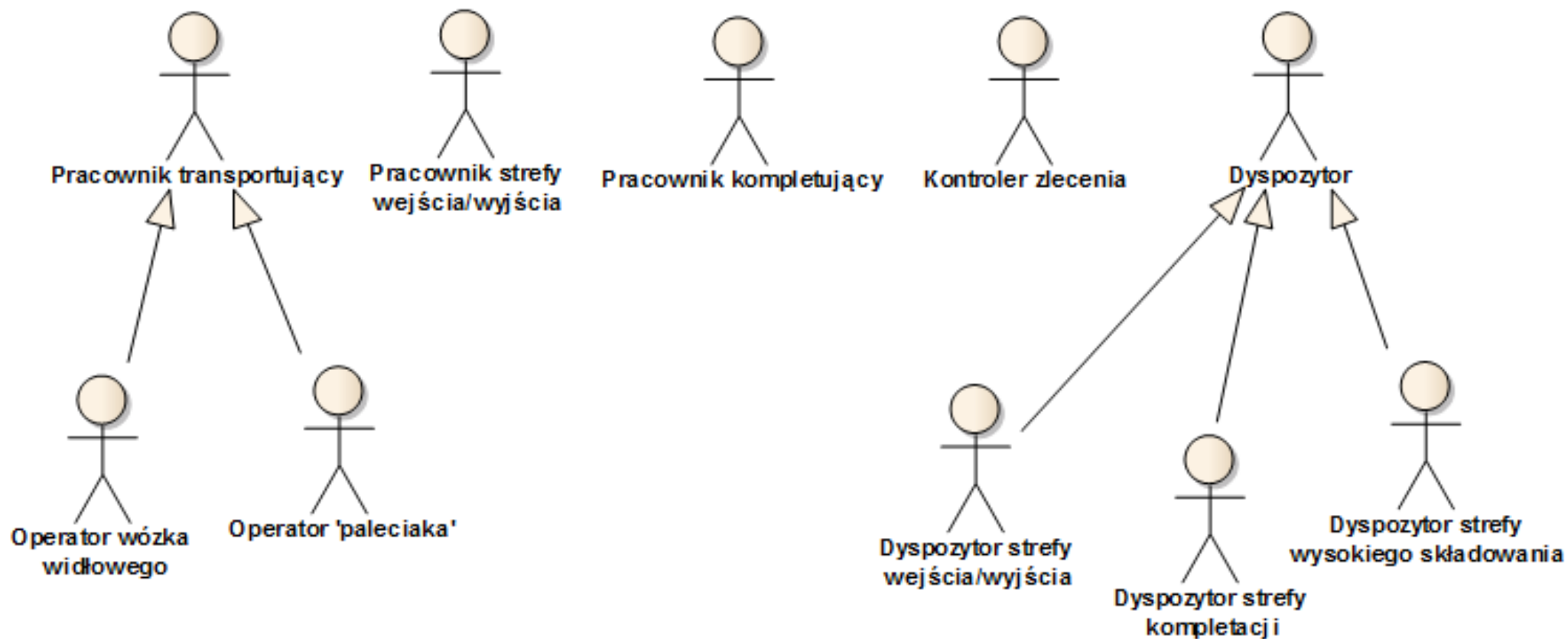


# Poznaj Użytkowników (A)

Kluczowe w UCD, ponieważ koncentrujemy się na użytkownikach

Aktorzy – to role rzeczywistych osób w organizacji

Przykład:



- **Obserwacja i wywiady** – najprostsze i jedne z najbardziej efektywnych (zanim zaczniesz obserwować przygotuj sobie pytania na które chcesz uzyskać odpowiedź)
- **Diagramy pokrewieństwa** (sortowanie kart) – celem jest podział namoduły, zbudowanie struktury aplikacji
  - Burza mózgów – określenie pomysłów (pojęć) związanych z systemem przez użytkowników
  - Grupowanie pomysłów
  - Tworzenie podgrup
  - Nadawanie nazw grupom i podgrupom
- **Test drzewa** (odwrotne sortowanie kart) – zadaniem osób uczestniczących jest odnalezienie elementów w strukturze, cel badania: wskazanie problemów ze znajdowaniem
- Limit pamięci roboczej – ludzka pamięć robocza pozwala na zapamiętanie co najwyżej 7 (lub nawet 4) elementów (Liczba Millera ->)



- Kluczem jest opracowanie odpowiednich pytań.
  - Czego chcemy się dowiedzieć
  - Poszukiwanie osób do badania
  - Określanie pytań
    - Pytania otwarte, np.: Jak często korzystałeś z wsparcia użytkowników?
    - Pytania wielokrotnego wyboru, np.: Ile razy korzystałeś z naszej aplikacji w zeszłym tygodniu:
      - Więcej niż 6 razy.
      - Między 2 a 6.
      - Raz.
      - Ani razu.
  - Humanizacja formularzy
    - postarajmy się aby użytkownik poczuł się elementem procesu (wyjaśnienie celów formularza)
    - skoncentrujmy się na celach
    - Używajmy 'przyjacielskiego' tonu





# Dokument osoby

- Służy do klasyfikowania charakterystyk grup użytkowników

## Kevin Bailey

**AGE** 53  
**OCCUPATION** Entrepreneur, Business Owner  
**LOCATION** On site & Remote  
**TIER** Daily use - Overview  
**ARCHETYPE** Leader

Decisive Value Orientated



"Up to date data is imperative for effective decision making."

### Motivations

Incentive	100%
Achievement	100%
Growth	80%
Power	60%
Social	40%

### Goals

- To receive a product of high standard, willing to pay for it
- Needs to know whether he is receiving value for money
- Would like to have 27/4 overview of current projects and ow resources are distributed

### Frustrations

- Not being able to get a clear answer
- Not being able to access up to date data no matter where he is located
- Not knowing factors why a product or service he is funding is not progressing

### Bio

A self made business person who knows how important time is to a business. Have different interests in multiple ventures. Believes that technology assist in growth.

### Personality

Extrovert	Introvert
Sensing	Intuition
Thinking	Feeling
Judging	Perceiving

### Technology

IT & Internet	80%
Software	70%
Mobile Apps	60%
Social Networks	50%

### Brands

Brands which inspire



# Dokument persony - szablon

- Np.: <https://www.behance.net/gallery/16378203/Our-User-Persona-Template>

**PERSONA TEMPLATE**

**NAME**

**AGE** \_\_\_\_\_

**OCCUPATION** \_\_\_\_\_

**STATUS** \_\_\_\_\_

**LOCATION** \_\_\_\_\_

**TIER** \_\_\_\_\_

**ARCHETYPE** \_\_\_\_\_

**MOTIVATIONS** (Shade bar to desired level)

Incentive \_\_\_\_\_

Fear \_\_\_\_\_

Acheivement \_\_\_\_\_

Growth \_\_\_\_\_

Power \_\_\_\_\_

Social \_\_\_\_\_

**PERSONALITY**

Extrovert \_\_\_\_\_ Introvert \_\_\_\_\_

**GOALS** (The objectives this person hopes to achieve)

- \_\_\_\_\_
- \_\_\_\_\_
- \_\_\_\_\_

**FRUSTRATIONS** (The pain points they'd like to avoid)

- \_\_\_\_\_
- \_\_\_\_\_
- \_\_\_\_\_

**BIO**

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

**IMAGE**

**QUOTE**

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

**FAKE CROW**

**PERSONA TEMPLATE**

**FIRST LAST**

**AGE** 1-100

**OCCUPATION** What they do for a living

**STATUS** Single or Married

**LOCATION** Where they live/work

**TIER** Enthusiast/Prosumer/Pro

**ARCHETYPE** Character model

Adjective 1 Adjective 2 Adjective 3

**MOTIVATIONS**

Incentive \_\_\_\_\_

Fear \_\_\_\_\_

Acheivement \_\_\_\_\_

Growth \_\_\_\_\_

Power \_\_\_\_\_

Social \_\_\_\_\_

**GOALS** (The objectives this person hopes to achieve)

- Goal 1
- Goal 2
- Goal 3

**FRUSTRATIONS** (The pain points they'd like to avoid)

- Frustration 1
- Frustration 2
- Frustration 3

**BIO**

A paragraph to describe the user journey. Should include some background leading up to a current use case.

**IMAGE**

(235 x 205px)

**TECHNOLOGY**

IT and Internet \_\_\_\_\_

Software \_\_\_\_\_

Mobile Apps \_\_\_\_\_

Social Networks \_\_\_\_\_

**BRANDS**

(A collection of greyscale brand logos representing the users favorite brands)

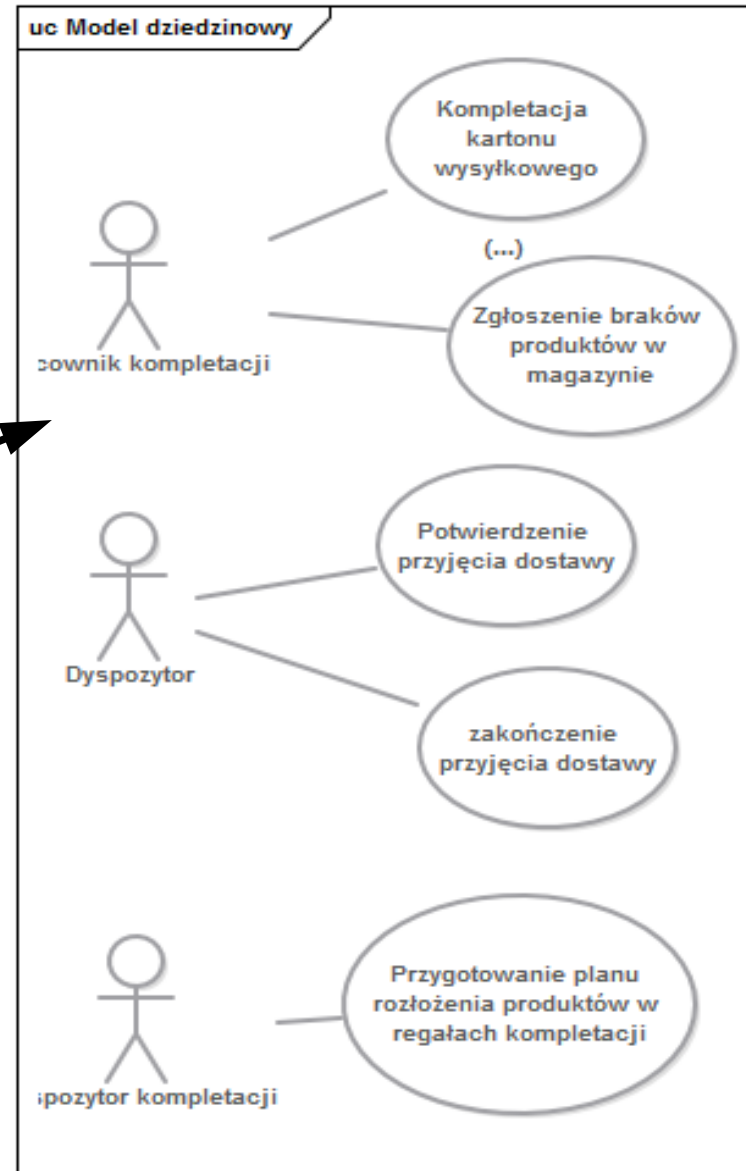
*"Personal quotation"*  
(Should capture the essence of this persona's personality)



# Projekt pojęciowy (A)

## Etap Odkrywania / Analizy:

- 1) określ klasy użytkowników (aktorów)
- 2) określ zadania użytkowników (co musi być zrobione, wymagania, zazwyczaj zakres obowiązków w firmie, procesy w których uczestniczą)
- 3) **Podróż użytkownika** (ang. *user journey*) scenariusze wykonywania zadań
- 4) zbuduj projekt przypadków użycia



# Zadania / procesy użytkowników (A)

## Zadanie związane z osiągnięciem pewnego wyniku! -

Każde zadanie powinno kończyć się osiągnięciem ściśle określonego **rezultatu** obserwowalnego z punktu widzenia organizacji. (Jest to podstawowe kryterium sukcesu w testach akceptacyjnych.)

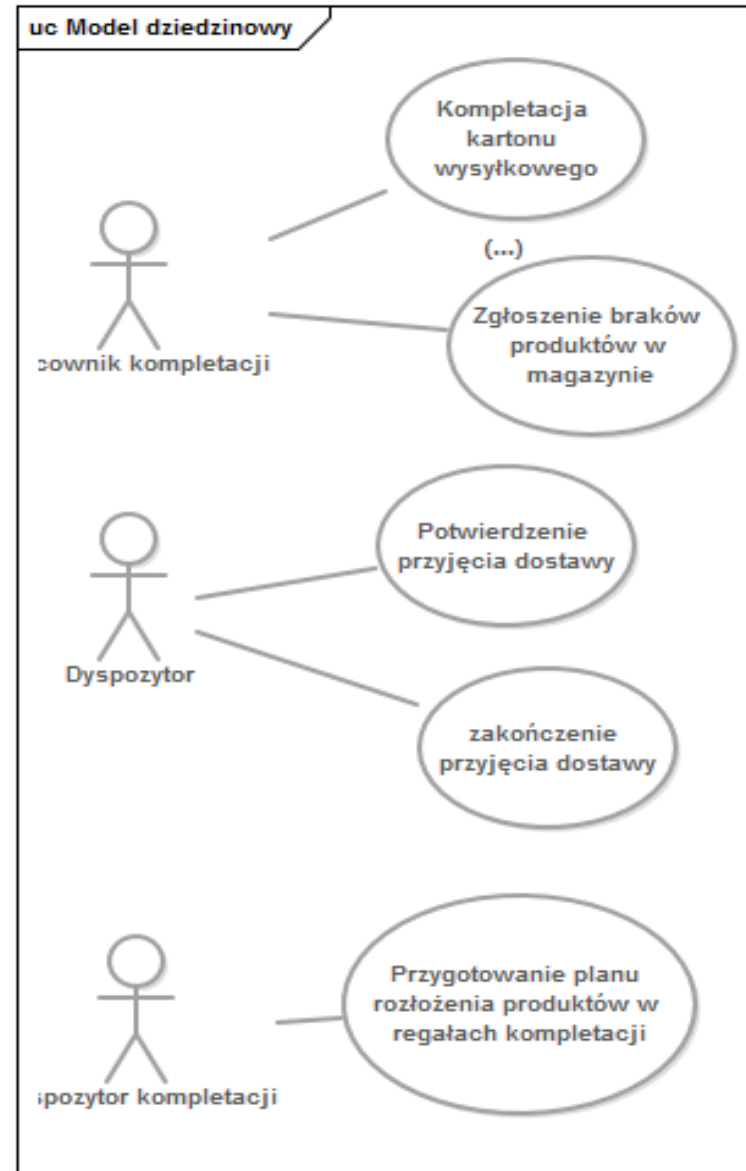
- Zadaniem nie jest np. *Przejrzenie zamówień*, ale jest już np. *Ustawienie kolejności realizacji zamówień.*, *Wyszukanie opóźnionych zamówień.*

Jak robić analizę zadań?

- Rozmowy z 'przyszłymi' użytkownikami lub kierownictwem organizacji.
- Bezpośrednia obserwacja aktualnie wykonywanej pracy.

Dla każdego zadania powinniśmy określić:

- Zależności: od jakich innych zadań zależy dane zadanie?
- Jakich informacji, danych, dokumentów potrzebuje użytkownik aby wykonać zadanie?  
(to będzie potrzebne do projektowania interfejsu)

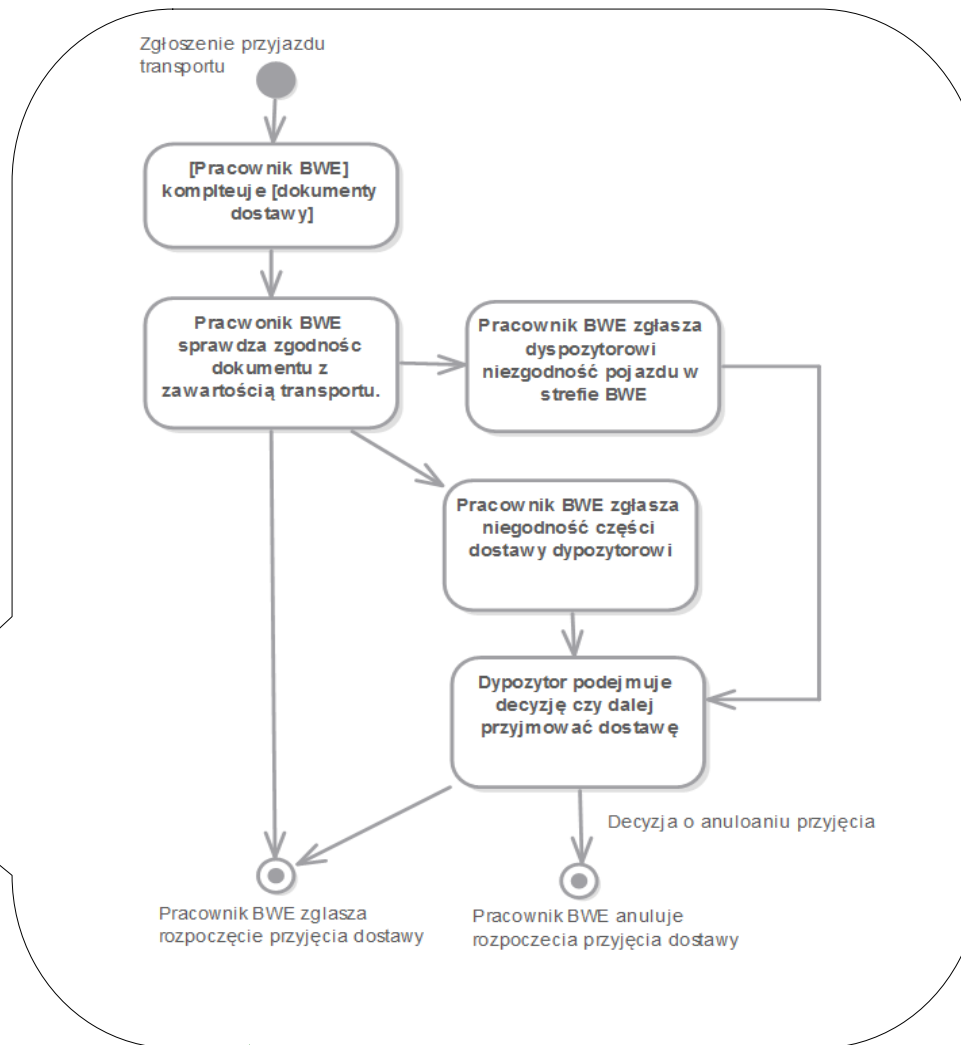




# Analiza wymagań (A)

## Etap Odkrywania / Analizy:

- 1) określ klasy użytkowników (aktorów)
- 2) określ zadania użytkowników (oczekiwania)
- **3) scenariusze wykonywania zadań**
- 4) zbuduj projekt przypadków użycia



**Diagram aktywności!**



# Scenariusze wykonywania zadań (A)

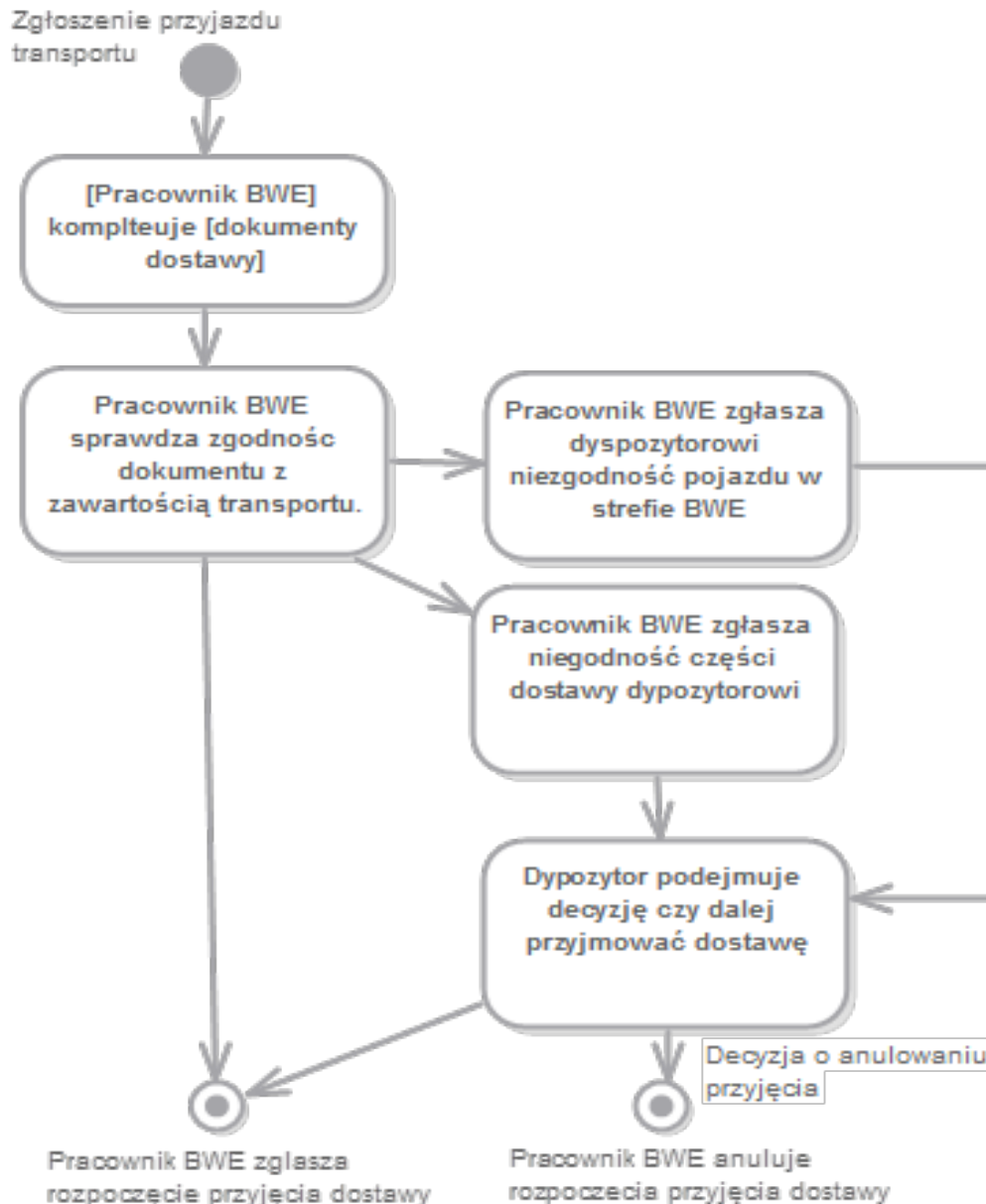
Przedstaw czynności w formie  
**procesu dziedzinowego**

