

# Projektowanie obiektowe oprogramowania

## Wykład 1 - Unified Process

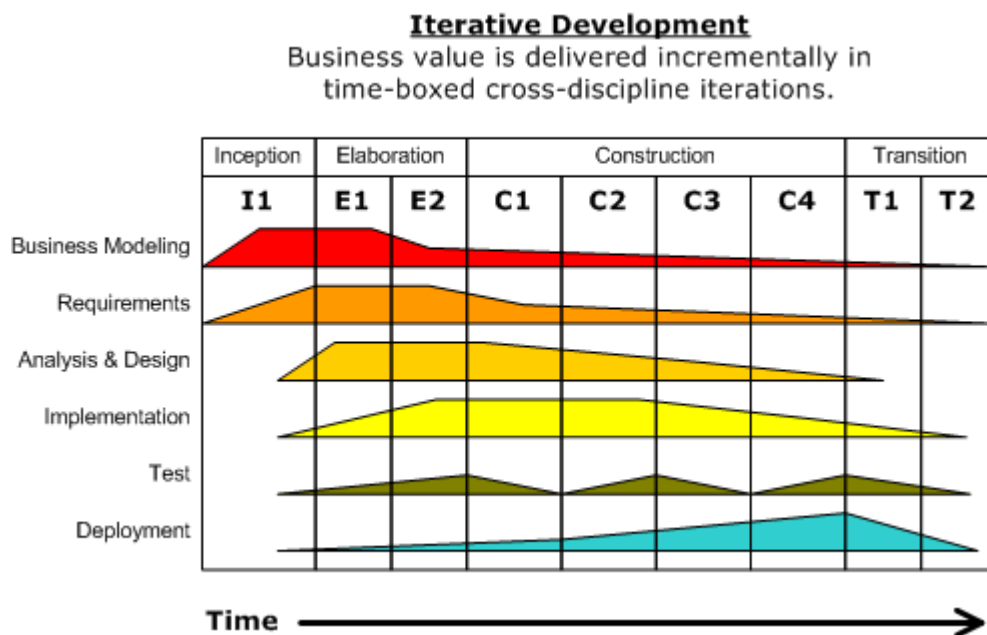
### Wiktor Zychla 2022

---

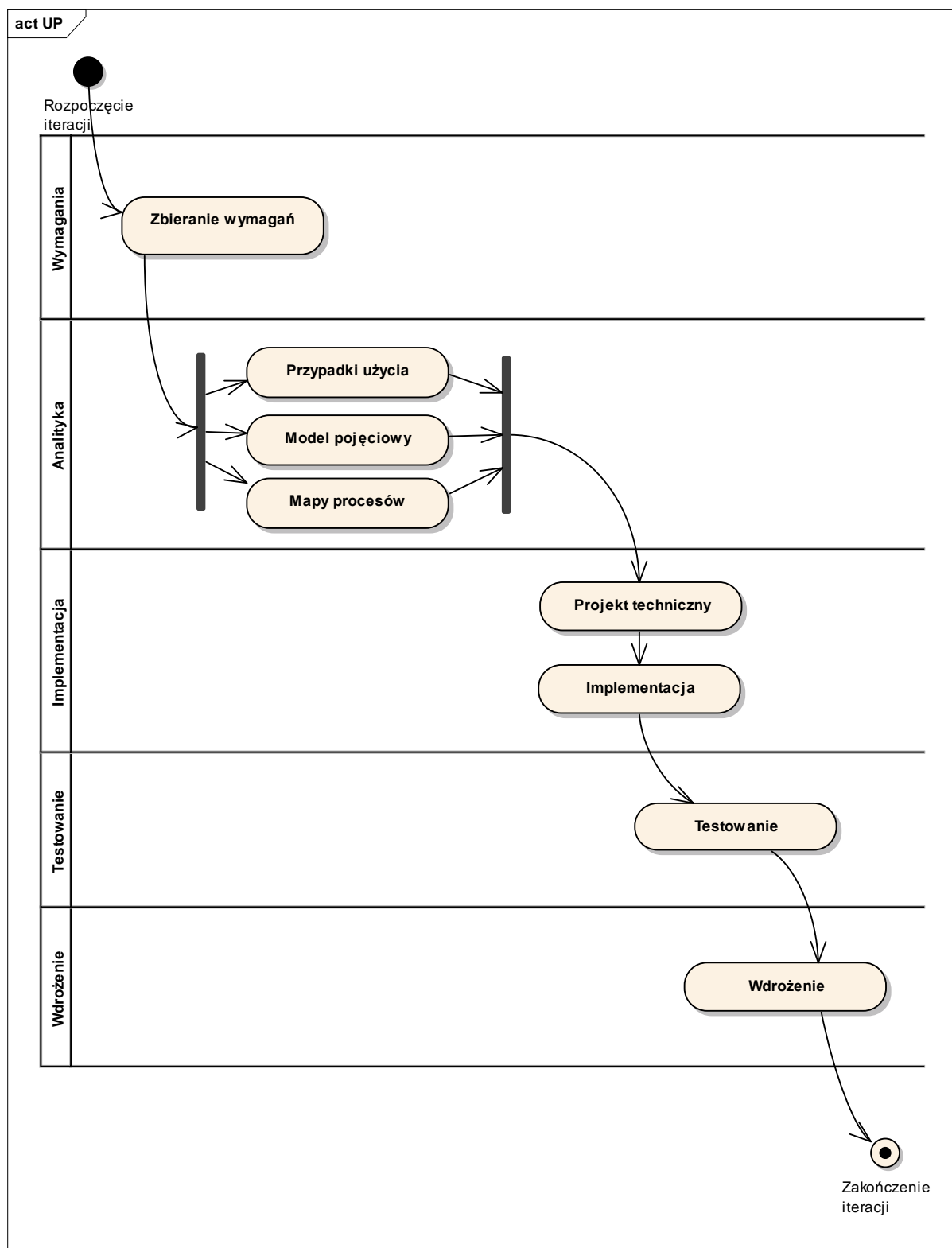
#### 1 Unified Process

Rama organizacji procesu wytwarzania oprogramowania. Posiada wyodrębnione fazy inicjowania, projektowania, implementowania, testowania i wdrażania.

Na niemal wszystkie metodyki wytwarzania oprogramowania można patrzeć jak na warianty UP.



Rysunek 1 <http://upload.wikimedia.org/wikipedia/commons/0/05/Development-iterative.gif>



Rysunek 2 Diagram procesowy UP

## 2 Faza rozpoczęcia (Business Modelling)

*Definicja.* **Faza rozpoczęcia** = określenie zakresu, wizji i uwarunkowań biznesowych.

Typowe artefakty:

Wizja i analiza biznesowa	Opis <b>celów</b> i uwarunkowań biznesowych
Słowniczek	Podstawowa terminologia dziedzinowa
Prototyp	Udowodnienie poprawności rozwiązań technicznych; doprecyzowanie wizji
Plan pierwszej iteracji	
Specyfikacja dodatkowa	Lista dodatkowych wymagań mających istotny wpływ na architekturę
Plan zarządzania ryzykiem	(aktualizowane na bieżąco) scenariusze alternatywne – biznesowe, technologiczne, organizacyjne

## 3 Zbieranie wymagań (Requirements)

### 3.1 FURPS+ - obszary wymagań

*Definicja.* **Wymagania** = zdolności które system musi posiadać i ograniczenia do których musi się dostosować.

[FURPS](#)

[Zwięzły opis](#)

[Przykładowy kwestionariusz wymagań FURPS](#)

Functional	Funkcjonalności, możliwości, bezpieczeństwo
Usability	Czynnik ludzki, pomoc, dokumentacja
Reliability	Awaryjność, odzyskiwanie, przewidywalność
Performance	Czas reakcji, przepustowość, dokładność, dostępność, wykorzystanie zasobów
Supportability	Dostosowanie, utrzymanie, konfiguracja, lokalizacja
Design	Wszelkie ograniczenia projektowe (np. relacyjna baza danych)
Implementation	Narzędzia, sprzęt, zasoby, standardy
Interface	Interfejsy zewnętrznych systemów
Physical	

Typowe problemy:

1. *Shopping cart mentality* – wszystko co tylko można bez świadomości kosztów
2. *All are equal* – brak priorytetyzacji
3. *Requirements can't be measured* – wymagania są niejasne albo niemierzalne

Przykłady wymagań narzuconych przez regulacje prawne:

- [Rozporządzenie o Ochronie Danych Osobowych \(RODO\)](#)
- Wymagania Krajowych Ram Interoperacyjności (KRI), w szczególności WCAG 2.0  
<http://www.dziennikustaw.gov.pl/du/2012/526/1>  
<https://www.w3.org/TR/WCAG20/>

Proszę z ciekawości prześledzić jak bardzo to jest szczegółowo rozpisane, np.:

<https://www.w3.org/WAI/WCAG21/quickref/?showtechniques=131#info-and-relationships>

i wymagania typu H63 (jedno z wielu!)