Podczas projektowania koncentruj się  
na procesie dziedzinowym, a nie  
na systemie. **Używaj jak  
najmniejszej liczby  
rzeczowników – dzięki temu  
zachowasz spójność.**

3.1 Ćwiczenie: nie używaj słowa system

Źródła informacji o procesach:

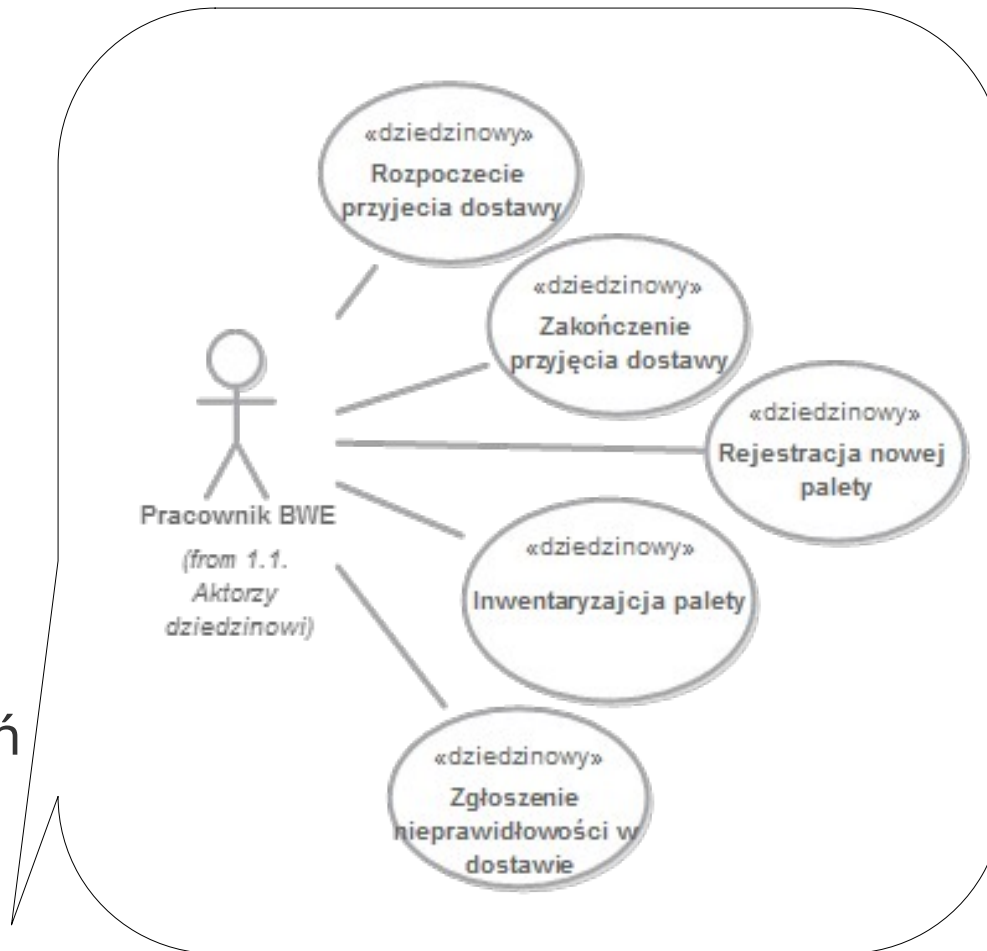
- opisy procesów otrzymanych od użytkownika
- obserwacja aktualnie wykonywanych czynności
- samodzielna analiza



# Analiza wymagań (A)

## Etap Odkrywania / Analizy:

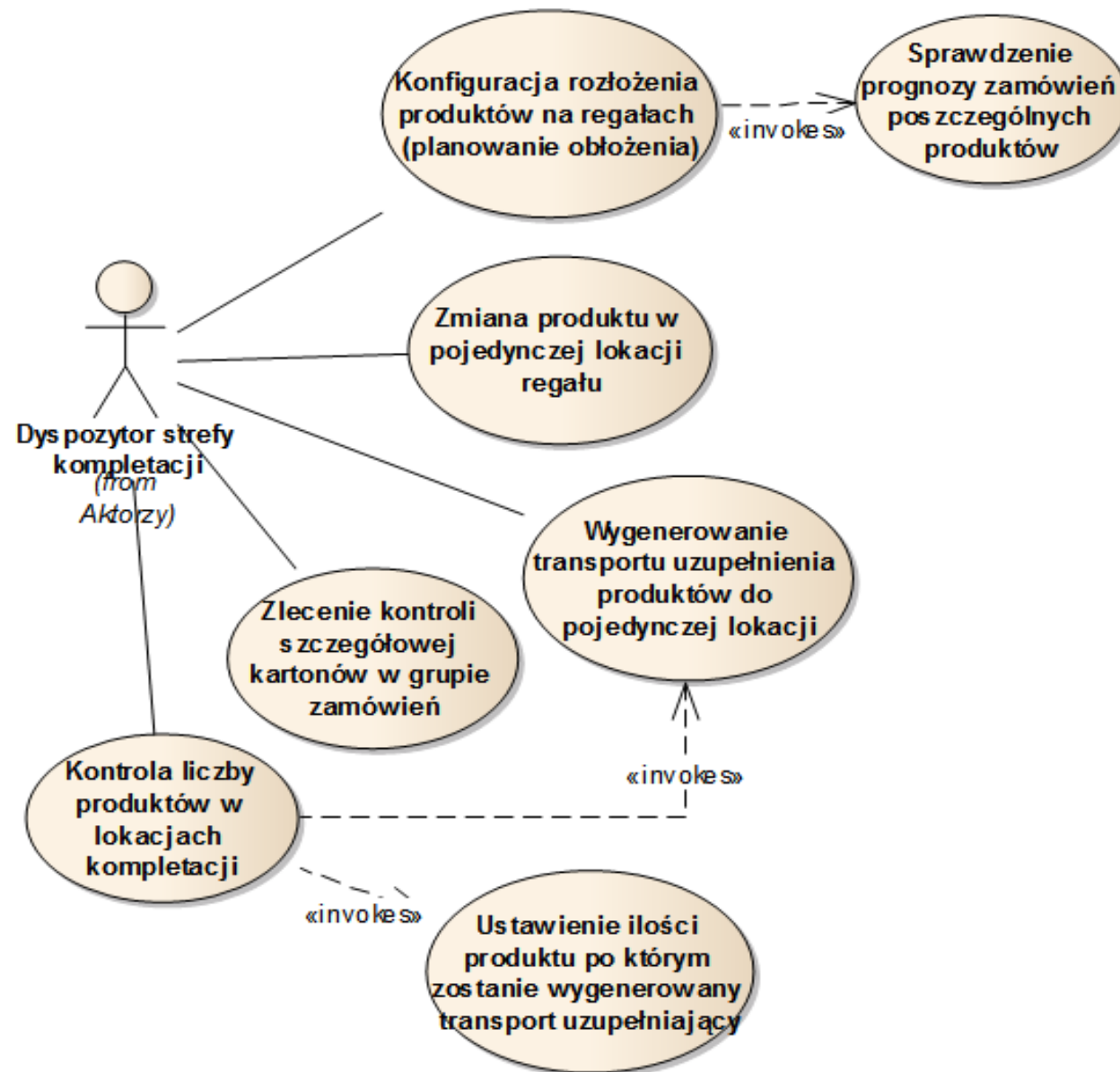
- 1) określ klasy użytkowników (aktorów)
- 2) określ zadania użytkowników (oczekiwania)
- 3) scenariusze wykonywania zadań



- 4) zbuduj projekt przypadków użycia  
*(które stają się podstawą formalnej specyfikacji wymagań; przypadki użycia w pierwszej wersji mogą być celami, z punktu 2., które mogą być osiągnięte przy wsparciu systemu informatycznego oraz złożonymi czynnościami z punktu 3.)*

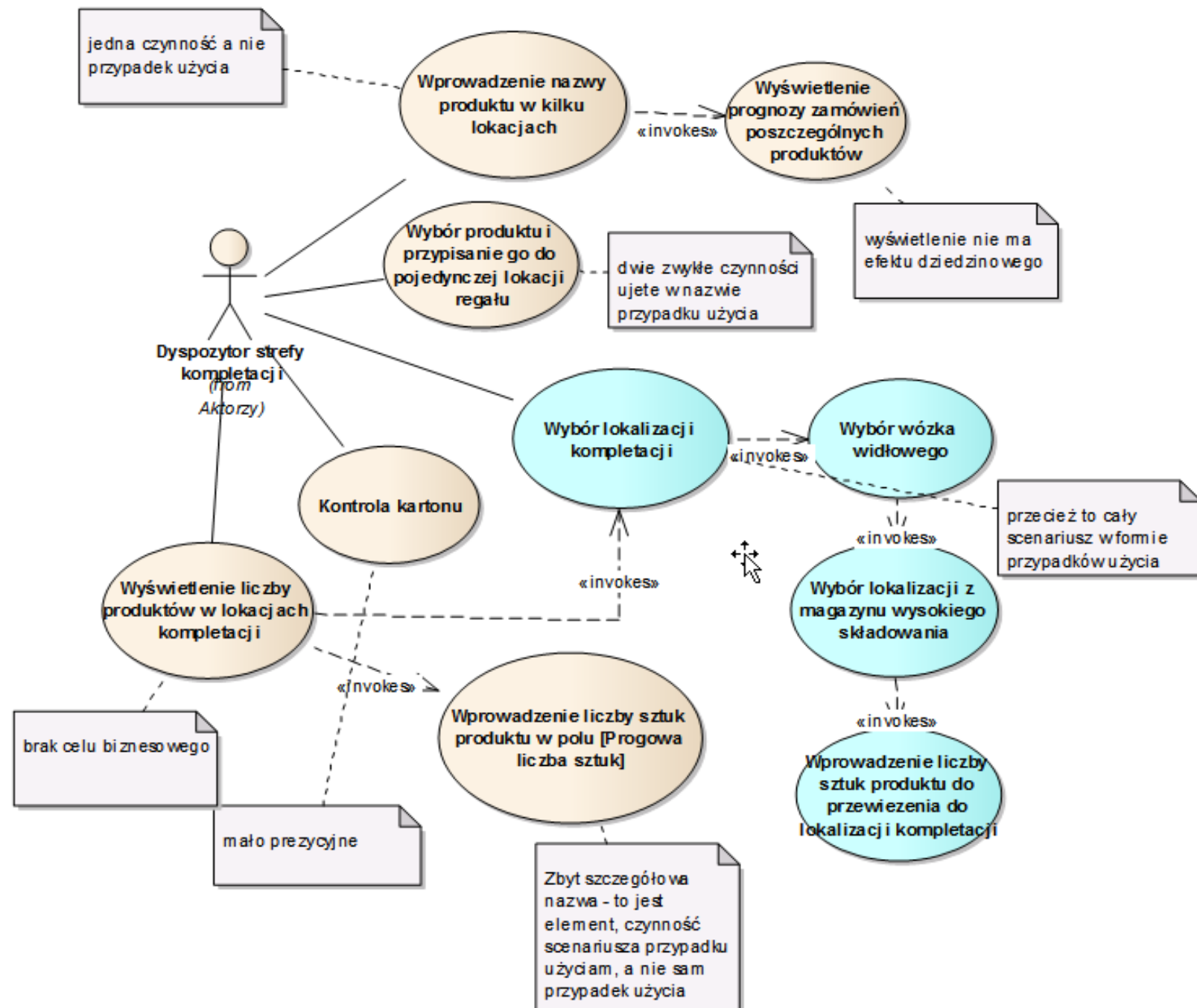


# Diagram przypadków użycia - DOBRZE

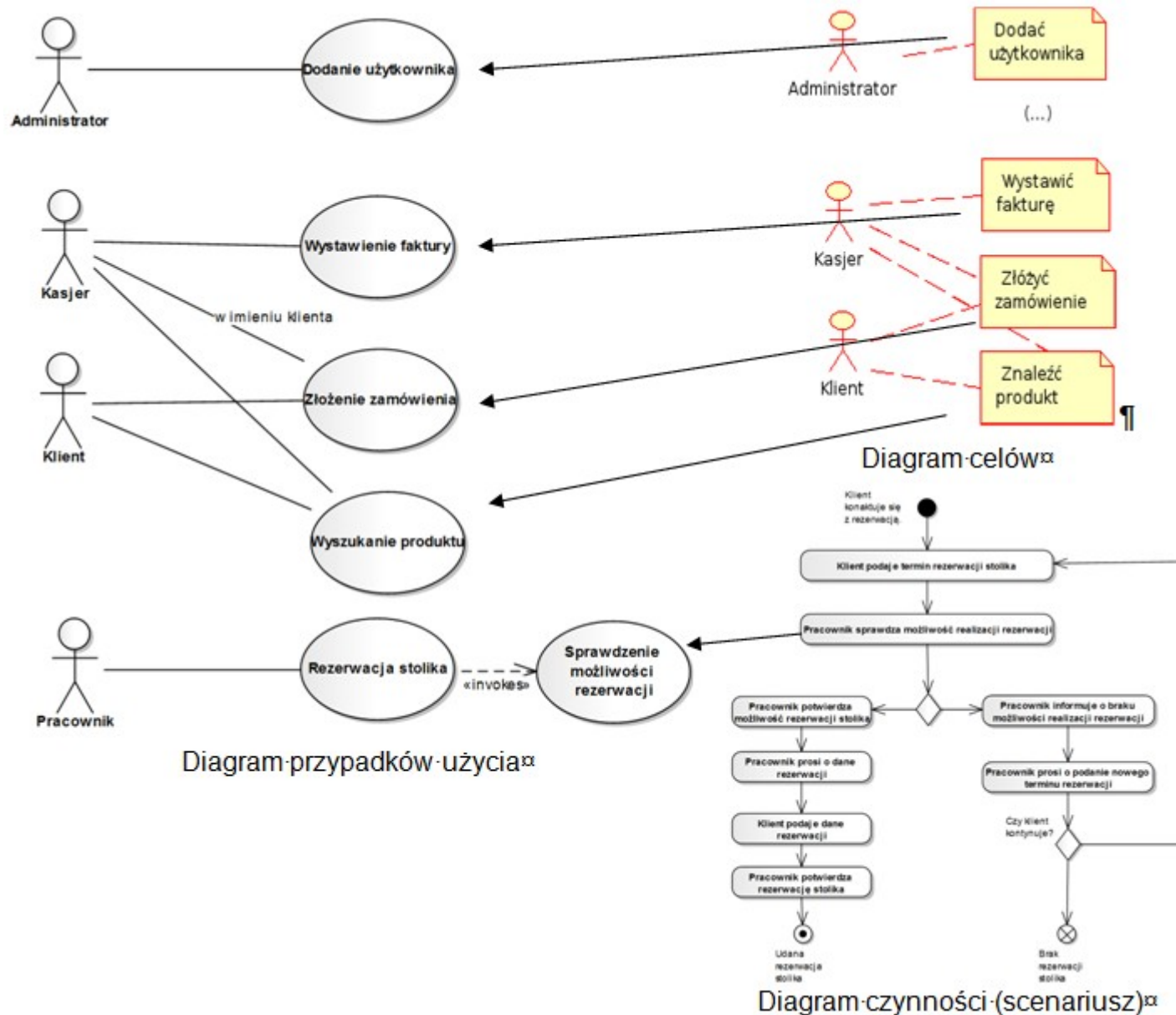




# Diagram przypadków użycia – typowe błędy



# Praktycznie: skąd wziąć przypadki użycia



Przypadkami użycia stają się:

- zadania
- czynności ze scenariuszy zadań

Przypadek użycia musi kończyć się weryfikowalnym efektem, zmianą stanu biznesowego lub systemowego.



# Projekt funkcjonalny (B) + sekwencyjny (C)

Projekt funkcjonalny – model struktury obiektów dziedzinowych oraz okien, widoków i operacji na wszystkich obiektach oraz danych wejściowych i wyjściowych.

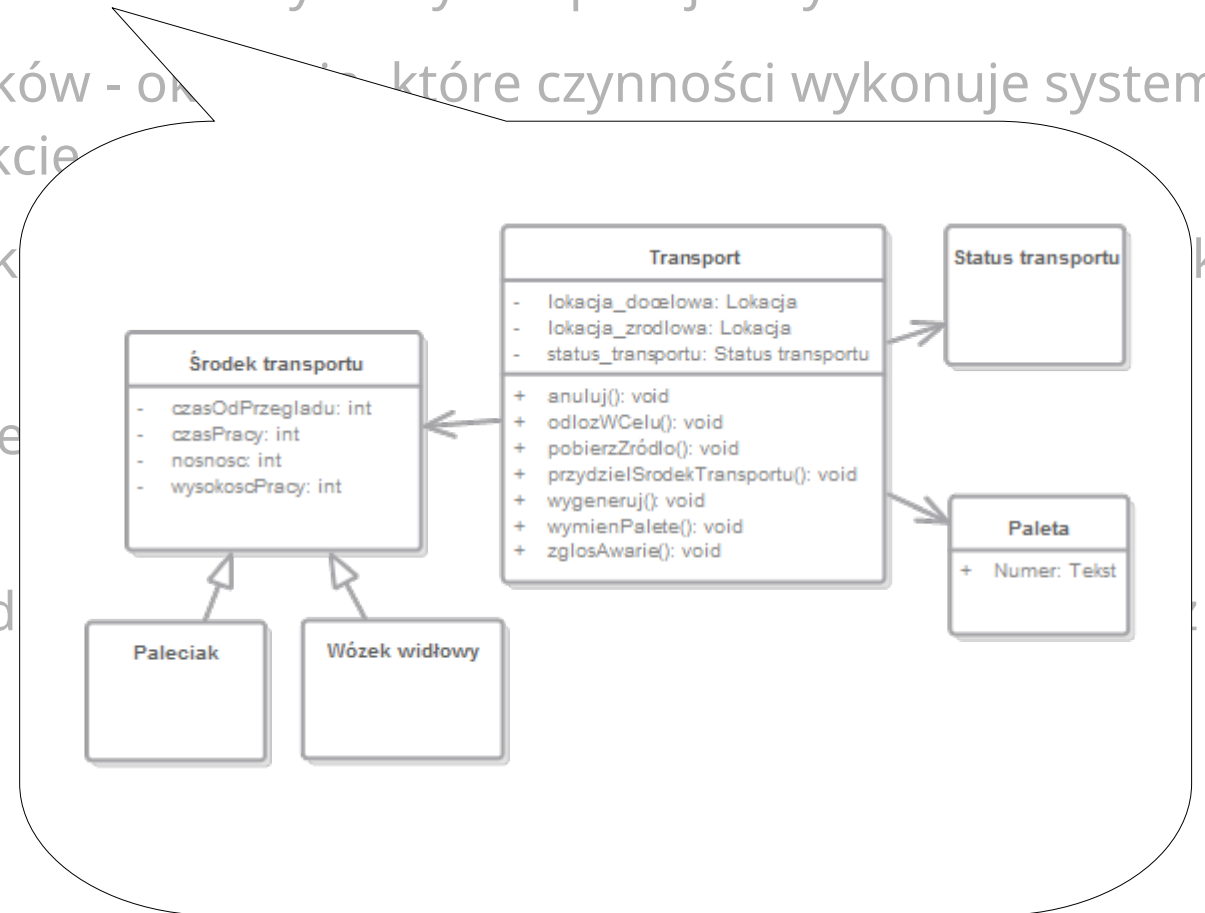
- 1) **Obiekty Dziedziny (B)** - określ obiekty / elementy / pojęcia na których wykonuje operacje użytkownik i system (słownik pojęć / dziedziny)
- 2) **Scenariusze Przypadków (B)** – określają, które czynności wykonuje system a które użytkownik w trakcie realizacji przypadku użycia – powstają punkty styku użytkownik-system
- 3) **Okna i widoki Abstrakcyjne (B)** - jakie informacje musi widzieć użytkownik aby ukończyć poszczególne zadania (związane z konkretnymi przypadkami użycia), które elementy obiektów dziedzinowych są w danym momencie potrzebne użytkownikowi, (powstaje statyczna **struktura logiczna interfejsu**)
- 4) **Przepływ Zadań (Sekwencje) (C)** - sprecyzuj dokładnie interakcję użytkownika z systemem za pomocą diagramów sekwencji uwzględniających cykl życia ekranów
- 5) **Stany Obiektów (C)** - co dzieje się danymi obiektami w trakcie interakcji z użytkownikiem → diagramy stanów obiektów (te diagramy będziemy wykorzystywać do różnej prezentacji wizualnej stanu)



# Projekt funkcjonalny (B)

Projekt funkcjonalny (semantyczny) – zdefiniowanie operacji na wszystkich obiektach oraz danych wejściowych i wyjściowych.