<https://www.w3.org/TR/WCAG20-TECHS/H63.html>

**SIWZ**, Specyfikacja istotnych warunków zamówienia

## **OPZ** Opis przedmiotu zamówienia

**Demo:** zapytanie do wyszukiwarki „specyfikacja istotnych warunków zamówienia system informatyczny”

### **3.2 S.M.A.R.T. – kryteria oceny wymagań**

Jak ocenić sformułowane wymagania? [S.M.A.R.T](#)

- Szczegółowy/prosty (**S**imple)
- Mierzalny (**M**easurable)
- Osiągalny/atrakcyjny (**A**chievable)
- Istotny/realistyczny (**R**elevant)
- Terminowy (**T**ime-specific)

#### **Specific – 5W**

- **What:** What do I want to accomplish?
- **Why:** Specific reasons, purpose or benefits of accomplishing the goal.
- **Who:** Who is involved?
- **Where:** Identify a location.
- **Which:** Identify requirements and constraints.

#### **Measurable**

- How much?
- How many?
- How will I know when it is accomplished?
- Indicators should be quantifiable

#### **Achievable**

- **How:** How can the goal be accomplished?

#### **Relevant**

- Does this seem worthwhile?
- Is this the right time?
- Does this match our other efforts/needs?
- Are you the right person?
- Is it applicable in current socio- economic- technical environment?

#### **Time-bound**

- When?
- What can I do six months from now?
- What can I do six weeks from now?
- What can I do today?

#### **Przykłady:**

- Oferowany system musi zapewnić dostęp do niego poprzez Internet z dowolnego komputera za pośrednictwem przeglądarki internetowej. Dostęp do systemu nie może być uzależniony od konieczności instalacji na jednostce lokalnej dodatkowego oprogramowania, które nie jest standardowym elementem przeglądarki internetowej.
- Oferowany system musi charakteryzować się dobrą wydajnością i pojemnością.
- System musi zapewniać mechanizmy kontroli dostępu użytkowników do gromadzonych danych oraz realizowanych operacji wraz z ich rejestracją i możliwością odtworzenia historii zmian.
- Średni czas odpowiedzi przy transakcjach bez zapisu informacji do bazy danych nie może przekraczać 5 sek. a czas maksymalny 20 sek.

## 4 Projektowanie analityczne – przypadki użycia (Analysis)

*Definicja.* **Przypadek użycia** = sekwencja prostych kroków opisująca interakcję między aktorem (użytkownikiem) a systemem.

**Uwaga!** Przypadki użycia należą do **wymagań funkcjonalnych**.

**Uwaga!** Przypadki użycia dokumentuje się w postaci tekstu, wspartego opcjonalnie diagramami przypadków użycia UML.

**Aktor** – byt charakteryzujący się zachowaniem (w tym sam system)

**Aktor pierwszoplanowy** – realizuje cele użytkownika (np. kasjer)

**Aktor drugoplanowy** – dostarcza informacji (np. system kart płatniczych) (opis zewnętrznych interfejsów i protokołów)

### 4.1 Dokumentacja:

**Nieformalna** (*brief*) – zwięzłe streszczenie o długości jednego akapitu, podstawowy scenariusz sukcesu

*Obsługa sprzedaży* – klient staje przy kasie z produktami, które chce kupić. Kasjer korzysta z systemu kasowego w celu odnotowania każdego produktu. System wyświetla informacje na temat poszczególnych produktów oraz sumę do zapłaty. Klient podaje dane pozwalające określić sposób płatności. Kasjer przyjmuje zapłatę, system aktualizuje stan magazynu, klient otrzymuje paragon.

**Pełna** (*fully dressed*) – wszystkie kroki i warianty opisane szczegółowo

- **Poziom** – cel użytkownika lub podprocedura
- **Interesariusze** i ich cele
- **Warunki początkowe**
- **Warunki zakończenia** (powodzenia)
- **Główny scenariusz sukcesu** – brak obsługi błędów, wyrażen warunkowych
- **Rozszerzenia** – obsługa błędów i sytuacji nietypowych
- **Kroki:**
  - **Zdarzenie** inicjujące przypadek użycia
  - **Interakcja** między aktorami
  - **Walidacja**
  - **Zmiana stanu** systemu (np. zapisanie danych)
- **Dodatkowe wymagania**
- **Technologia i format danych**

#### 4.1.1 Przykład

<https://jira.atlassian.com/secure/attachment/48985/Use+case+POS.pdf>

#### 4.1.2 Przykład – licytuj towar

**Nazwa:** Licytuj towar

**Numer:** 1

**Twórca:** Jan Kowalski

**Poziom ważności:** Wysoki

**Typ przypadku użycia:** Ogólny, niezbędny

**Aktorzy:** Uczestnik aukcji [kupujący]

**Krótki opis:** Licytacja wskazanego towaru

**Warunki wstępne:** Uczestnik aukcji posiada niezablokowane konto

**Warunki końcowe:** Oferta została zarejestrowana lub wyświetlony został komunikat o błędzie a stan systemu nie uległ zmianie

**Główny scenariusz sukcesu:**

- 1) Uczestnik aukcji wskazuje aukcję, w której chce uczestniczyć
- 2) System wyświetla formularz do wpisania oferty
- 3) Uczestnik aukcji wpisuje ofertę, a następnie wybiera opcję licytuj
- 4a) System rejestruje ofertę i informuje o tym Uczestnika aukcji
- 5) Następuje rozszerzenie aukcji o przypadek Finalizuj transakcję

**Alternatywne przebiegi zdarzeń:**

4b) Jeżeli w kroku 3) Uczestnik aukcji wprowadził kwotę niezgodną z regułami licytacji, system informuje o błędzie i następuje przejście do kroku 2)

**Wyjątki w przepływach**

4c) Jeżeli z powodu awarii technicznej lub zakończenia aukcji system nie może zarejestrować

oferty, informuje o tym Uczestnika aukcji i następuje zakończenie przypadku

**Specjalne wymagania:** brak

**Notatki i kwestie:** Po zakończeniu aukcji system informuje kupującego i sprzedającego o wyniku licytacji W dowolnym momencie Uczestnik aukcji może zrezygnować z licytacji i następuje zakończenie przypadku

#### 4.1.3 Przykład – finalizuj transakcję

**Nazwa:** Finalizuj transakcję

**Numer:** 2

**Twórca:** Jan Kowalski

**Poziom ważności:** Wysoki

**Typ przypadku użycia:** Ogólny, niezbędny

**Aktorzy:** Uczestnik aukcji [kupujący], oraz Uczestnik aukcji [sprzedający]

**Krótki opis:** Finalizacja rozstrzygniętych aukcji

**Warunki wstępne:**

- 1) Uczestnik aukcji posiada niezablokowane konto
- 2) Uczestnik aukcji [sprzedający] był oferentem aukcji
- 3) Uczestnik aukcji [kupujący] wygrał licytację

**Warunki końcowe:**

Transakcja została zakończona lub aukcja została unieważniona

**Główny scenariusz sukcesu:**

- 1) System informuje Uczestników aukcji o zakończeniu licytacji
- 2a) Kupujący określa sposób płatności oraz wybiera formę dostarczenia towaru
- 3) System wysyła do sprzedającego informację o sposobie płatności oraz wybranej przez kupującego formie dostarczenia towaru
- 4) Sprzedający wystawia ocenę kupującemu
- 5) W przypadku negatywnej oceny system wysyła informację do Administratora
- 6) Kupujący wystawia ocenę sprzedającemu
- 7) W przypadku negatywnej oceny system wysyła informację do administratora
- 8) Administrator w przypadku uzasadnionych skarg uczestników transakcji i (lub) naruszenia regulaminu może unieważnić transakcję

**Alternatywne przebiegi zdarzeń:**

2b) Jeżeli w ciągu 3 dni od zawarcia transakcji nie poinformował sprzedawcy o wyborze sposobu

płatności, sprzedawca może unieważnić transakcję

**Specjalne wymagania:** brak



**Notatki i kwestie:** Pomiędzy kolejnymi zdarzeniami mogą wystąpić kilkudniowe odstępy czasowe

Kroki 6) i 7) mogą wystąpić przed krokami 4) i 5)

## 4.2 Poszukiwanie przypadków użycia

Najbardziej oczywiste - określenie aktorów i celów.