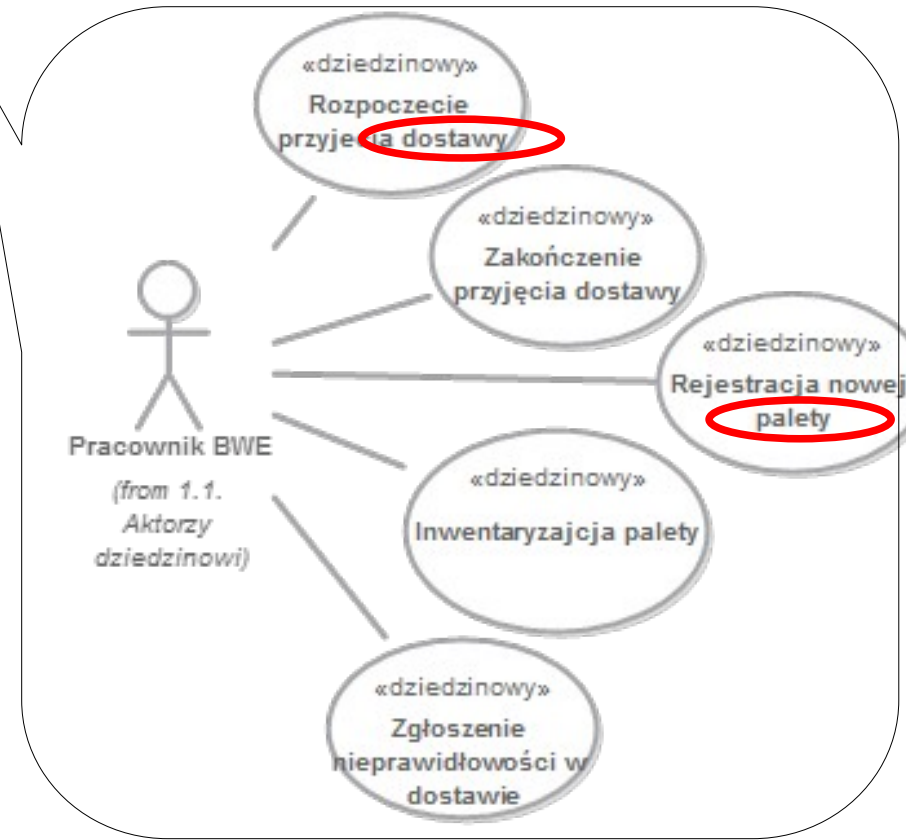
- 1) **Obiekty Dziedziny** - określ obiekty którymi operuje użytkownik
- 2) Scenariusze Przypadków - określ które czynności wykonuje system a które użytkownik w trakcie
- 3) Okna i widoki Abstrakcyjne - określ jak system ma wyglądać aby ukończyć zadanie
- 4) Przepływ Zadań - określ jak system ma działać
- 5) Stany Obiektów - określ co dzieje się z danymi użytkownika





# Projekt funkcjonalny – Obiekty dziedziny (B)

- 1) Wyodrębnij obiekty – rzeczowniki z diagramu zawierającego zadania użytkowników
- 2) Czasami tworzy się tabelę zależności, w celu wyselekcjonowania kluczowych obiektów (ważne w iteracjach całego procesu)
- np.: konkretne obiekty, osoby, dokumenty, użytkownicy, obiekty abstrakcyjne
- 3) Stosujemy jednoznaczne i zrozumiałe dla użytkownika nazwy obiektów (nazywa się to słownikiem pojęciowym / dziedziny).
- 4) Uporządkuj obiekty w modelu uwzględniając ich definicje, atrybuty, operacje oraz relacje w postaci diagramu klas.



# Obiekty dziedziny

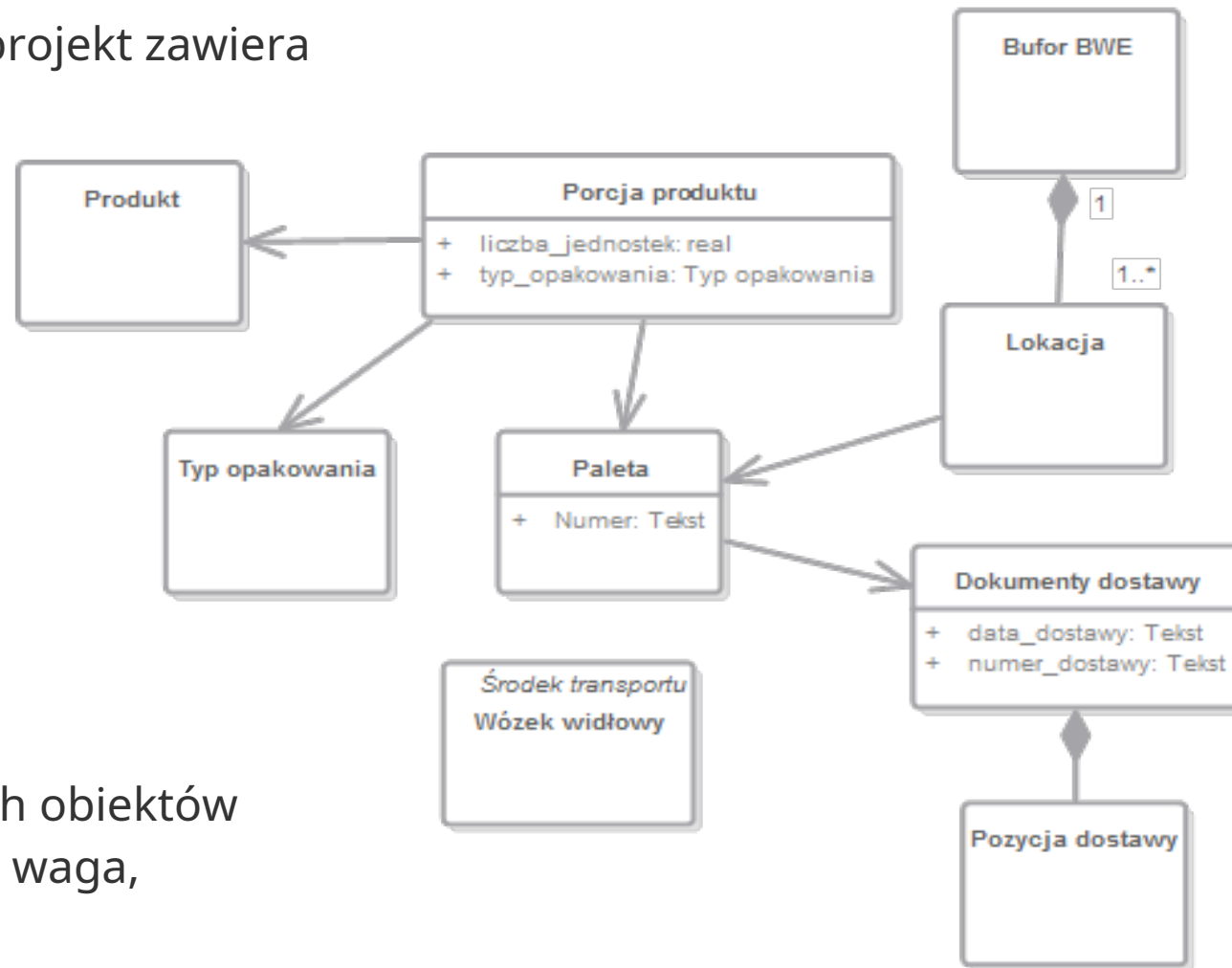
W początkowych etapach projekt zawiera zazwyczaj same nazwy klas i ogólne relacje między nimi.

W trakcie tworzenia projektu pojęciowego klasy powinniśmy uzupełniać o atrybuty, operacje oraz uszczegółowić relacje.

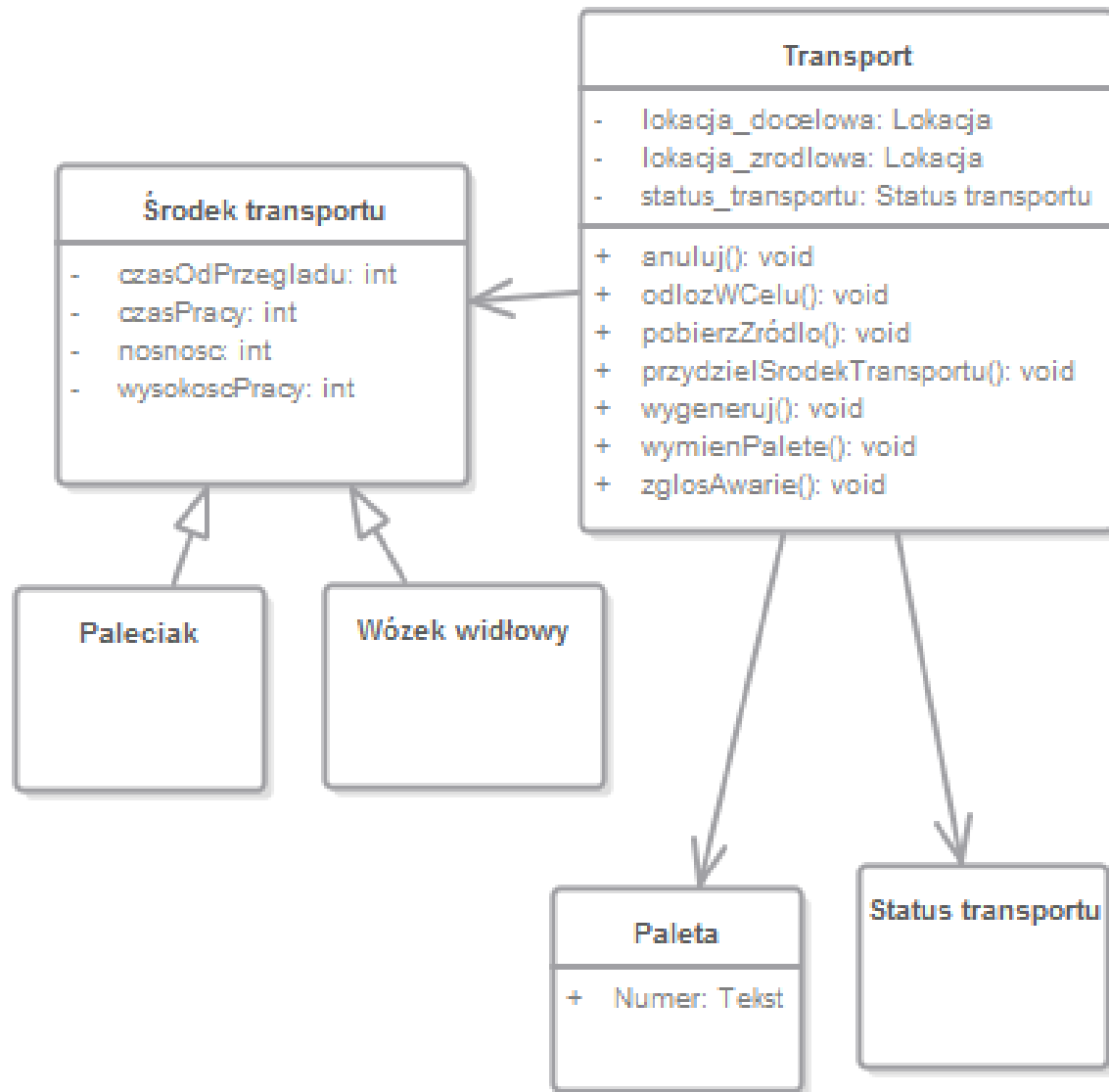
Atrybuty są cechami danych obiektów (np.. kod kreskowy, nazwa, waga, wysokość, nr regału).

Metody są operacjami, które możemy na danych obiektach wykonać.

- np. Pracownik BEW odkłada paletę.



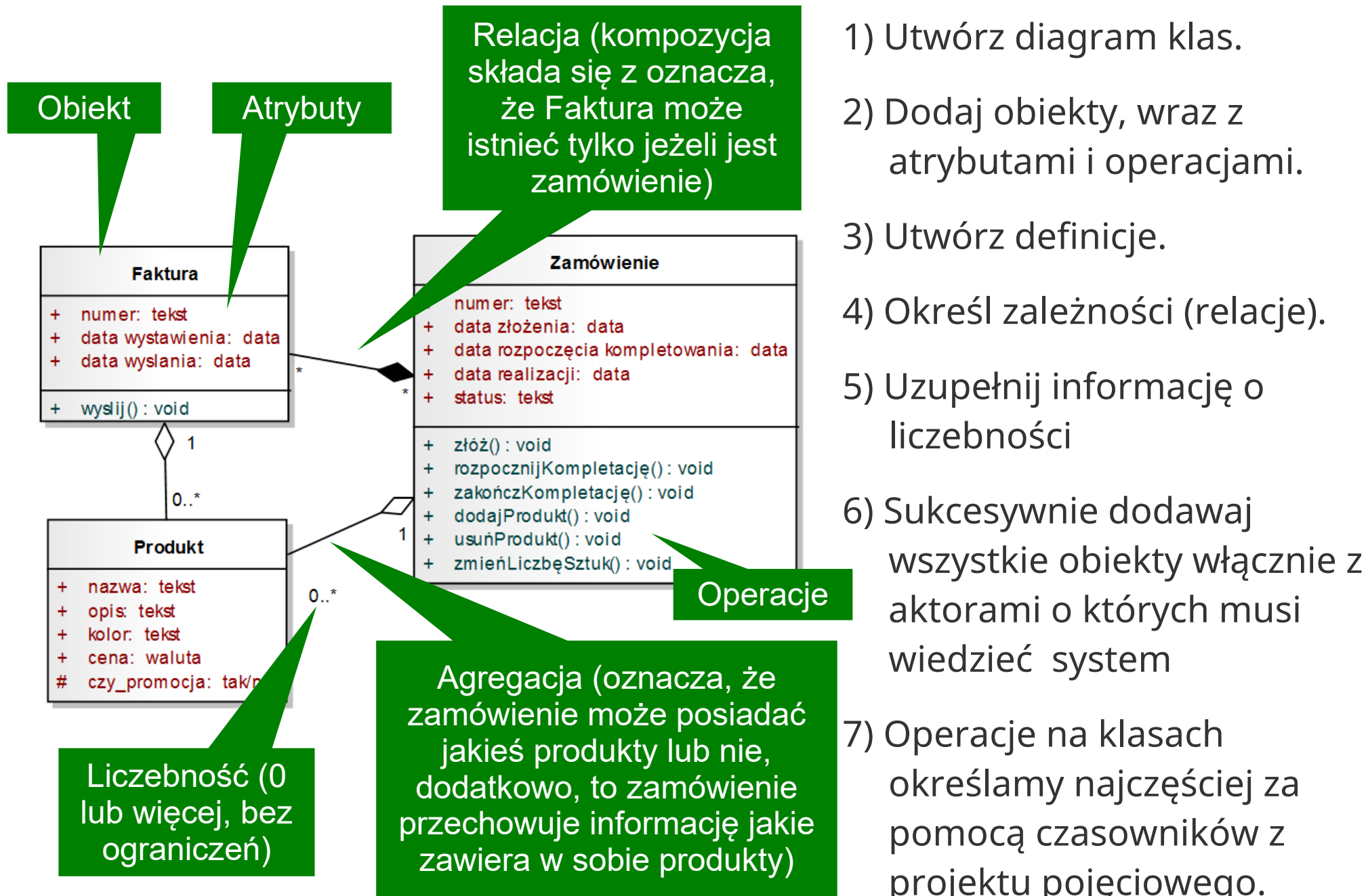
# Jak tworzyć model obiektów dziedziny? (B)



- 1) Utwórz diagram klas.
- 2) Dodaj obiekty, wraz z atrybutami i operacjami.
- 3) Utwórz definicje.
- 4) Określ zależności (relacje).
- 5) Uzupełnij informację o liczebności
- 6) Sukcesywnie dodawaj wszystkie obiekty włącznie z aktorami o których musi wiedzieć system
- 7) Operacje na klasach określamy najczęściej za pomocą czasowników z projektu pojęciowego.



# Jak tworzyć model obiektów dziedziny? (B)



- 1) Utwórz diagram klas.
- 2) Dodaj obiekty, wraz z atrybutami i operacjami.
- 3) Utwórz definicje.
- 4) Określ zależności (relacje).
- 5) Uzupełnij informację o liczebności
- 6) Sukcesywnie dodawaj wszystkie obiekty włącznie z aktorami o których musi wiedzieć system
- 7) Operacje na klasach określamy najczęściej za pomocą czasowników z projektu pojęciowego.





# Złożoność diagramu modeli - stylistyka

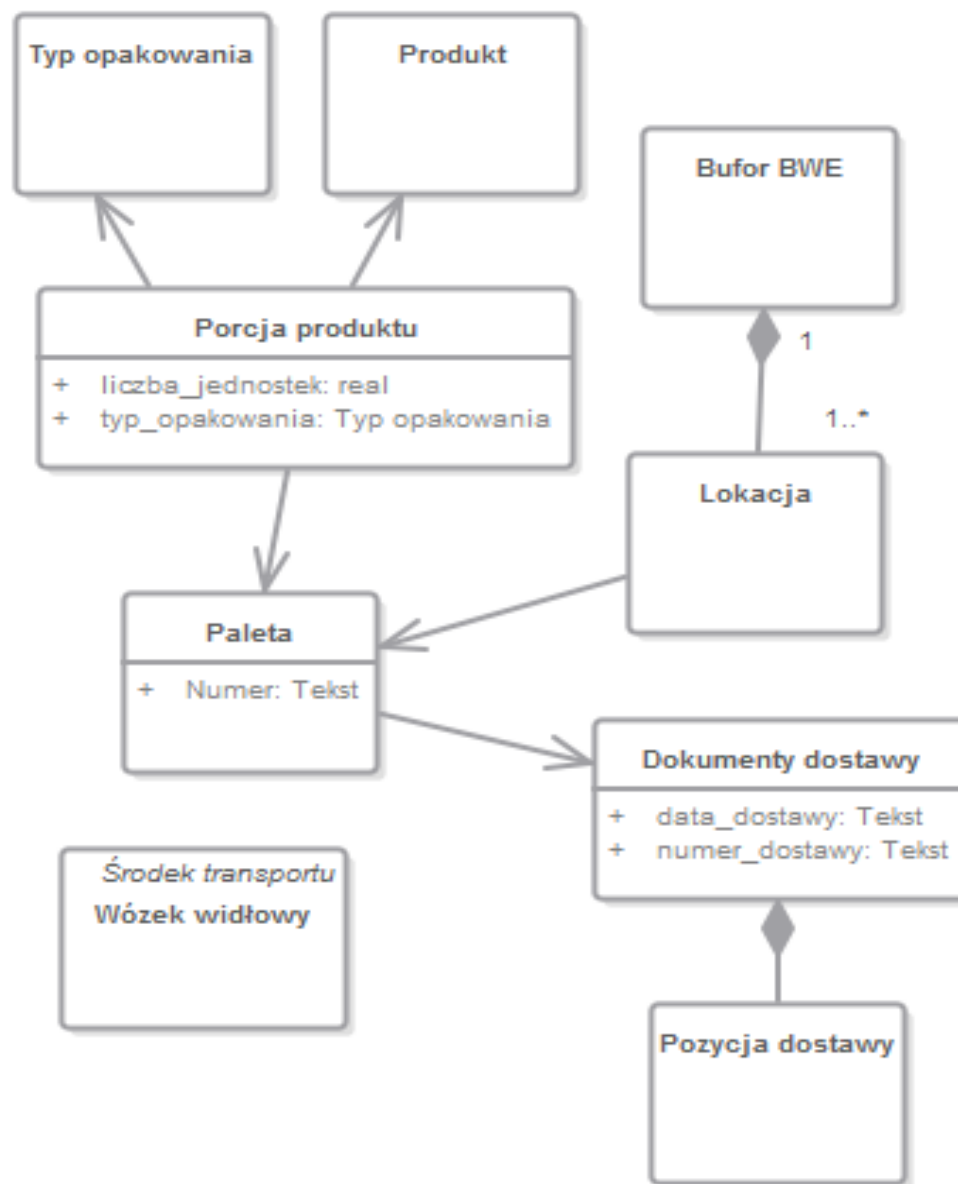
1) **10-12** - Pojedynczy diagram nie powinien zawierać więcej niż 10-12 klas jeżeli klasy nie wyświetlają atrybutów i operacji.