Kasjer	<ul style="list-style-type: none"><li>• Przetwarzanie sprzedaży</li><li>• Wkładanie pieniędzy do kasy</li><li>• Wypłacanie pieniędzy z kasy</li><li>• ...</li></ul>
Kierownik	<ul style="list-style-type: none"><li>• Uruchamianie systemu</li><li>• Wyłączanie systemu</li><li>• ...</li></ul>
Administrator systemu	<ul style="list-style-type: none"><li>• Zarządzanie użytkownikami</li><li>• Określanie zasad bezpieczeństwa</li><li>• ...</li></ul>

Dodatkowe przypadki użycia:

1. Kto uruchamia i zatrzymuje system
2. Kto administruje systemem
3. Kto zarządza użytkownikami
4. Czy działanie systemu samoistnie zmienia się z upływem czasu
5. Kto ocenia działanie i wydajność
6. Jak obsługuje się aktualizacje
7. Itd... wynikają wprost z wymagań

## 4.3 Kryteria oceny przypadków użycia

**Test EBP (Elementary Business Process)** – Zadanie wykonywane przez jedną osobę w jednym miejscu i określonym czasie w odpowiedzi na pewne zdarzenie biznesowe. Zadanie prowadzi do uzyskania mierzalnej wartości biznesowej. Po jego wykonaniu dane są w spójnym stanie.

**Test rozmiaru** – intuicyjnie zbyt mały lub zbyt duży

1. Negocjuj umowę z dostawcą – nie przechodzi testu rozmiaru (za duży)
2. Obsłuż zwroty – ok
3. Zaloguj się – nie przechodzi testu rozmiaru (za mały)
4. Przesuń pionek na planszy – nie przechodzi testu rozmiaru (za mały)

## 5 Projektowanie analityczne – modele pojęciowe (Analysis)

*Definicja.* **Model dziedziny** (model pojęciowy, konceptualny) = przedstawienie **pojęć** reprezentujących byty ze świata rzeczywistego („użytkownik”, „drukarka”, „faktura”, „ocena”), istotnych dla danej dziedziny oraz **relacji** między nimi („ma”, „zjada”, „lubi”, „rozpoczyna”).

W podejściu **Domain Drive Design** (DDD), *model dziedziny* jest osią dookoła której zorientowany jest cały proces projektowania. DDD ma cztery podstawowe kategorie konceptualne:

### Context

The setting in which a word or statement appears that determines its meaning;

### Domain

A sphere of knowledge (**ontology**), influence, or activity. The subject area to which the user applies a program is the domain of the software

### Model

A system of abstractions that describes selected aspects of a domain and can be used to solve problems related to that domain

### Ubiquitous Language

A language structured around the **domain model** and used by all team members to connect all the activities of the team with the software.

## 5.1 Tworzenie modelu pojęciowego

### 5.1.1 Metoda „fraz rzeczownikowych”

Wykorzystuje się przypadki użycia, np.

#### Główny scenariusz sukcesu:

- 1) **Uczestnik aukcji** wskazuje **aukcję**, w której chce uczestniczyć
- 2) System wyświetla formularz do wpisania **oferty**
- 3) Uczestnik aukcji wpisuje ofertę, a następnie wybiera **opcję** licytacji
- 4a) System rejestruje ofertę i informuje o tym Uczestnika aukcji
- 5) Następuje rozszerzenie aukcji o przypadek Finalizuj transakcję

#### Główny scenariusz sukcesu:

- 1) System informuje Uczestników aukcji o zakończeniu **licytacji**
- 2a) Kupujący określa sposób **płatności** oraz wybiera formę **dostarczenia towaru**
- 3) System wysyła do sprzedającego **powiadomienie** o sposobie płatności oraz wybranej przez kupującego formie dostarczenia towaru
- 4) Sprzedający wystawia **ocenę** kupującemu
- 5) W przypadku negatywnej oceny system wysyła informację do Administratora
- 6) Kupujący wystawia ocenę sprzedającemu
- 7) W przypadku negatywnej oceny system wysyła informację do administratora
- 8) Administrator w przypadku uzasadnionych **skarg** uczestników transakcji i (lub) naruszenia regulaminu może unieważnić transakcję

### 5.1.2 Metoda „lista kategorii”

Transakcje biznesowe	<b>Sale Payment</b>
----------------------	-------------------------