2) **7** - W przypadku, gdy klasy wyświetlają atrybuty i operacji diagram nie powinien przekraczać około 7 klas.

3) **CEL** - Tworząc diagram musimy zdecydować się co chcemy za jego pomocą przedstawić:

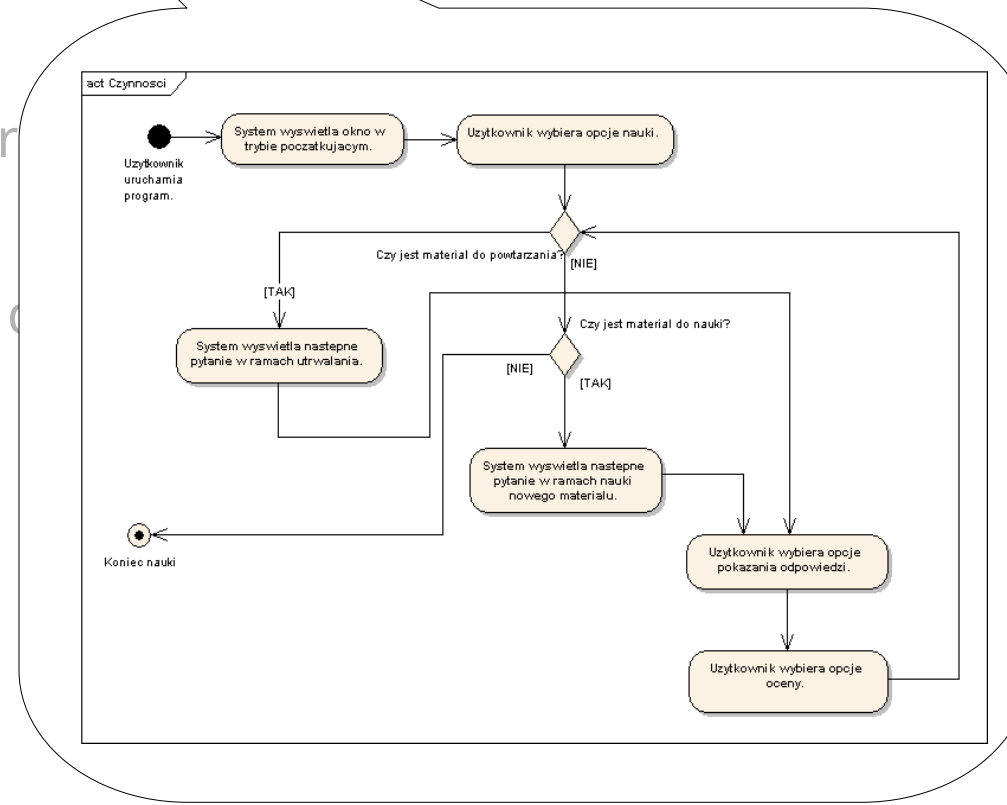
- a) strukturę ogólną pojęć,
- b) zależności między pojęciami,
- c) szczegółową dokumentację każdego pojęcia.

4) **OBIEKT CENTRALNY** – warto skupić się na jednym obiekcie, wokół którego dodajemy obiekty zależne



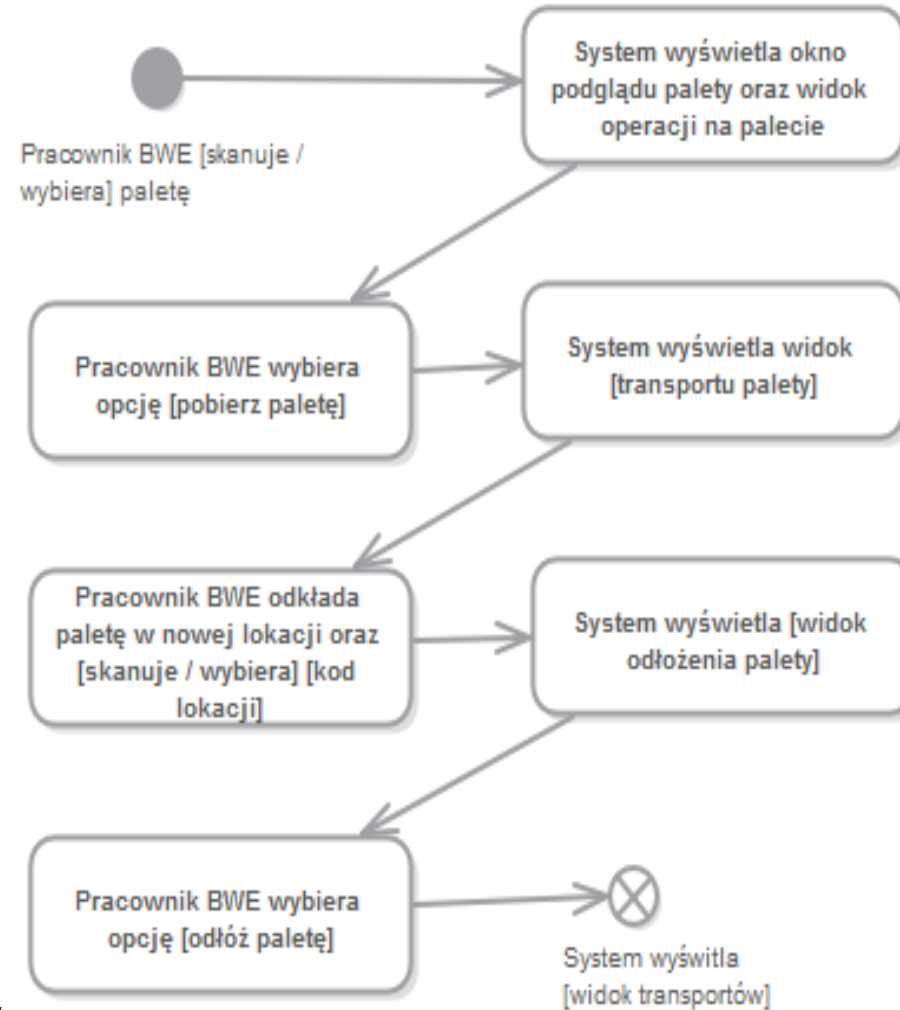
# Projekt funkcjonalny (B)

- 1) Obiekty Dziedziny - określ obiekty którymi operuje użytkownik
- 2) **Scenariusze Przypadków** - określają, które czynności wykonuje system a które użytkownik w trakcie realizacji przypadku użycia (diagramy czynności)
- 3) Okna i widoki Abstrakcyjne - informacje musi widzieć użytkownik aby ukończyć zadanie
- 4) Przepływ Zadań - spracowanie danych przez system
- 5) Stany Obiektów - co obiekty w systemie widzą użytkownik



## Scenariusze przypadków użycia - określ

- Definicja: wysokiego poziomu opis określający dynamikę funkcjonalności (ogólnie sekwencji aktywności oraz punktów decyzyjnych), którą system udostępnia dla użytkownika w celu zrealizowania intencji.
- Punktem startowym modelu jest odwzorowanie aktualnych procesów w instytucji użytkownika, które wykonaliśmy w projekcie pojęciowym (modelu biznesowym, dziedzinowym)



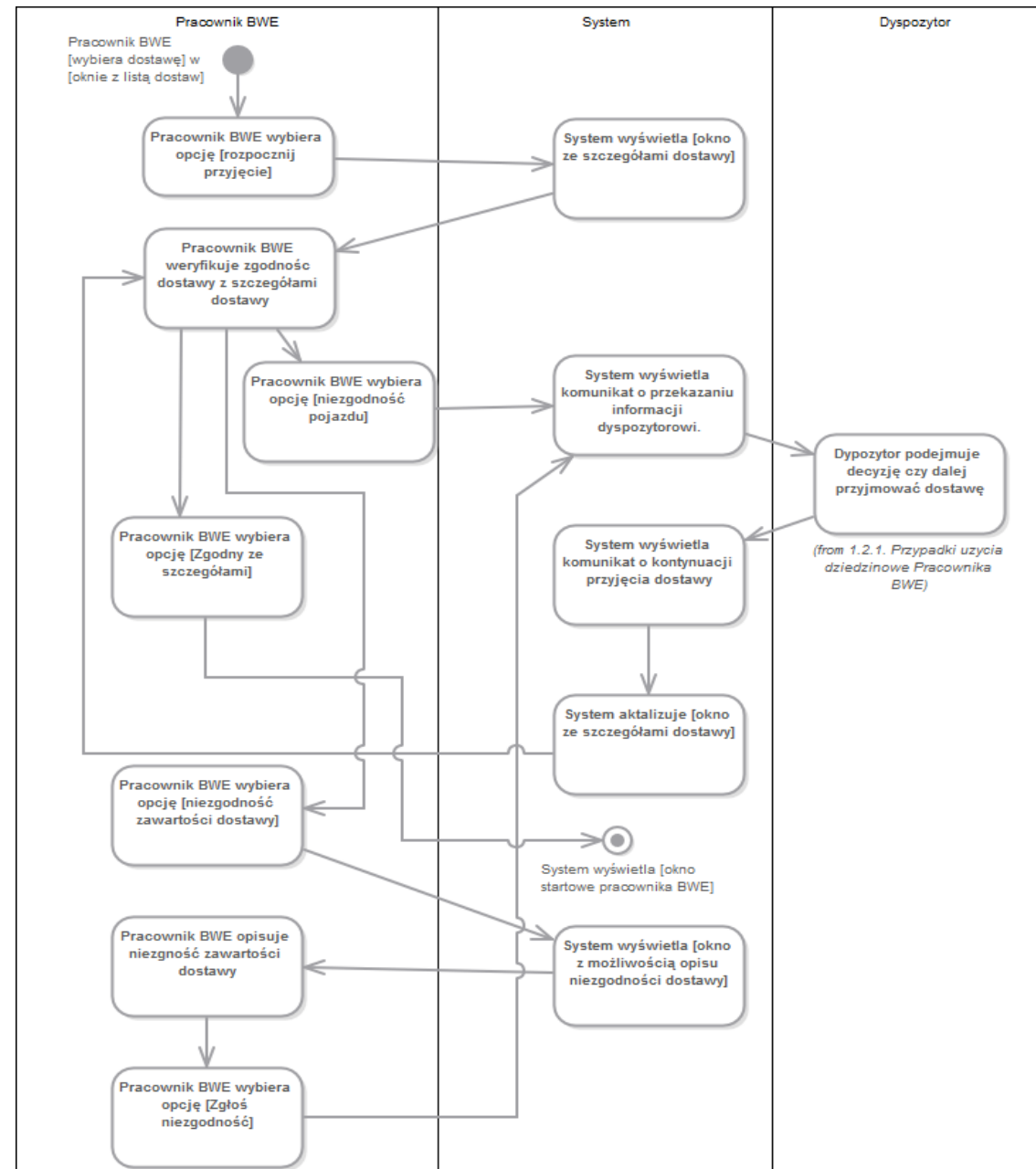
# Podział zadań

Jednoznacznie określamy kto jest odpowiedzialny za wykonanie danej czynności. Stąd nazwa procedury: przegląd zadań.

- 1) za pomocą PODMIOTU zdania
- 2) za pomocą umiejscowienia na diagramie (EA: swimlanes)

Staraj się używać jak najwięcej wyodrębnionych obiektów użytkownika (czyli zdefiniowanych klas) w słowniku dziedzinowym.

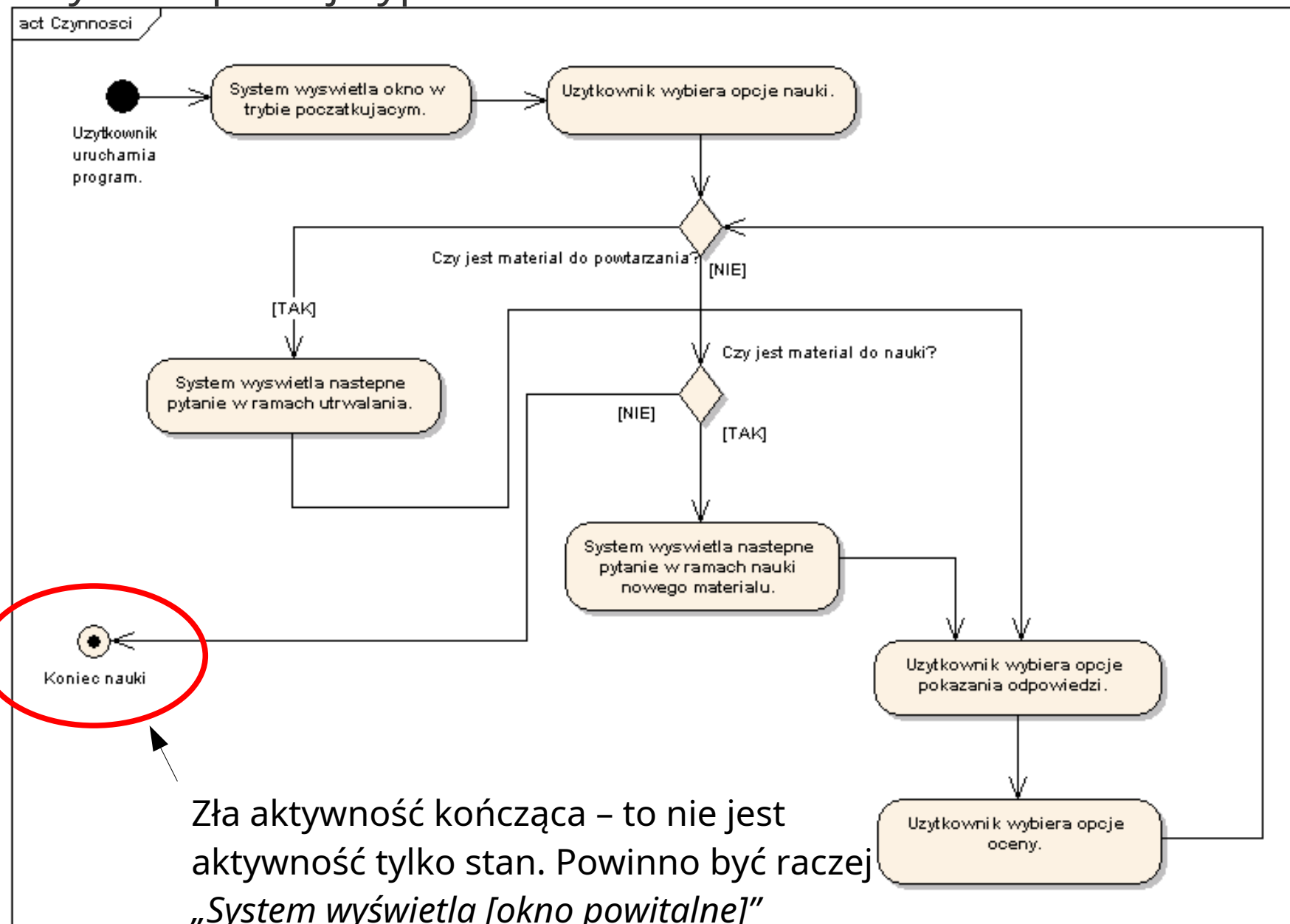
W scenariuszach przypadków użycia powstaje słownik systemowy: obiekty interfejsu użytkownika oraz obiekty logiki biznesowej systemu.





## Diagram czynności – Przykład 2 (B)

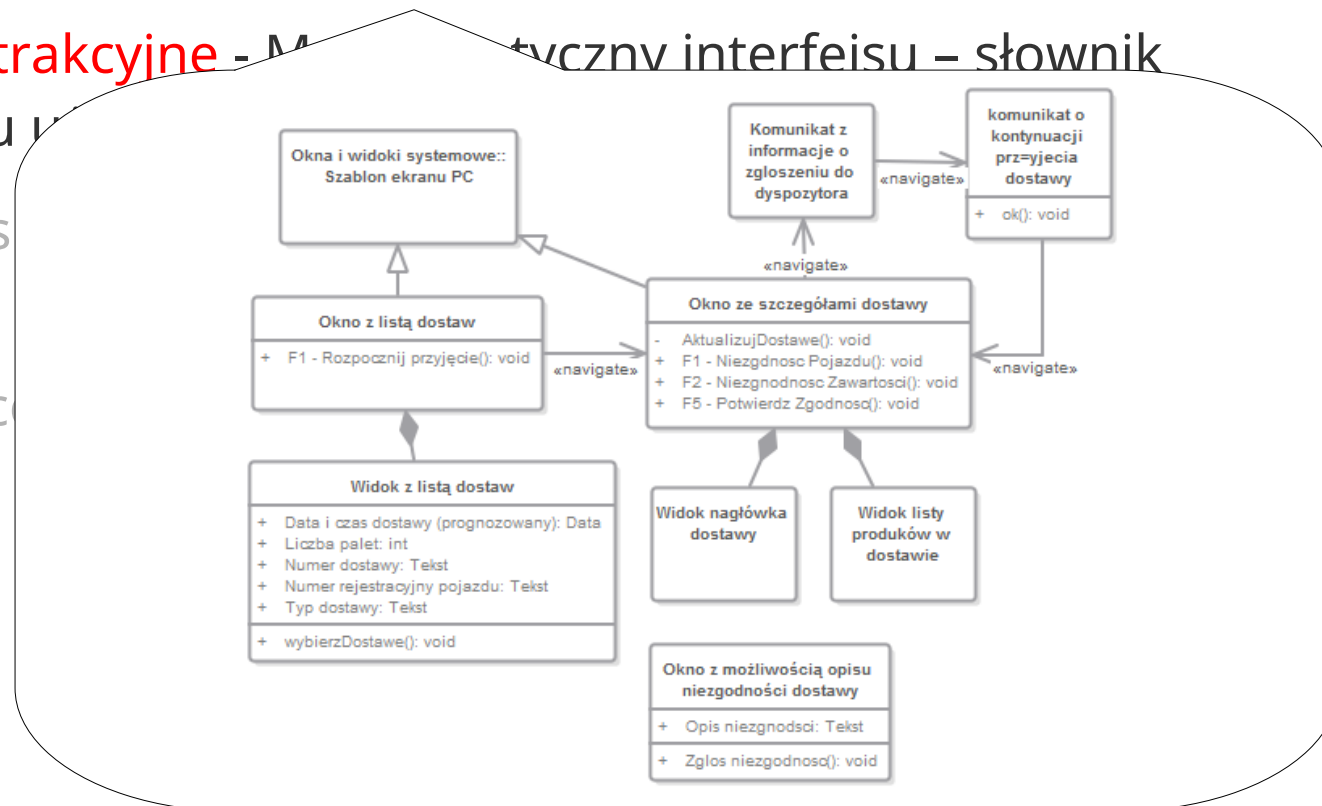
### Przykład aplikacji typu FlashCard.



# Projekt funkcjonalny (B)

Projekt funkcjonalny (semantyczny) – zdefiniowanie operacji na wszystkich obiektach oraz danych wejściowych i wyjściowych.

- 1) Obiekty Użytkownika - określ obiekty/elementy/pojęcia którymi operuje użytkownik
  - 2) Przegląd Zadań - określ jakie czynności wykonuje system a jakie użytkownik
  - 3) **Okna i widoki Abstrakcyjne** - Merytoryczny interfejsu – słownik systemowy interfejsu
  - 4) Przepływ Zadań - s
  - 5) Stany Obiektów - c
- użytkownikiem



# Model statyczny – Okna i widoki abstrakcyjne

## Okna i widoki Abstrakcyjne –

- Definicja 1.: Struktura logiczna, funkcjonalna interfejsu użytkownika.
- Definicja 2.: niezależna od implementacji reprezentacja tego co musi widzieć użytkownik do zakończenia zadania w postaci hierarchicznej struktury elementów widocznych na ekranie.
- Abstrakcyjny model interfejsu użytkownika tworzymy wykorzystując model obiektów użytkownika (diagram klas) oraz przegląd zadań (diagram sekwencji z podziału zadań).
- Mapowanie funkcjonalne obiektów użytkownika na UI.

## Motywacja:

- cel: określenie minimalnych informacji z poszczególnych obiektów, które musi widzieć użytkownik aby ukończyć zadanie oraz jaka w danym kroku musi być udostępniona funkcjonalność.
- Ideą okien i widoków abstrakcyjnych jest możliwość skoncentrowania się na treści oraz funkcjonalności odseparowanej od prezentacji.
- W widokach abstrakcyjnych nie interesuje nas jak, ale **CO?** będzie zaprezentowane.



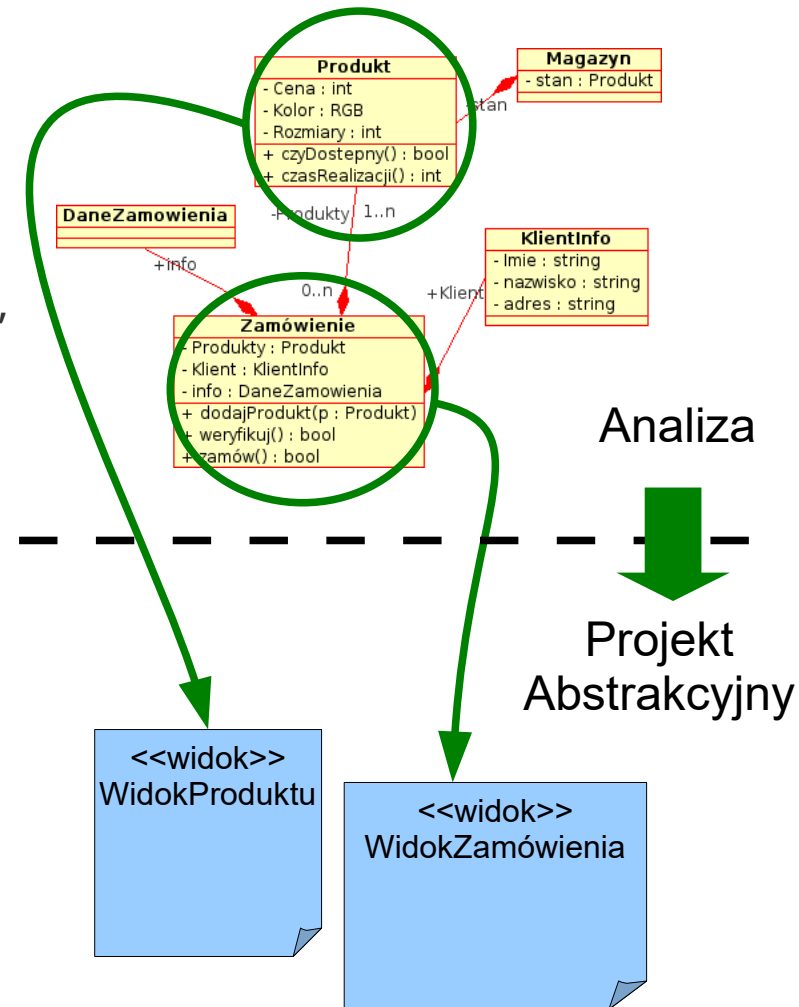
# Widoki i okna abstrakcyjne z obiektów użytkownika

**1. Widoki abstrakcyjne** są elementami, które nie mogą pojawiać samodzielnie. Zawsze są one częścią innych widoków lub są częścią okien.

- **Widoki dziedzinowe obiektów** abstrakcyjne tworzymy zazwyczaj dla każdego obiektu użytkownika, który pojawia się w fazie analizy. Użytkownicy nie pracują na obiektach ale na ich widokach, czyli ich graficznej reprezentacji.
- **Widoki dziedzinowe funkcyjne** abstrakcyjne związane z wykonywaniem poszczególnych czynności, zadań.
- **widoki systemowe** - niezbędne do funkcjonowania systemu z punktu widzenia technicznego.

**2. Okna abstrakcyjne** są samodzielnymi elementami, które pojawiają się użytkownikowi w trakcie pracy z systemem.

- Okna abstrakcyjne są elementami które pojawiają się w scenariuszach.
- Okna abstrakcyjne zbudowane są zazwyczaj z widoków abstrakcyjnych.

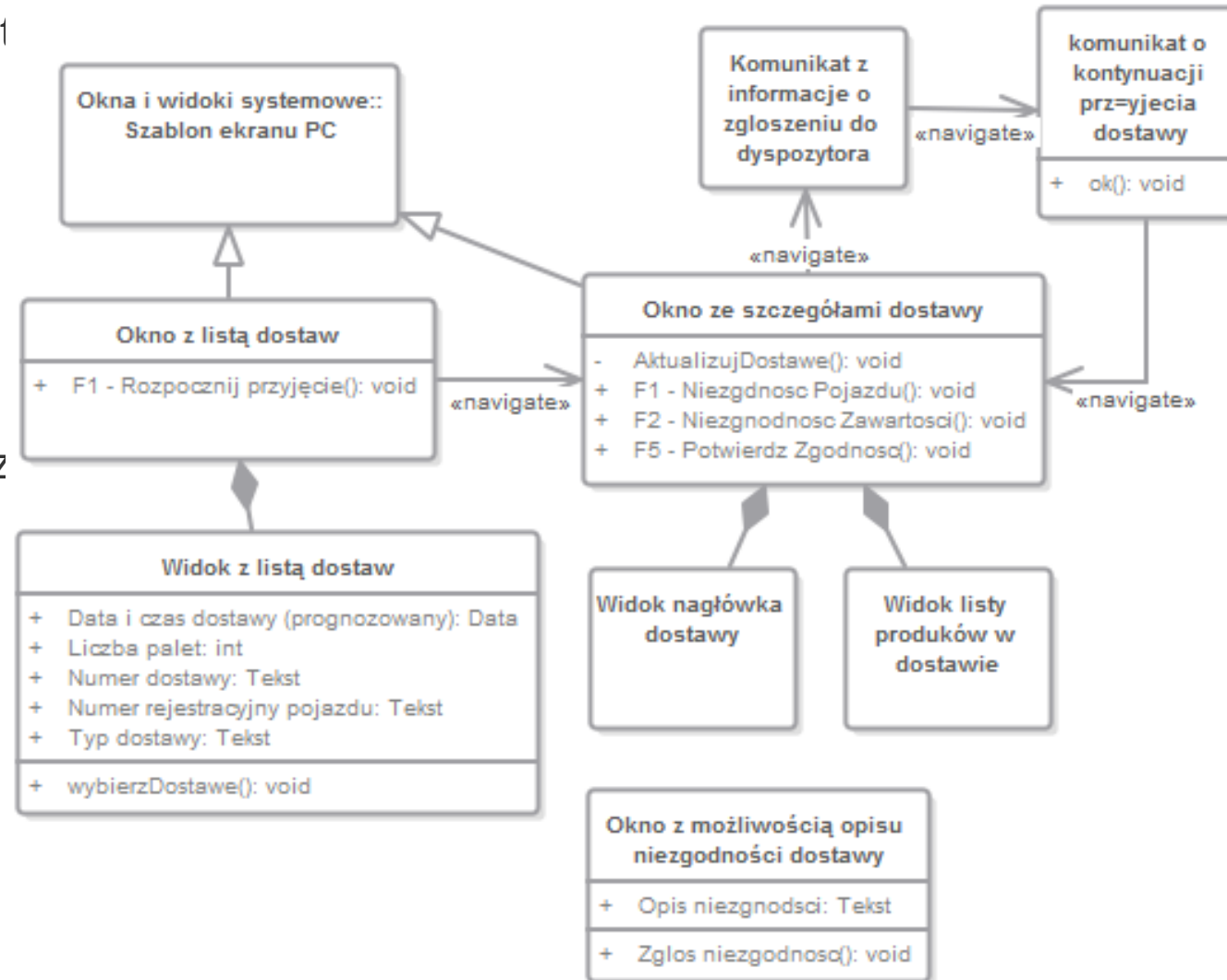


# Widoki dziedzinowe obiektów?

1) Widok dziedzinowy obiektów powinien być związany z jednym lub kilkoma obiektami i służyć do wyświetlania informacji o nich. W widoku określamy jakie komendy mogą być wywołane na danym obiekcie, które elementy są niezbędne do przedstawienia w celu realizacji jakichś powiązanych z widokiem zadań. Jeden obiekt ma zazwyczaj wiele widoków go wyświetlających.

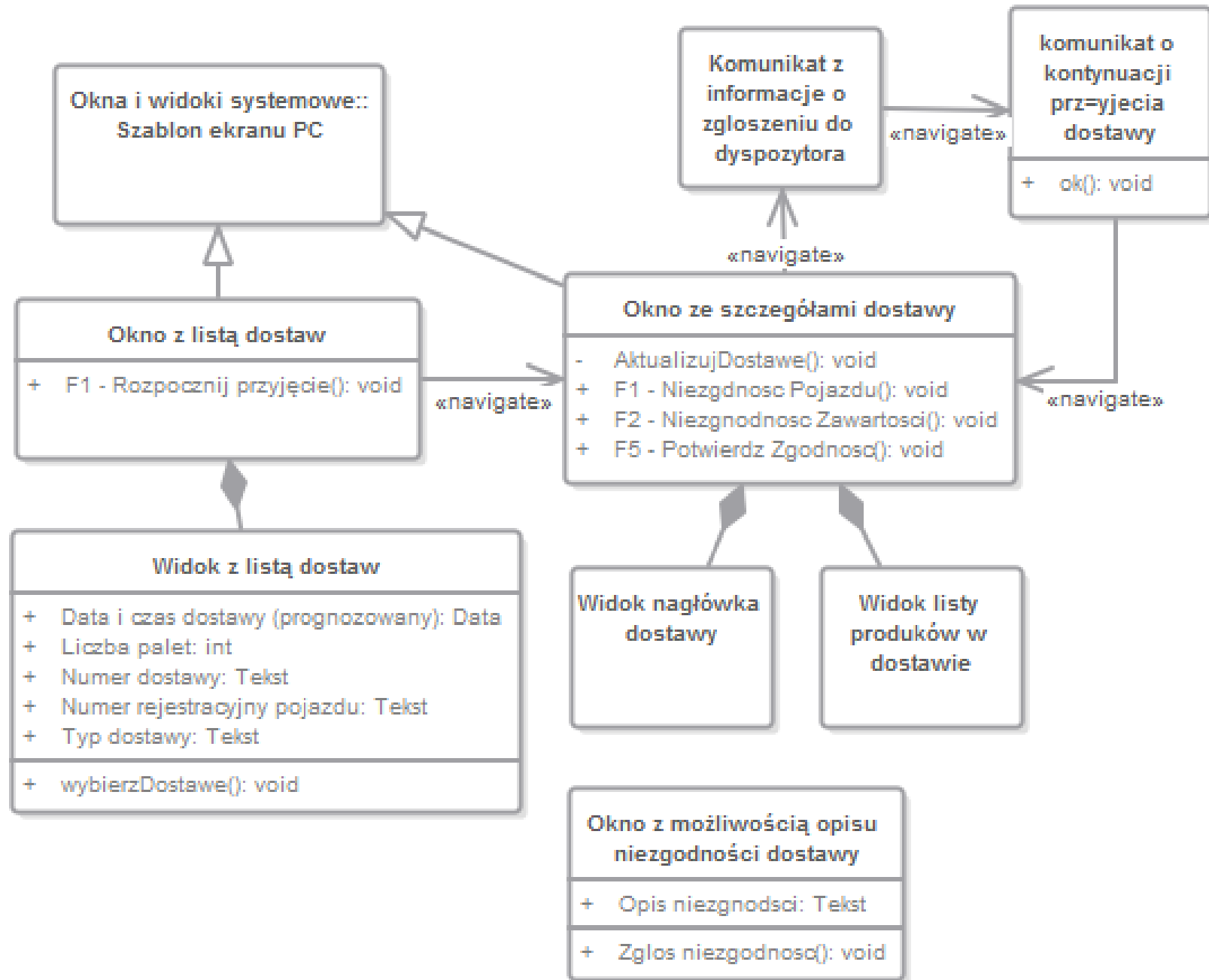
2) Widoki mogą być zbudowane z innych widoków.

3) Przykładowy diagram przedstawia strukturę fragmentu interfejsu wraz z powiązaniem dla Pracownika BWE.





# „Metamodel” reprezentacji struktury UI ? (B)



# „Metamodel” reprezentacji struktury UI

- Klasy reprezentują widoki (<<widok>>), okna (<<okno>>), szablony (<<szablon>>)
- Atrybuty reprezentują elementy wizualne:
  - private – niewidoczne, ale konieczne do zapamiętania stanu interfejsu
  - protected – tylko do odczytu (wartości dziedzinowe, lub etykiety)
  - public – pola edycyjne (można używać dodatkowych stereotypów aby określić rodzaj pola edycyjnego, np.: <<textfield>>, <<select>>)
- Relacje
  - asocjacja – ogólna zależność widoków (jeden może znajdować się w drugim, itp., mogą wyświetlać, odczytywać stan)
  - ukierunkowana asocjacja – wykorzystujemy głównie do określenia nawigacji,
  - agregacja – widok lub okno zawierające w sobie inny widok, którego widoczność może się zmienić w zależności od stanu
  - kompozycja – widok lub okno zawierające w sobie inny widok, który jest zawsze widoczny,
  - dziedziczenie – dziedziczenie funkcjonalności z widoku nadrzędnego



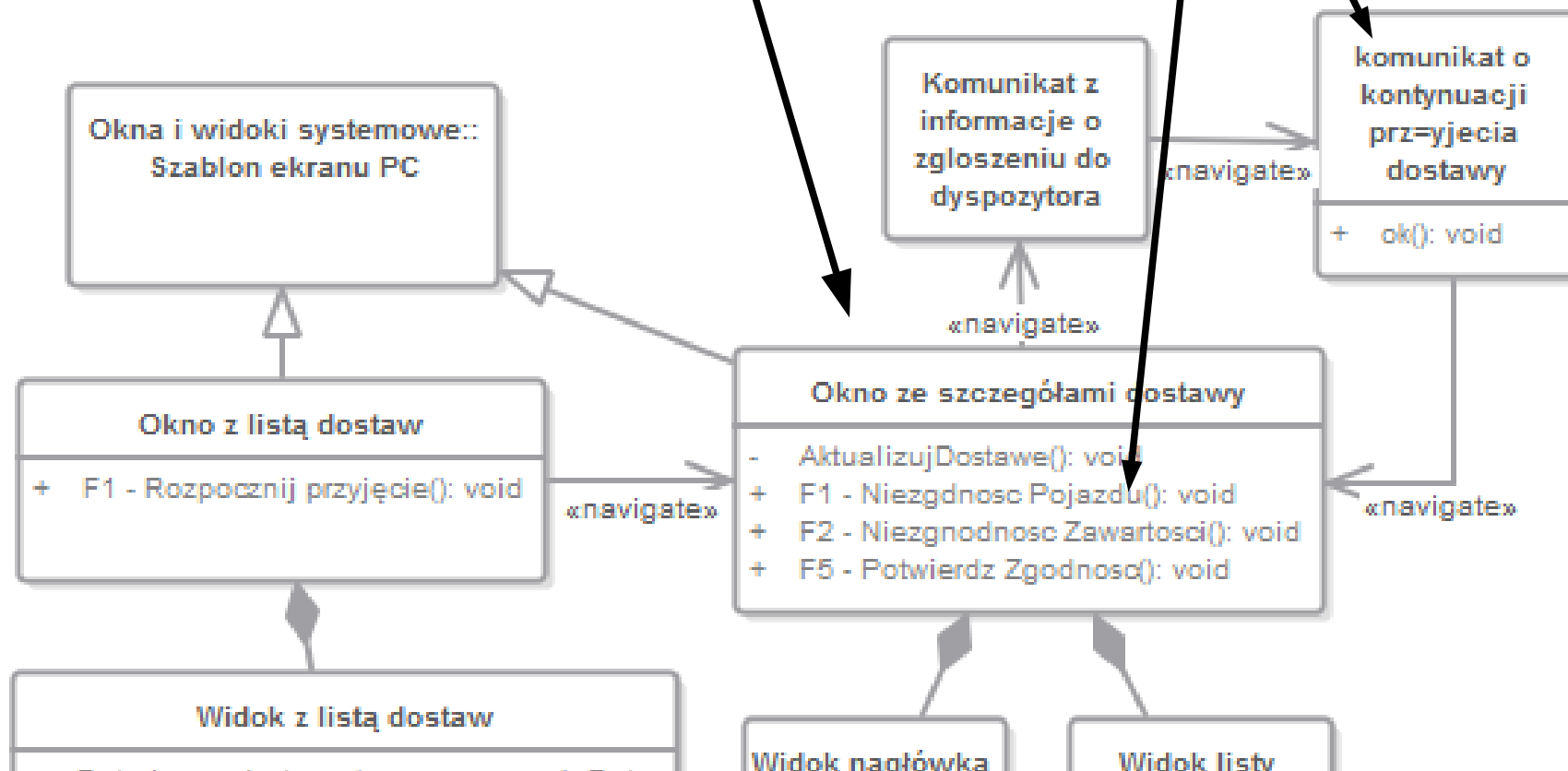
# Widoki i okna funkcyjne z podziału zadań

Kolejnym źródłem powstawania okien i widoków są wszystkie **rzeczowniki z przeglądu zadań i scenariuszy**, które stanowią elementy interfejsu użytkownika:

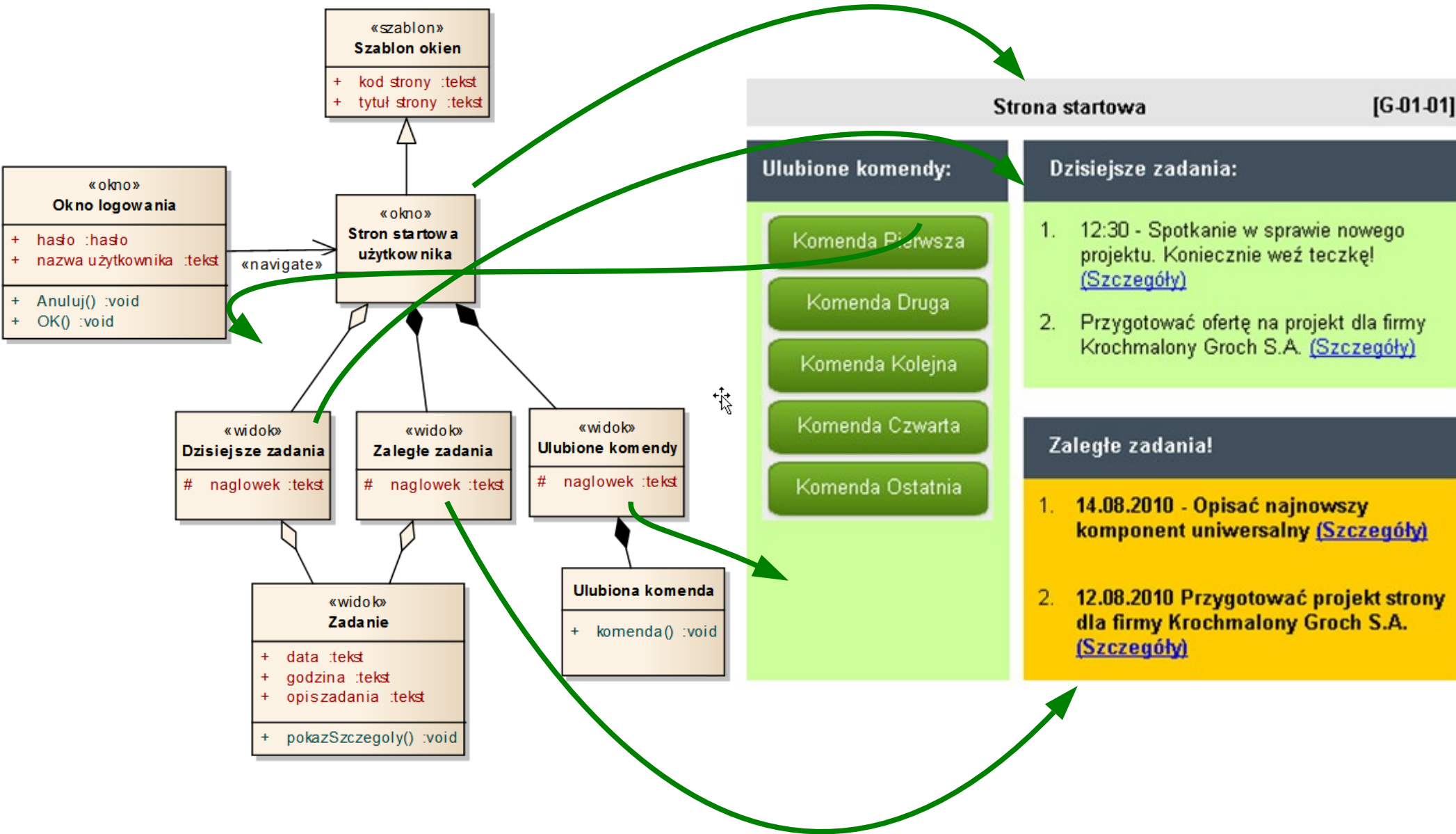
- System wyświetla [Okno ze szczegółami dostawy].
- Pracownik BWE wybiera opcję [Niezgodność pojazdu] dostawy.
- System wyświetla [Komunikat z informacją o zgłoszeniu do dyspozytora].