	<b>Reservation</b>
Przedmioty transakcji	<b>SalesItem</b>
Produkt lub usługa związane z transakcją lub przedmiotem transakcji	<b>Item</b> <b>Flight</b> <b>Seat</b> <b>Meal</b>
Gdzie odnotowywana jest transakcja	<b>Register</b> <b>Ledger</b> <b>FlightManifest</b>
Role osób i organizacji związanych z transakcją, aktorzy przypadków użycia	<b>Cashier</b> <b>Customer</b> <b>Store</b> <b>Passenger</b> <b>Player</b>
Miejsce transakcji lub usługi	<b>Store</b>
Ważne zdarzenia które trzeba pamiętać	<b>Sale</b> <b>Payment</b> <b>Game</b>
Obiekty fizyczne	<b>Board</b> <b>Piece</b> <b>Die</b> <b>Airplane</b> <b>Item</b> <b>Register</b>
Opisy	<b>ProductDescription</b> <b>FlightDescription</b>
Katalogi	<b>ProductCatalog</b> <b>FlightCatalog</b>
Pojemniki	<b>Store</b> <b>Board</b> <b>Airplane</b>
Przedmioty w pojemnikach	<b>Item</b> <b>Square</b> <b>Passenger</b>
Systemy współpracujące	<b>CreditAuthorizationSystem</b>
Dokumenty (finansowe, pracownicze, umowy)	<b>Receipt</b> <b>Log</b> <b>Cash</b> <b>Check</b> <b>Ticket</b>
Harmonogramy, rozkłady, instrukcje	<b>Schedule</b>

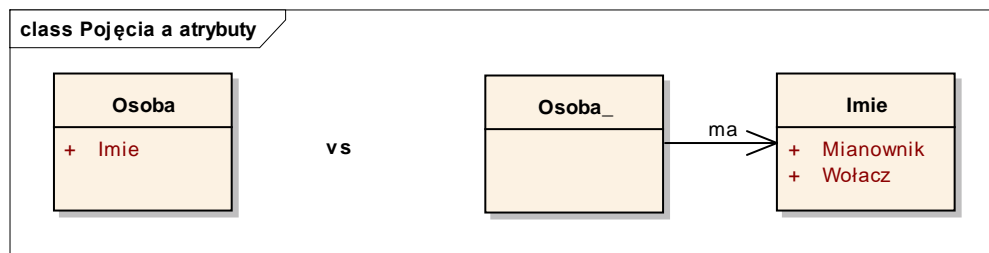
### 5.1.3 Użyj istniejącego modelu

[Fowler - Analysis Patterns](#)

## 5.2 Wskazówki

### 5.2.1 Atrybuty a pojęcia

Jeżeli o X myślimy inaczej niż jako o zmiennej **typu prostego** to prawdopodobnie X jest pojęciem a nie atrybutem.



Pytania pomocnicze:

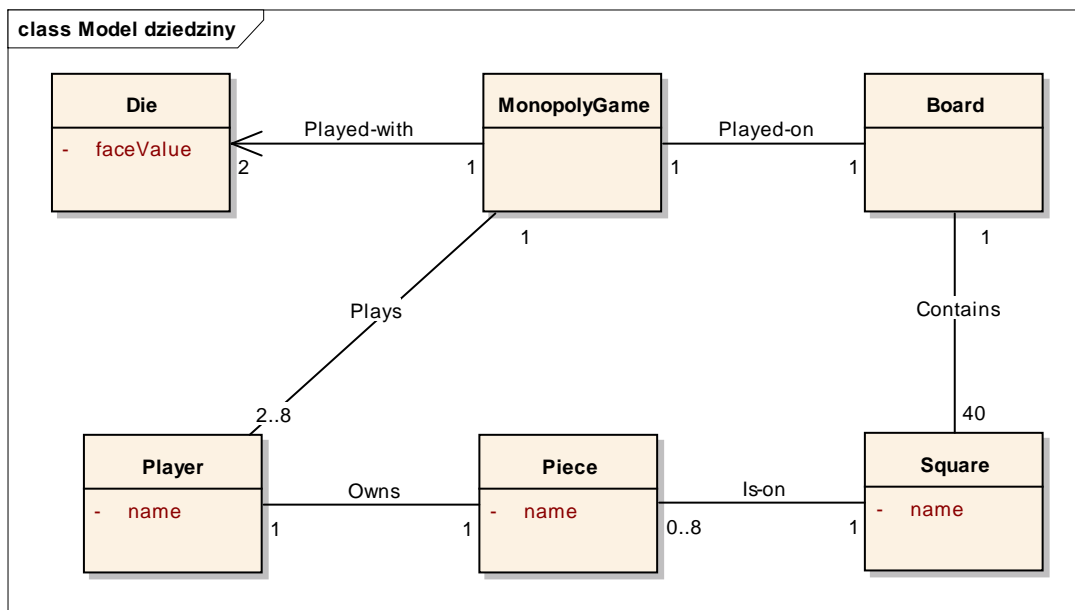
1. Czy dane wewnątrz atrybutu składają się z rozłącznych sekcji (dane osobowe, adres)?
2. Czy z atrybutem wiążą się jakieś operacje? (walidacja)
3. Czy atrybut sam ma atrybuty? (cena promocyjna od-do)