Analiza

Projekt Abstrakcyjny



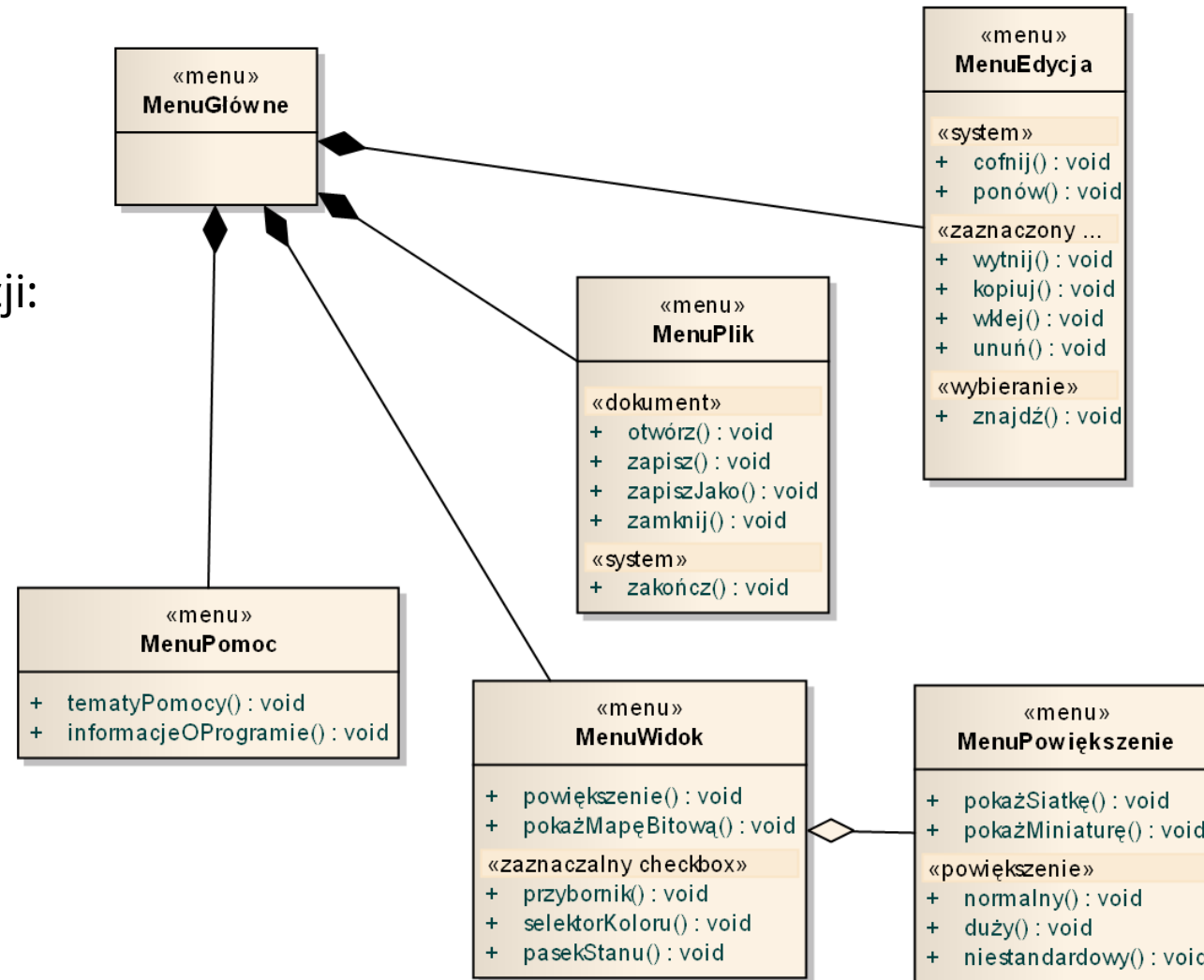
# Przykładowa realizacja graficzna projektu abstrakcyjnego [StronaStartowaUzytkownika]



# Widoki SYSTEMOWE

- Widoki systemowe nie są związane z modelem biznesowym, dziedziną użytkownika lecz są niezbędne do funkcjonowania systemu (np. menu aplikacji, edytor parametrów aplikacji, itp.)

Przykład menu aplikacji:





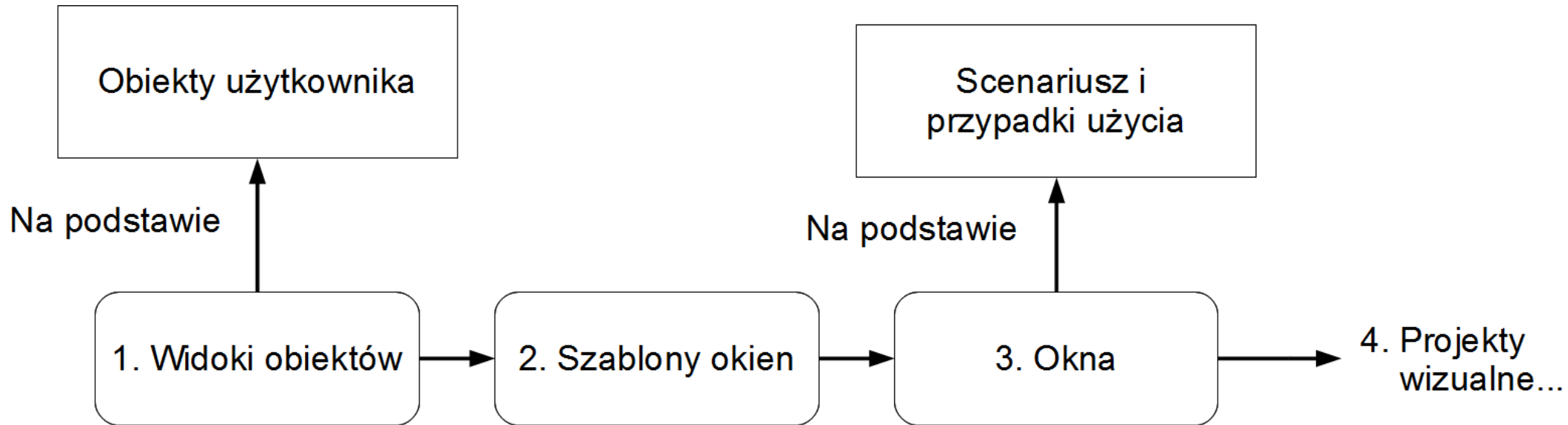
# Widoki SYSTEMOWE

- Widoki systemowe nie są związane z modelem biznesowym, dziedziną użytkownika lecz są niezbędne do funkcjonowania systemu (np. menu aplikacji, edytor parametrów aplikacji, itp.)

Przykład dla Centrum Logistycznego:



# Kolejność tworzenia modelu statycznego



# Projekt sekwencyjny - dynamika (C)

Definiuje kolejność operacji wprowadzania danych wejściowych i wyjściowych.

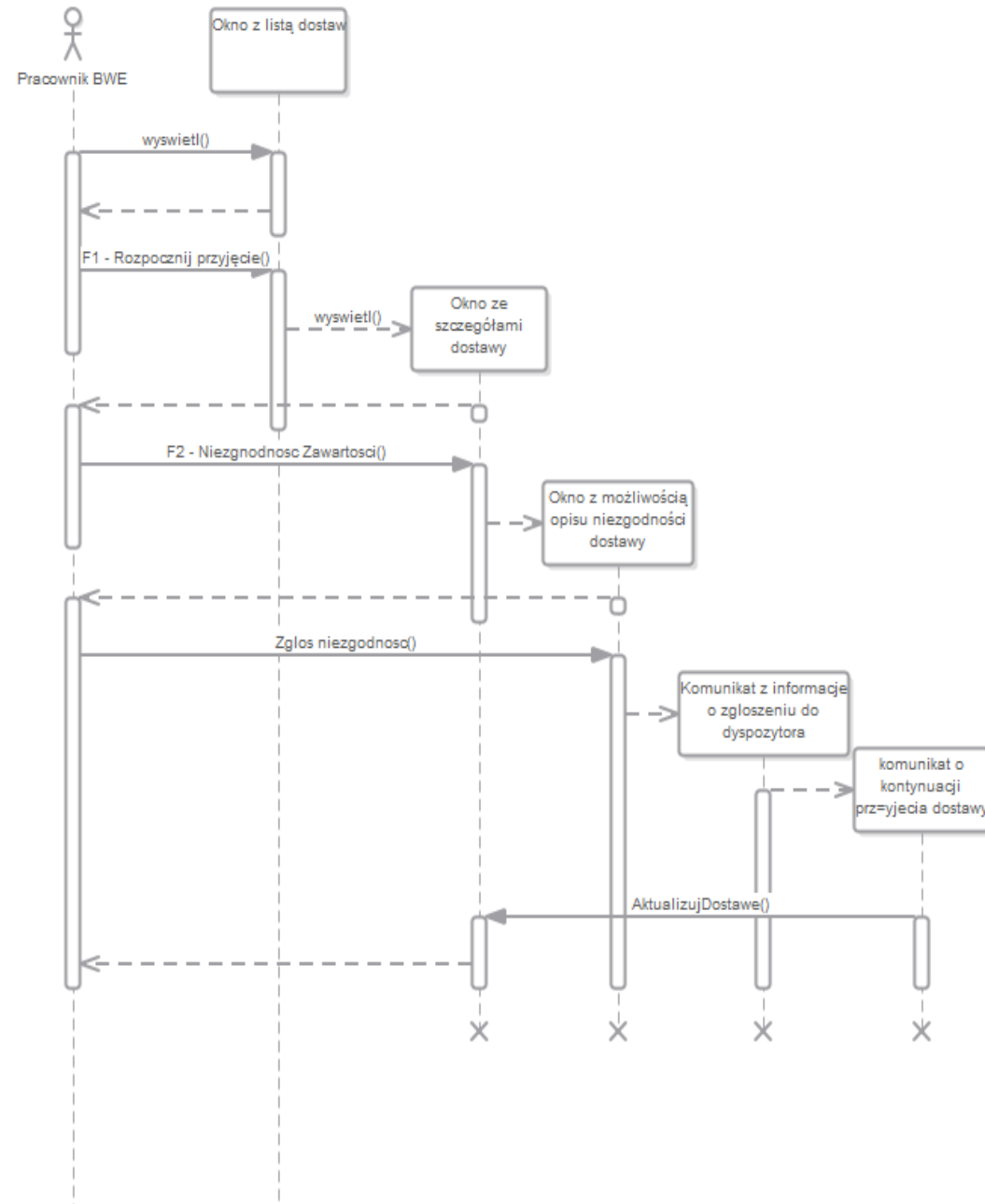
- ...
- 4) **Przepływ Zadań (C)** - sprecyzuj dokładnie **interakcję** użytkownika z systemem oraz **cykl życia** poszczególnych ekranów w postaci diagramów sekwencji,
- 5) **Stany Obiektów (C)** - co dzieje się danymi obiektami w trakcie interakcji z użytkownikiem – udokumentuj wszystkie możliwe stany jakie może przyjmować obiekt dziedziny, aby móc zaprojektować ich wizualną reprezentację!



# Projekt sekwencyjny (C)

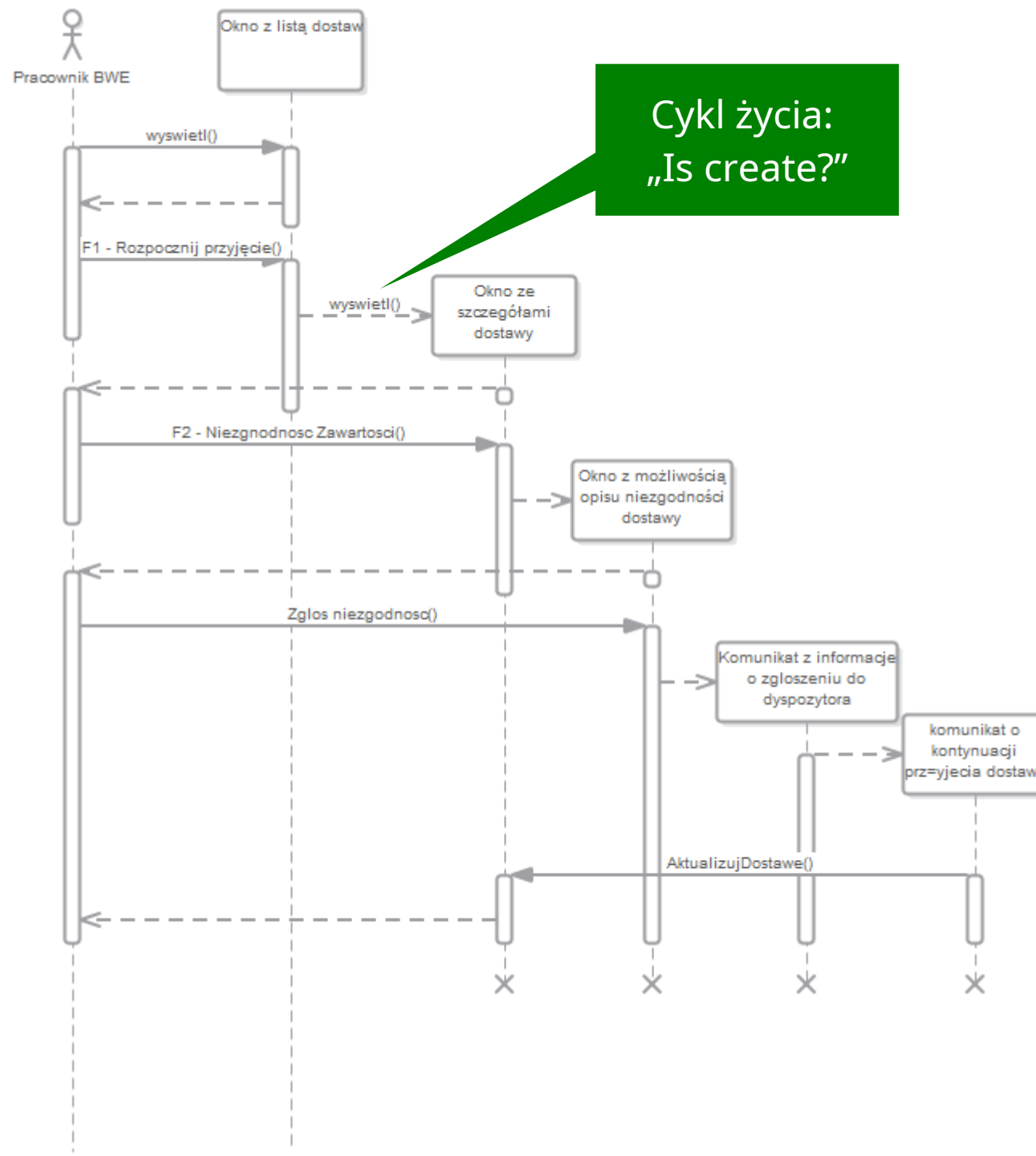
Definiuje kolejność operacji wprowadzania danych wejściowych i wyjściowych.

- 1) **Przepływ Zadań (C)** -  
sprecyzuj dokładnie interakcję  
użytkownika z systemem oraz  
cykl życia poszczególnych ekranów
- 2) Stany Obiektów - co dzieje się  
danymi obiektami w trakcie  
interakcji z użytkownikiem



# Przeptyw zadań – diagram sekwencji

- W diagramie przepływu zadań stworzymy szczegółowy projekt interakcji użytkownika systemu z poszczególnymi oknami oraz widokami.
- Motywacja:
  - Dokumentacja ta będzie służyć do implementacji prototypu oraz gotowego systemu
  - Dokumentacja będzie dodatkowo służyć do definiowania automatycznych testów interfejsu użytkownika.
  - Dokumentacja ta będzie pomocna w Help Desku w przydziale osób odpowiedzialnych za usunięcie usterek lub określeniu scenariusza i ekranów, przez które przechodził użytkownik..
- Wykorzystujemy diagram sekwencji!



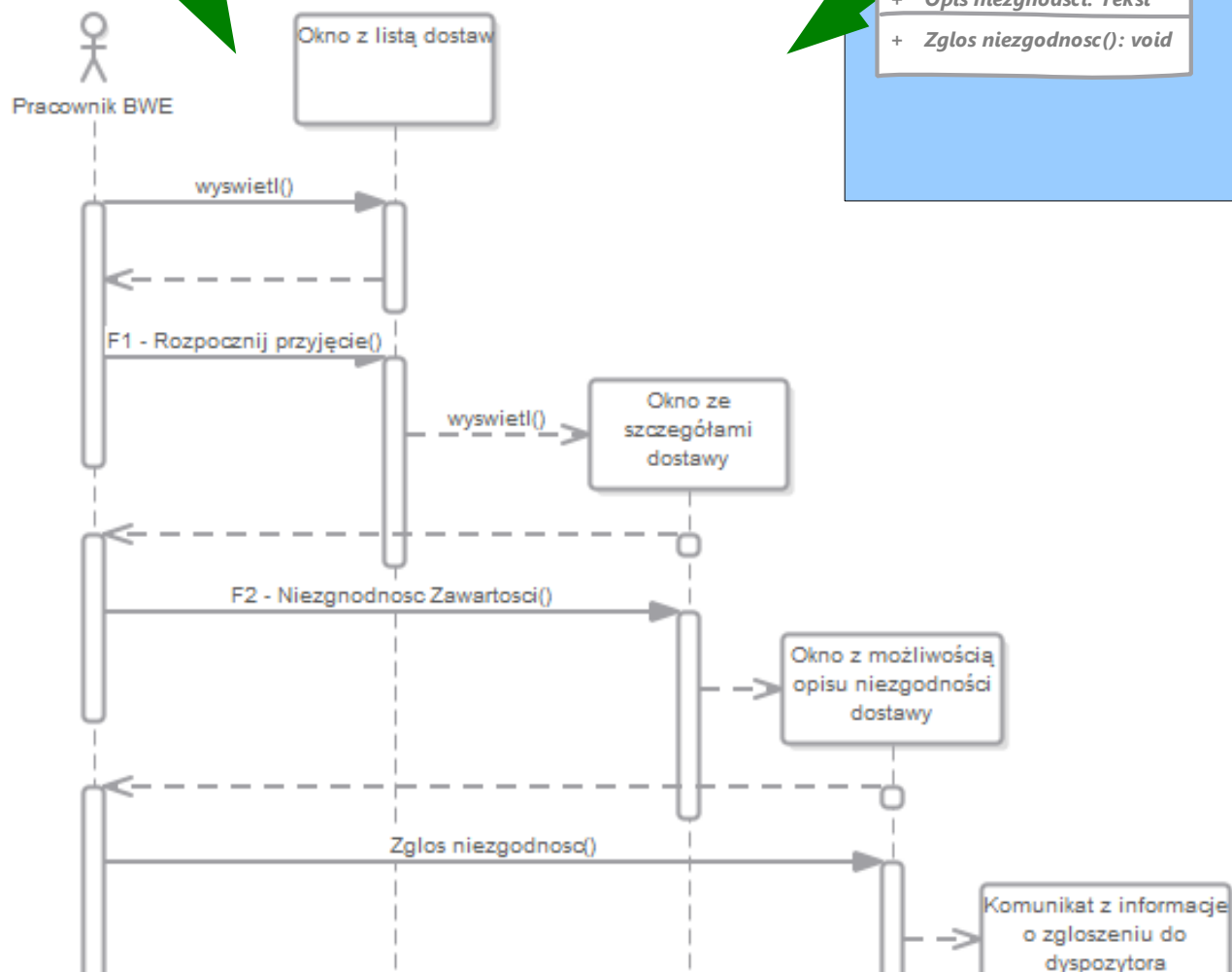
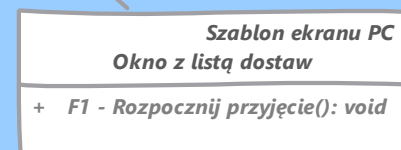
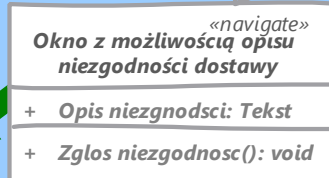
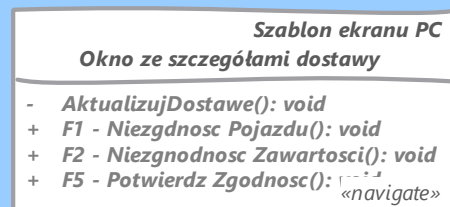


# Połączenie scenariusza ze strukturą

## Przegląd zadań (scenariusz):

- System wyświetla [okno z listą dostaw].
- Użytkownik wybiera [Rozpocznij przyjęcie]
- System wyświetla [okno ze szczegółami dostawy].
- Użytkownik wybiera [Niezgodność Zawartości].

## Projekt abstrakcyjny UI:

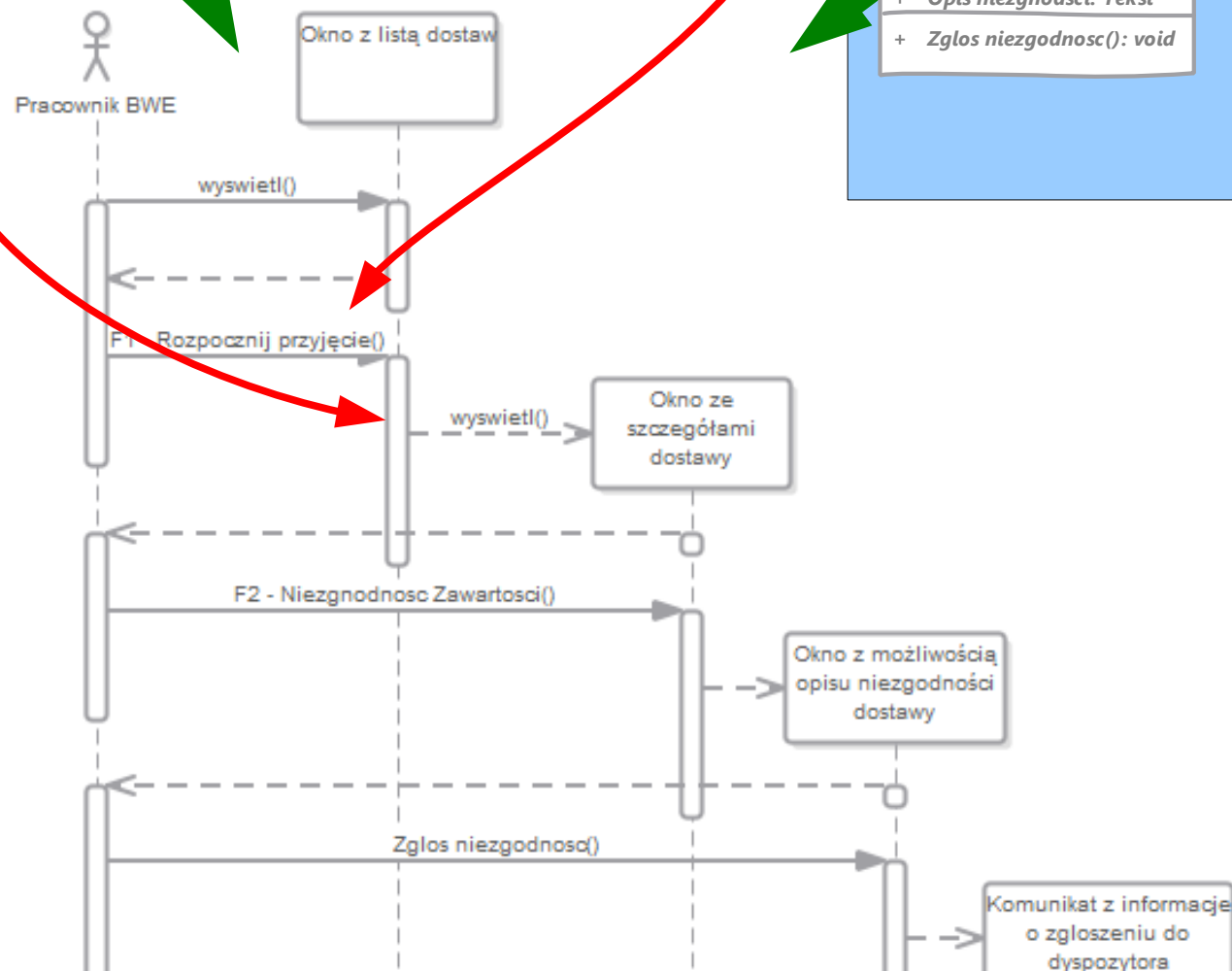
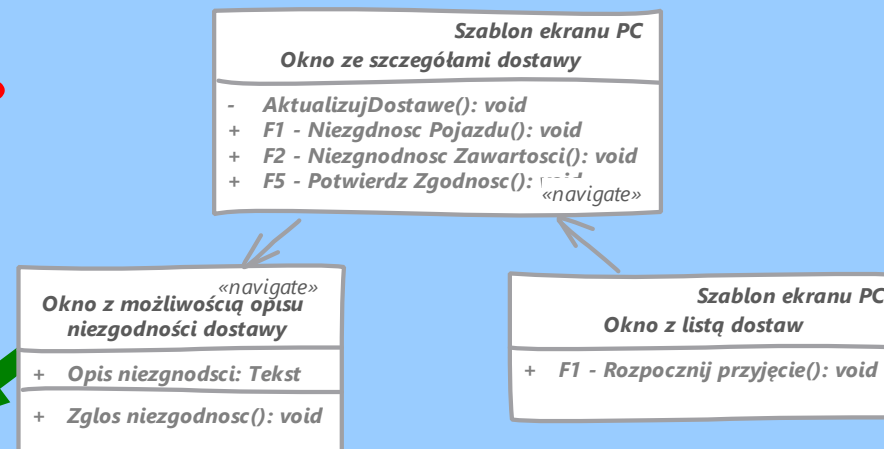


# Połączenie scenariusza ze strukturą

## Przegląd zadań (scenariusz):

- System wyświetla [okno z listą dostaw].
- Użytkownik wybiera [Rozpocznij przyjęcie]
- System wyświetla [okno ze szczegółami dostawy].
- Użytkownik wybiera [Niezgoda Zawartości].

## Projekt abstrakcyjny UI:

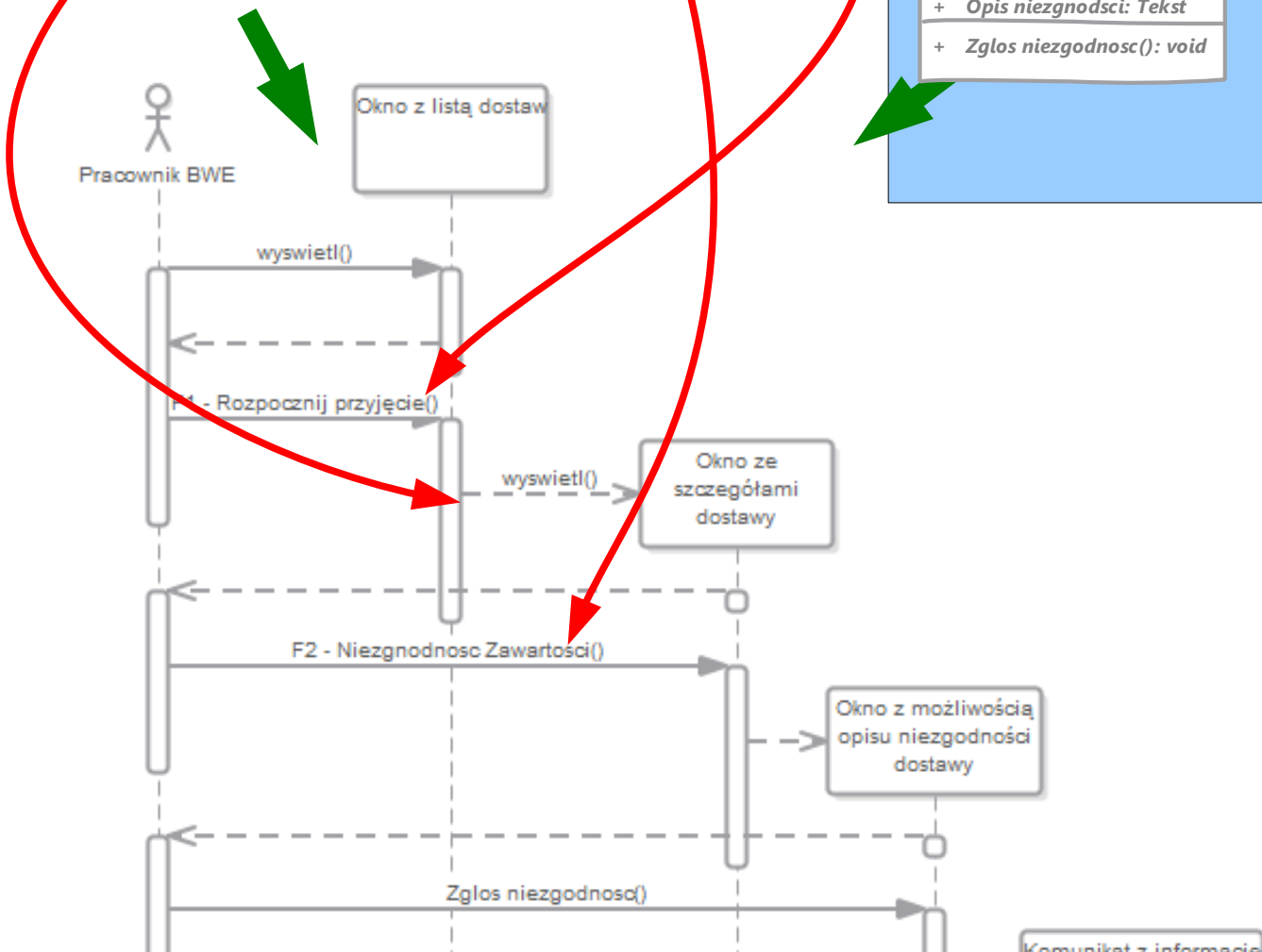
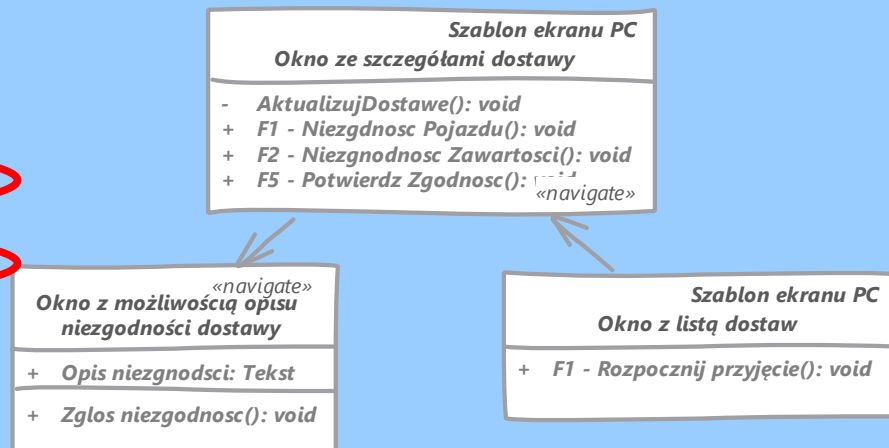


# Połączenie scenariusza ze strukturą

## Przegląd zadań (scenariusz):

- System wyświetla [okno z listą dostaw].
- Użytkownik wybiera [Rozpocznij przyjęcie]
- System wyświetla [okno ze szczegółami dostawy].
- Użytkownik wybiera [Niezgodność Zawartości].

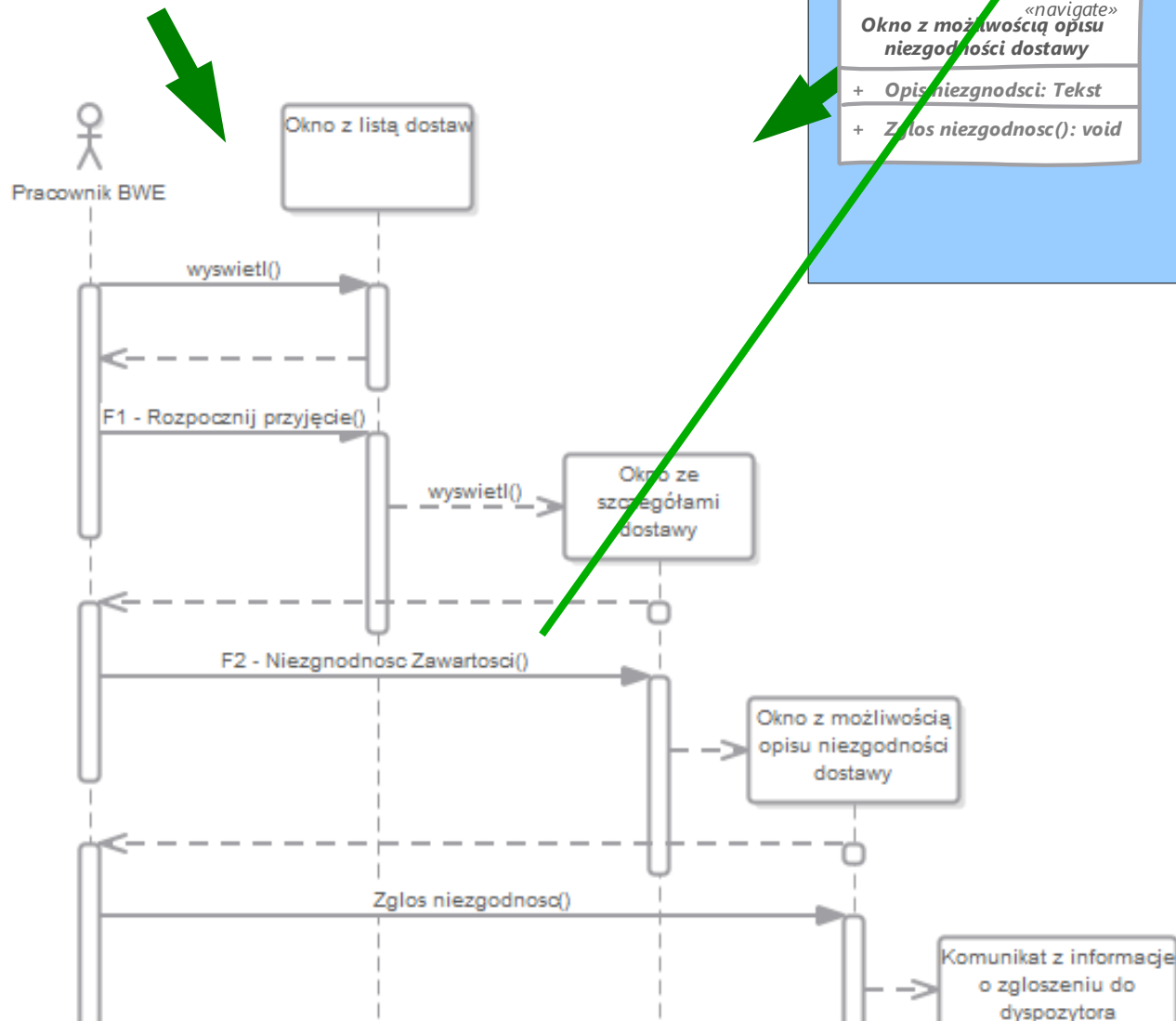
## Projekt abstrakcyjny UI:



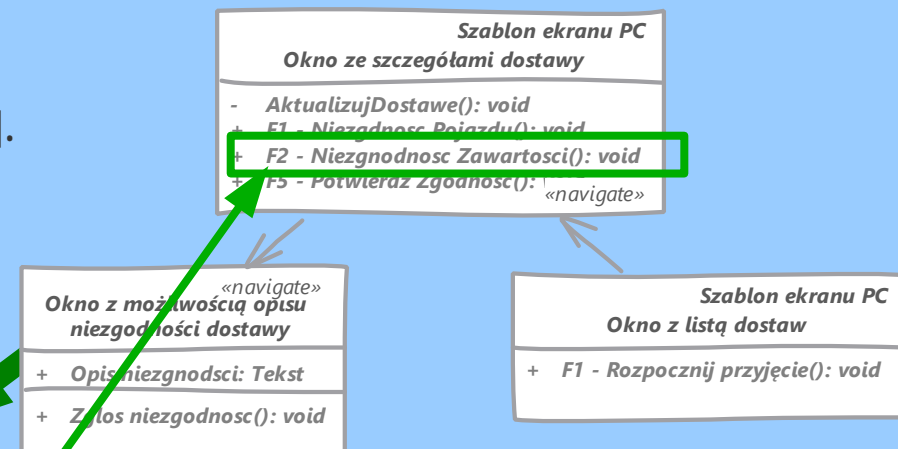
# Połączenie scenariusza ze strukturą

## Przegląd zadań (scenariusz):

- System wyświetla [okno z listą dostaw].
- Użytkownik wybiera [Rozpocznij przyjęcie]
- System wyświetla [okno ze szczegółami dostawy].
- Użytkownik wybiera [Niezgodność Zawartości].
- ...



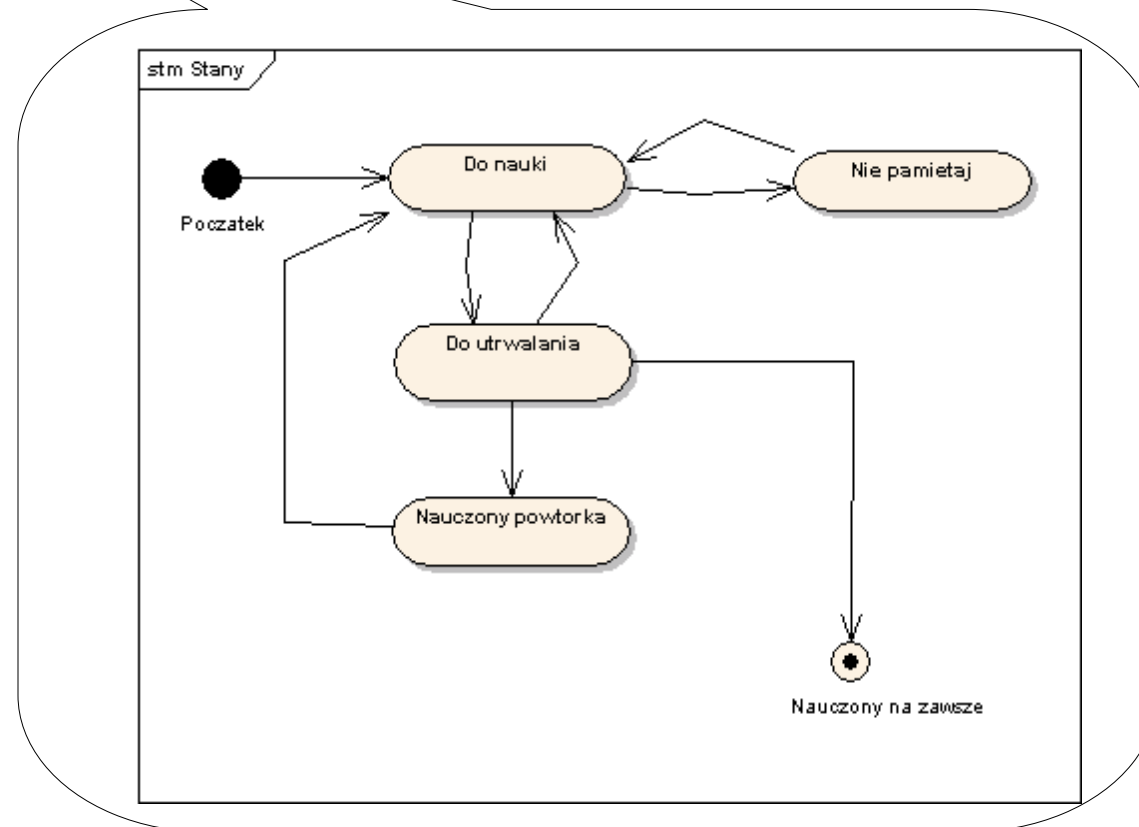
## Projekt abstrakcyjny UI:



# Projekt sekwencyjny – Stany obiektów (C)

Definiuje kolejność operacji wprowadzania danych wejściowych i wyjściowych.

- 1) Przepływ Zadań - sprecyzuj dokładnie interakcję użytkownika z systemem
- 2) **Stany Obiektów** - co dzieje się z danymi obiektami w trakcie interakcji z użytkownikiem

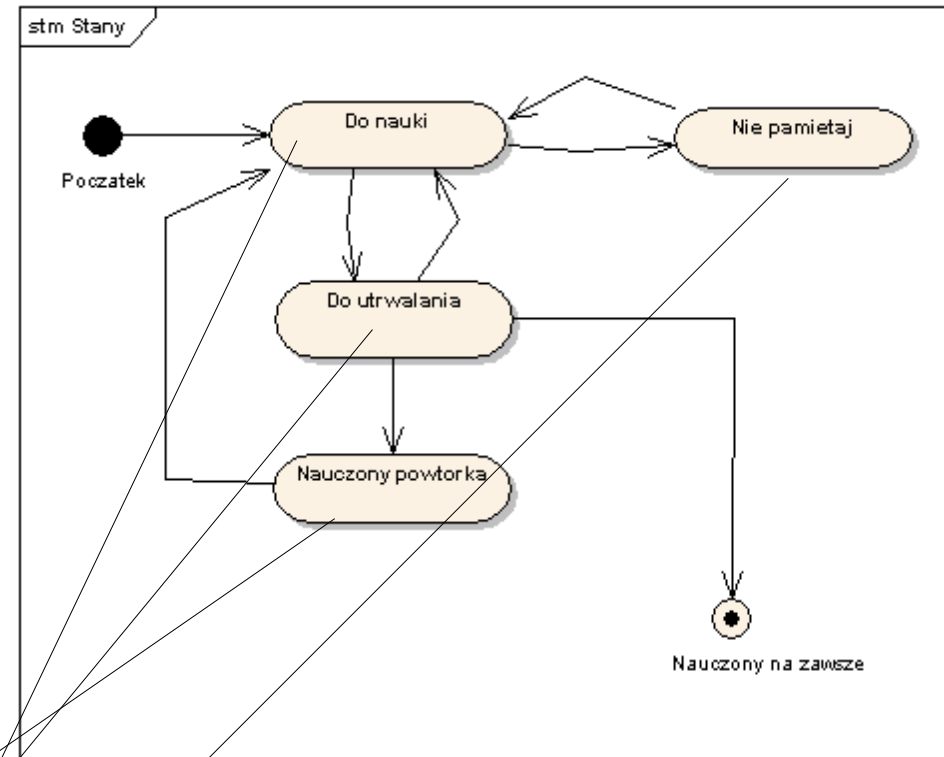




# Stany obiektów

- Model stanów obiektów wykorzystujemy do dokumentacji wizualnej reprezentacji stanu obiektu.
- Etap ten jest związany z zaleceniem projektowym dotyczącym wyraźnej reprezentacji poszczególnych stanów.