### 5.3 Asocjacje

<b>A jest transakcją związaną z inną transakcją B</b>	CashPayment – Sale (płatność – sprzedaż) Cancellation – Reservation
<b>A jest składnikiem transakcji B</b>	SalesLineItem – Sale (skasowany produkt – sprzedaż)
<b>A jest produktem lub usługą sprzedawaną w transakcji B</b>	Flight – Reservation
<b>A jest rolą związaną z transakcją B</b>	Customer – Payment Passenger – Ticket
<b>A jest fizyczną lub logiczną częścią B</b>	Square – Board Seat – Airplane
<b>A jest fizyczne lub logicznie zawarte w B</b>	Passenger – Plane Register – Store (kasa – sklep)
<b>A jest opisem B</b>	FlightDescription – Flight
<b>A jest zapisane/znane/odnotowane w B</b>	Sale – Register (sprzedaż – kasa)
<b>A jest członkiem B</b>	Player – Game Pilot – Airline
<b>A jest organizacyjną podjednostką B</b>	Department – Store
<b>A wykorzystuje/zarządza/posiada B</b>	Player – Piece
<b>A jest obok B</b>	Square – Square

### 5.4 Przykład z podręcznika

Do reprezentacji modelu pojęciowego używa się diagramów klas UML (będziemy o tym mówić na kolejnym wykładzie).



**Uwaga!** To nie jest jeszcze diagram klas tylko właśnie diagram modelu pojęciowego. Diagram modelu pojęciowego próbuje jedynie uchwycić pojęcia i relacje między nimi, nie bardzo przejmując się tym czy i jak z **pojęć** powstaną **klasy** (obiekty).

Diagram klas z kolei jest elementem projektu architektury i stanowi część prac implementacyjnych i byłby o wiele bardziej szczegółowy. Obejmowałby m.in. kwalifikatory dostępu i akcesory, relacje dziedziczenia i implementowania interfejsów. Z kolei asocjacje na diagramie klas mają inne znaczenie niż na diagramie pojęć: na diagramie pojęć asocjacja określa związek (jakiś) między pojęciami. Na diagramie klas asocjacja oznacza istnienie relacji między klasami (klasa ma pole typu takiego jak klasa na którą wskazuje asocjacja).

## 5.5 Przykład z życia

**Demo:** Fragment rzeczywistego dokumentu analitycznego.

Cechy samego oddziału (nie dziedziczone z grupy oddziałów):

- **rok szkolny otwarcia oddziału** - można to pamiętać jako zwykły rok np. dla roku szkolnego 2007/2008 pamiętać 2007.
- **poziom początkowy** ze słownika (domyślnie 1 w szkołach, w przedszkolach bez domyślnego).
  - dwulatki
  - trzylatki
  - czterolatki
  - pięciolatki
  - 0
  - 1
  - 2
  - 3

- 4
- 5
- 6
- 7
- 8
- 9 (bo mamy 9-letnią szkołę baletową)

nie przewidujemy poziomu mieszanego (tak jest teraz w arkuszu) ponieważ będziemy mieli mechanizm „częściowych” oddziałów. Jak ktoś będzie chciał koniecznie opisać oddział mieszany, to opíše 0.5 oddziału 3-latków i 0.5 oddziału 4-latków.