Okno: Lista kart			
Twoje karty:			
1	to break	łamać	(powt.)
2	bread	chleb	(powt.)
3	to teach	uczyć kogoś	(utrwal)
4	to lie	kłamać	(utrwal)
5	to sleep	spać	(nauka)
6	cookie	ciastko	(nauka)
7	to learn	uczyć się	(nie powt)
Zamknij			



Zwróćmy uwagę na dwukanałową Reprezentację stanu obiektu:

- Kolor tła wiersza,
- Etykieta w ostatniej kolumnie.



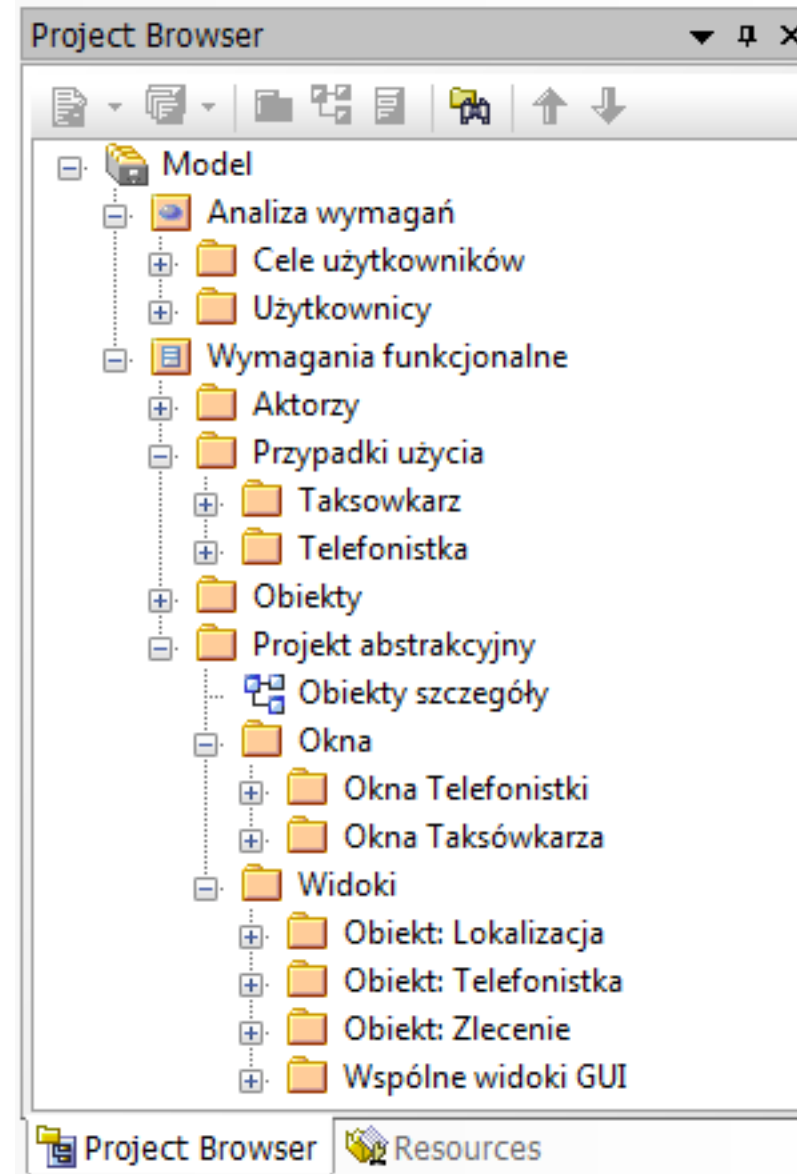
# Podsumowanie UML - interfejs użytkownika

- **Analiza wymagań**

- Diagram aktorów (ról)
- Diagramy zadań (diagram „notatek”)
- Diagramy scenariuszy zadań (diagram czynności)
- Diagram przypadków użycia

- **Model interfejsu - specyfikacja funkcjonalna**

- Scenariusze przypadków użycia (diagramy czynności)
- Model obiektów dziedziny użytkownika (diagramy klas)
- Widoki i okna abstrakcyjne (diagramy klas)
- Model sekwencji interakcji (diagramy sekwencji)
- Model stanów obiektów dziedziny (diagram stanów)



## Zgodność z projektem abstrakcyjnym

- Wszystkie elementy i decyzje podjęte w fazie projektu abstrakcyjnego powinny mieć swój odpowiednik w projekcie wizualnym.
- Nie powinno się robić żadnych rozszerzeń.

## Czytelność reprezentowanych obiektów

- Obiekty, wyodrębnione w fazie abstrakcyjnej powinny być w czytelny sposób reprezentowane dla użytkownika.
- Obiekty powinny być łatwo rozpoznawalne przez użytkownika na wszystkich etapach procesu używania systemu (np. Etykiety w nagłówkach paneli widoków).
- W jasny i czytelny sposób powinny być reprezentowane aktualne stany obiektów oraz ich zmiany (kolory oraz kody stanów).

Zgodność i czytelność mają bezpośredni wpływ na użyteczność systemu.



# Przykład widoku obiektu dziedzinowego - konsultacje

«obiekt dziedziny»

Konsultacje

+ lokalizacja: sala

+ powiadom\_o\_zgloszeniu: boolean

+ pracownik: osoba

+ rodzaj: int

+ termin: int

+ termin\_waznosc\_do: Data

+ termin\_waznosc\_od: Data

+ uwagi: String

+ wizyta\_u\_dziekana: boolean

+ wymagane\_zgloszenia: boolean

+ zgloszenia\_godziny\_przed: int

+ zgloszenia\_limit: int

Wtorek, w godzinach: 14:15 - 15:00, Gmach Elektrotechniki pokój 223  
w okresie 18.02.2019 - 12.06.2019 , na studiach stacjonarnych (wymagane zapisy - na najbliższy termin zajęte 1 zgłoszenia z dostępnych 4, zgłoszenia minimum 14 godzin przed terminem ([wyświetl zgłoszenia](#))) ([Zmień](#))

Jednostka organizacyjna: ZETiS

Semestr: 2019L

Tryb studiów: stacjonarne

[Wyczyść filtr](#)

Szukaj

[Drukuj Zamiast](#)

Godziny konsultacji

Zakład Elektrotechniki Teoretycznej i Informatyki Stosowanej

w semestrze 2019L

Imię i nazwisko	Konsultacje				Uwagi
	Pokój	Dzień tygodnia	Godziny	Tryb studiów	
Basinski Radosław mgr inż.				-	
Bołkowski Stanisław prof. dr hab.				-	
Brociek Wiesław doc. dr inż.				-	
Chaber Bartosz dr inż.	GE-b 223	Poniedziałek	10:15-11:00	stacjo.	
Chojnowski Michał mgr inż.				-	
Cichocki Andrzej prof. dr hab. inż.				-	

Konsultacje w 2019L:

[Pokaż/ukryj archiwalne](#)

Wtorek, w godzinach: 14:15 - 15:00, Gmach Elektrotechniki pokój 223

w okresie 18.02.2019 - 12.06.2019 , na studiach stacjonarnych (wymagane zapisy - na najbliższy termin zajęte 1 zgłoszenia z dostępnych 4, zgłoszenia minimum 14 godzin przed terminem )

Zapisz się na termin

Wtorek, w godzinach: 14:15 - 15:00, Gmach Elektrotechniki pokój 223  
w okresie 18.02.2019 - 12.06.2019 , na studiach stacjonarnych (wymagane zapisy - na najbliższy termin zajęte 1 zgłoszenia z dostępnych 4, zgłoszenia minimum 14 godzin przed terminem ([wyświetl zgłoszenia](#))) ([Zmień](#))

Zgłoszenia na konsultacje w dniu: 12.03.2019 ([e-mail do wszystkich](#))

[\(Wyświetl historyczne zgłoszenia\)](#)

1. (nr 302585 - Yakubov Zafar

Temat spotkania: Consultation and the discussion of some questions from the course Numerical Methods.

1) (Zarejestrowano: 7.03.2019 21:59)

[\(Zamknij\)](#)

Piątek, w godzinach: 09:15 - 10:00, Gmach Elektrotechniki pokój 223

Tryb prowadzenia zajęć: \* stacjonarne

Wizyta u Dziekana? ☐ (Tylko zaznaczone terminy będą wyświetlać się w portalu.)

Termin: \* Wtorek godzina od 14:15 (hh:mm) godzina do 15:00 (hh:mm)

Termin obowiązuje w okresie od: \* 18.02.2019 do 12.06.2019 (cały semestr)

Miejsce konsultacji: \* Gmach Elektrotechniki pokój 223

Uwagi:

Wymagana rezerwacja terminu: ☒ (Znacznik wskazujący czy studenci przed wizytą muszą zgłosić się na termin w ISODzie.)

Maksymalna liczba zgłoszeń na termin: \* 4 (Tylko tyle osób będzie mogło zgłosić się w jednym terminie.)

Zgłoszenia co najmniej godzin przed terminem: \* 14 (Minimalna liczba godzin przed terminem konsultacji kiedy mogą zgłaszać się studenci.)

Powiadom mnie e-mailem o nowym zgłoszeniu: ☒

Zapisz Usun (Anuluj)

iCal

Piątek

owanie

ów

3

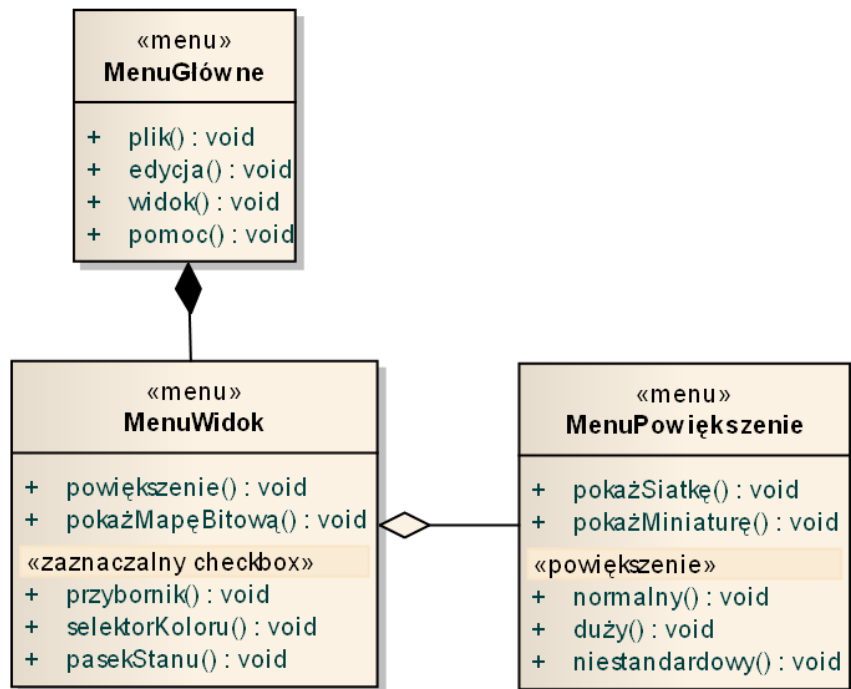
Konsultacje 9:15-10:00 w okresie 18.02.2019 - 12.06.2019 (wymagane zapisy) , w sali GE-b 223

1DA2223-A - Python programming and data analysis

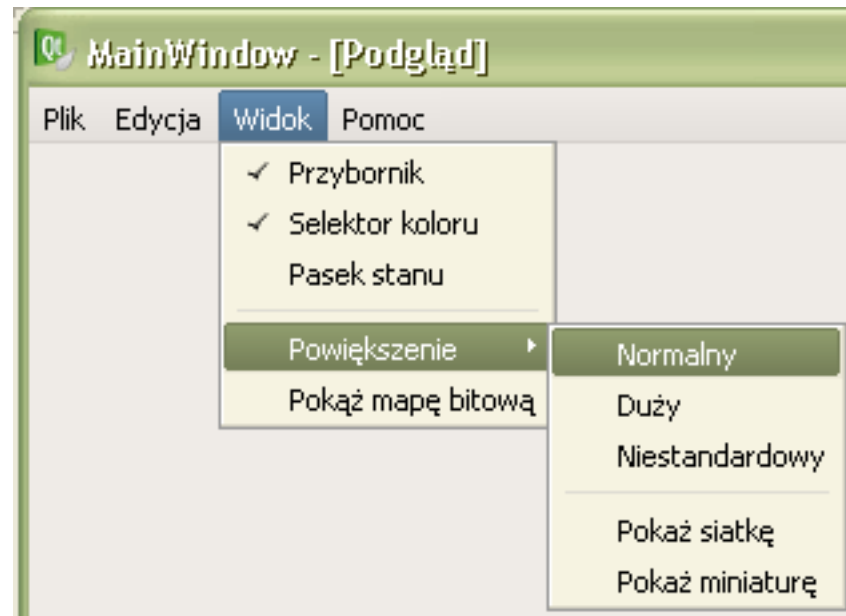


# Przykład projektu wizualnego Menu

Przykładowy projekt wizualny dla projektu funkcjonalnego menu głównego aplikacji



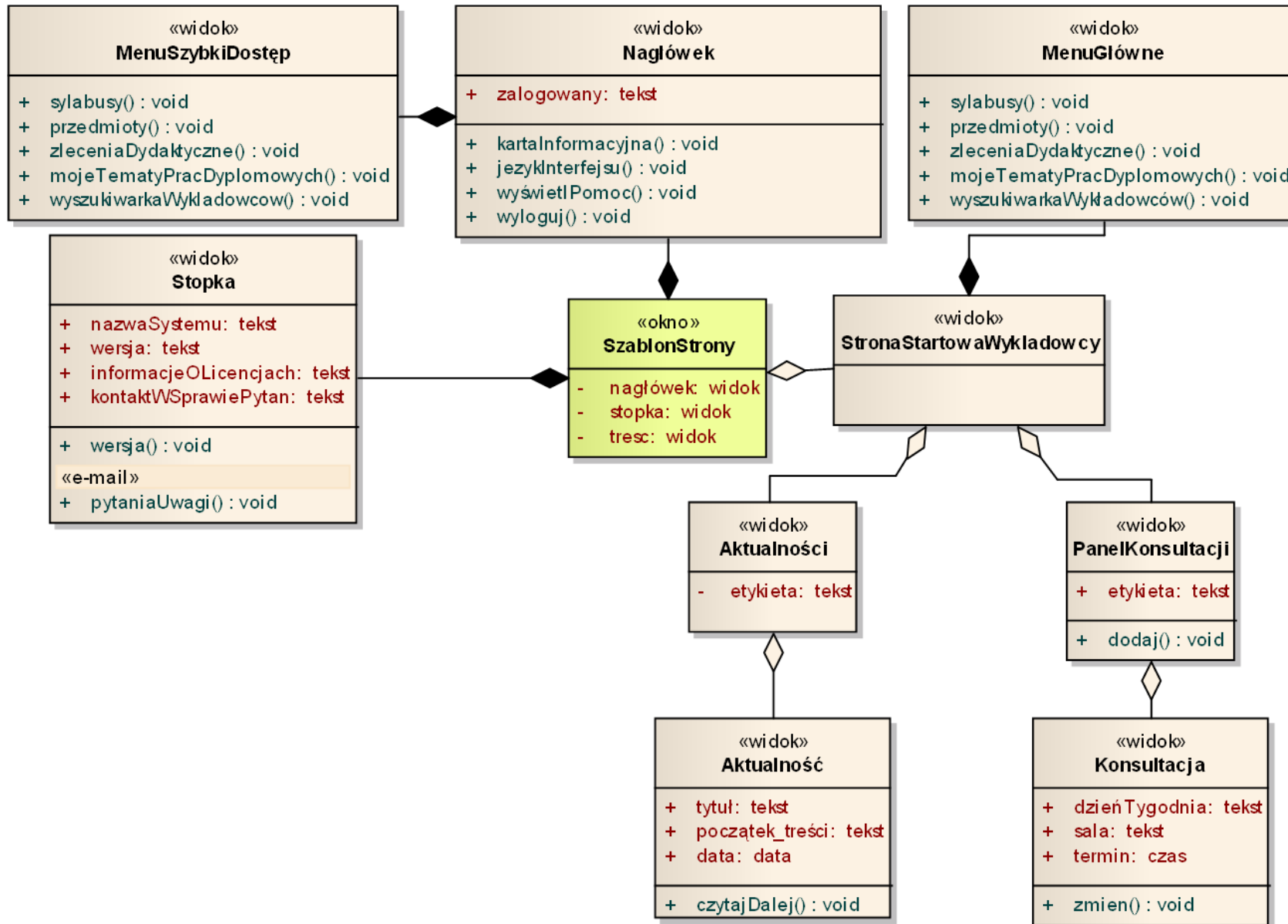
projekt funkcjonalny



projekt wizualny






# Przykład projektu wizualnego – struktura logiczna



# Przykład projektu wizualnego – realizacja graficzna

Zalogowany jako:  [prof. Jan Kowalski](#) [English](#)  [Pomoc](#) [Wyloguj](#)

## Czynności

 <a href="#">Sylabusy</a> Zobacz i edytuj dostępne sylabusy	 <a href="#">Przedmioty</a> Zobacz katalog przedmiotów	 <a href="#">Zlecenia dydaktyczne</a> Zarządzaj zleceniami dydaktycznymi	 <a href="#">Moje tematy prac dyplomowych</a> Modyfikuj i dodaj nowych tematów prac dyplomowych	 <a href="#">Wyszukiwarka wykładowców</a> Sprawdź dane kontaktowe, terminy konsultacji, tematy prac
--	---	---	--	--

## Aktualności

### ● Słownik pojęć Wirtualnego Dziekanatu

Przedmiot – jest to ...

[\(czytaj dalej\)](#)

18.03.2010

## Moje konsultacje w 2010 L

Wtorek, Gmach Elektrotechniki 23, 14:00 - 15:00, [\(Zmień\)](#)

Środa, Gmach Mechaniki 234, 14:20 - 16:00, [\(Zmień\)](#)

[\(Dodaj\)](#)

Internetowy System Obsługi Dziekanatu 2010 (wersja: [1.1.2](#)) | Icons by <http://dryicons.com>

Uwagi oraz pytania: [isod.wsparcie@iem.pw.edu.pl](mailto:isod.wsparcie@iem.pw.edu.pl)



# Przykład projektu wizualnego – zaznaczone widoki

## Strona startowa wykładowcy

Menu szybki dostęp



Zalogowany jako: [prof. Jan Kowalski](#)

[English](#) [Pomoc](#) [Wyloguj](#)

Nagłówek

### Czynności



[Sylabusy](#)

Zobacz i edytuj dostępne sylabusy



[Przedmioty](#)

Zobacz katalog przedmiotów



[Zlecenia dydaktyczne](#)

Zarządzaj zleceniami dydaktycznymi



[Moje tematy prac dyplomowych](#)

Modyfikuj i dodaj nowych tematów prac dyplomowych



[Wyszukiwarka wykładowców](#)

Sprawdź dane kontaktowe, terminy konsultacji, tematy prac

Menu główne

Aktualności

### Aktualności

Aktualność

● **Słownik pojęć Wirtualnego Dziekanatu**

Przedmiot – jest to ...

[\(czytaj dalej\)](#)

18.03.2010

Panel konsultacji

### Moje konsultacje w 2010 L

Konsultacja

Wtorek, Gmach Elektrotechniki 23, 14:00 - 15:00, [\(Zmień\)](#)

Środa, Gmach Mechaniki 234, 14:20 - 16:00, [\(Zmień\)](#)

[\(Dodaj\)](#)

Stopka

Internetowy System Obsługi Dziekanatu 2010 (wersja: [1.1.2](#)) | Icons by <http://dryicons.com>

Uwagi oraz pytania: [isod.wsparcie@iem.pw.edu.pl](mailto:isod.wsparcie@iem.pw.edu.pl)



Całkowity odbiór aplikacji przez użytkownika musi uwzględniać:

- odpowiednie diagramy sekwencji,
- być zgodnym ze scenariuszami przypadków użycia, o ile były stworzone w wymaganiach,
- o ile to możliwe pozwalać na interakcję (prototypy),
- uwzględniać możliwość występowania kilku wersji do wyboru przez użytkownika,

Jest realizowany za pomocą:

- scenopisów (np. „storyborads”)
  - dokumenty tekstowe
  - prezentacja OpenOffice Impress lub PowerPoint itp.
- prototypów



## Dziękuję za uwagę.

Chcemy być coraz lepsi! Masz uwagi? Napisz proszę e-maila:  
[robert.szmurlo@ee.pw.edu.pl](mailto:robert.szmurlo@ee.pw.edu.pl)