- **Poziom końcowy** – z tego samego słownika, co powyżej (w podstawówce to będzie najczęściej 6, w trzyletniej, semestralnej szkole dla dorosłych również 6, które jednak w SP oznaczają 6 lat, a w szkole dla dorosłych 6 semestrów, czyli trzy lata)
- **„trwanie poziomów”** – rzadko, ale zdarza się, że oddziały nie zmieniają poziomu z biegiem czasu. Tak się zdarza np. w oddziałach specjalnych, w których uczniowie są 2 lata w jednej klasie, po czym dopiero uzyskują promocję do następnej klasy. Aby w takich sytuacjach nie mieć kłopotu z wyznaczeniem poziomu oddziału i móc zapamiętać, że oddział pozostaje np. na poziomie 1 przez dwa lata należy zapamiętać dla każdego poziomu liczbę okresów jego trwania. Domyślnie okres trwania dla każdego poziomu to 1, ale należy umożliwić dowolną tego zmianę. Wpisanie okresu trwania 100 na poziomie 1 oznacza w praktyce, że oddział nie zmienia nigdy poziomu. Trwanie poziomów inne niż 1 będzie się w praktyce zdarzało bardzo rzadko. Wprowadzanie i modyfikacja tych danych powinna być zatem dość głęboko ukryta, aby nie przeszkadzała typowym użytkownikom (np. po wciśnięciu dodatkowego przycisku umieszczonego przy liście typowych atrybutów oddziału).
- **poziom (klasa)** – Pole wyliczane na życzenie lub automatycznie 1 IX i 1 II (nie ma możliwości, aby poziom oddziału/semestru zmienił się innego dnia). Wykorzystywany na zestawieniach i w budowie widocznego identyfikatora oddziału. Poziom nieokreślony mógłby być wyróżnikiem oddziału archiwalnego .
- **status**
  - projektowany
  - istniejący
  - archiwalny

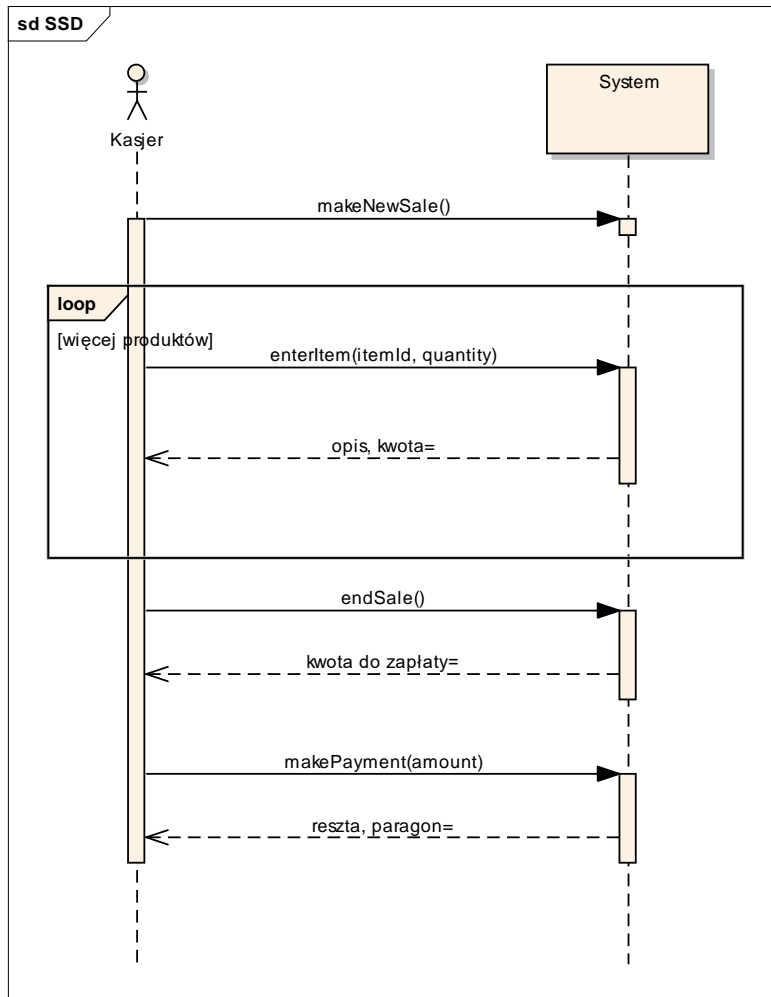
Pole wyliczane równocześnie z poziomem. Oddział, którego data założenia jest późniejsza od bieżącej, jest projektowany. Oddział, którego poziom jest większy od maksymalnego jest archiwalny

ltd..

## 6 Projektowanie analityczne – mapy procesów (Analysis)

### 6.1 Diagramy czynności i sekwencji jako uzupełnienie przypadków użycia

Systemowy Diagram Sekwencji - zdarzenia systemowe w jednym, głównym scenariuszu przypadku użycia.



**Demo:** przykład procesu biznesowego z rzeczywistego dokumentu projektowego

## **7 Dalsze etapy UP (Design, Implementation, Test, Deployment)**

Etapowi projektowania architektury poświęcimy praktycznie cały pozostały czas wykładu. Pozostałymi elementami UP nie będziemy się zajmować.